Assignment # 02

Cloud Computing

Advanced Terraform & Nginx Multi-Tier Architecture

Submitted By:

Name: Arooj Saleem

roll # 2023-bse-013-A

Instructor Name: Waqas Saleem

submission Date: 30th dec, 2025

Table of Contents

1. Executive Summary

This assignment focuses on designing and implementing a multi-tier cloud infrastructure using Terraform and Nginx on Amazon Web Services (AWS). The infrastructure includes a reverse proxy Nginx server, multiple backend Apache web servers, modular Terraform code, and secure networking components.

The project demonstrates Infrastructure as Code (IaC) principles by using reusable Terraform modules for networking, security, and web servers. High availability is achieved through Nginx load balancing with a backup server configuration, while HTTPS, caching, and security headers improve performance and security.

The assignment covers complete lifecycle management including infrastructure deployment, testing, performance verification, and safe resource cleanup. This project enhanced practical understanding of Terraform, AWS EC2, Nginx reverse proxy configuration, and real-world cloud deployment practices.

2. Architecture Design

2.1 Architecture Overview

The architecture of this project follows a multi-tier design deployed on Amazon Web Services (AWS). The infrastructure consists of a public-facing Nginx reverse proxy server and multiple backend Apache web servers. Terraform is used to provision and manage all cloud resources using Infrastructure as Code (IaC) principles.

Client requests from the internet are first received by the Nginx server, which acts as a load balancer and SSL termination point. The Nginx server then forwards the requests to backend web servers running Apache, which serve the application content. A backup backend server is also configured to ensure high availability in case one of the primary servers fails.

The architecture is designed to be scalable, secure, and fault tolerant. Backend servers are isolated from direct internet access and can only be reached through the Nginx reverse proxy. This separation improves security and simplifies traffic management.

2.2 Architecture Diagram

The following text-based diagram represents the overall architecture of the deployed multi-tier web infrastructure.

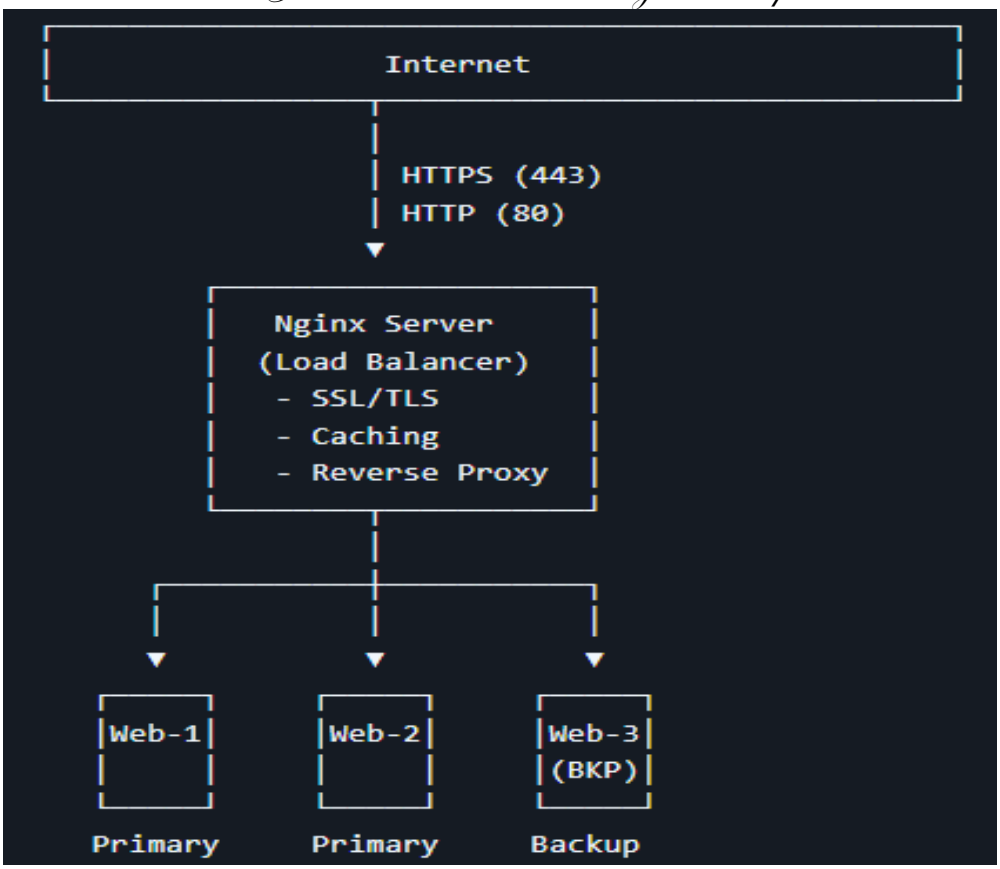


Figure 1: Text-based representation of the AWS multi-tier architecture

2.3 Security Design

2.3.1 Security Groups

Security groups are used to control access to EC2 instances. The Nginx reverse proxy allows inbound HTTP (80) and HTTPS (443) traffic from the internet, while SSH access is restricted for administrative use. Backend web servers only allow HTTP traffic from the Nginx server and are not directly accessible from the internet, improving overall security.

2.3.2 Network Access Control

Network access control is enforced by deploying the Nginx server in a public subnet and backend servers in private subnets. All external traffic is routed through the Nginx reverse proxy before reaching backend servers. This approach ensures controlled access, centralized traffic handling, and enhanced security.

3. Implementation Details

3.1 Infrastructure Setup

In this part, the base AWS infrastructure was provisioned using Terraform. A modular project structure was implemented to improve maintainability and reuse. Networking resources including VPC, subnet,

internet gateway, and routing configuration were created using a dedicated networking module. Variables and locals were used to manage configuration values, and a .gitignore file was applied to prevent sensitive files from being committed. The infrastructure was validated and deployed successfully using Terraform.

```
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $ tree -L 3
.
├── README.md
├── locals.tf
├── main.tf
├── modules
│   ├── networking
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   ├── security
│   │   ├── main.tf
│   │   ├── outputs.tf
│   │   └── variables.tf
│   └── webserver
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.tf
├── scripts
│   ├── apache-setup.sh
│   └── nginx-setup.sh
├── terraform.tfvars
└── variables.tf

6 directories, 17 files
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $
```

```
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $ cat .gitignore
.terraform/
terraform.tfstate
terraform.tfstate.backup

# Variable files (may contain secrets)
*.tfvars

# SSH keys
*.pem
*.key

# Logs
*.log
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $
```

README.md > # Assignment2 – Terraform Infrastructure Deployment > ## Project Structure

```
# Assignment2 – Terraform Infrastructure Deployment

## Project Overview
This project demonstrates the use of Terraform to deploy a
modular cloud infrastructure on AWS. The infrastructure follows
Infrastructure as Code (IaC) principles and is organized using
reusable Terraform modules for networking, security, and web
servers.

The deployment includes an Nginx reverse proxy and multiple
Apache backend servers to achieve load balancing and high
availability.

---

## Project Structure

Assignment2/
├── main.tf # Root Terraform configuration
├── variables.tf # Input variable definitions
├── outputs.tf # Output values
├── locals.tf # Local values and reusable expressions
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash

```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ code README.md
@23-22411-013-sys →/workspaces/Assignment2 (main) $
```

```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ cat variables.tf
variable "vpc_cidr_block" {
  description = "CIDR block for the VPC"
  type        = string

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

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

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

# Availability Zone
variable "availability_zone" {
  description = "Availability zone where resources will be deployed"
  type        = string
}

# Environment prefix
variable "env_prefix" {
```

0 0 Ln 79, Col 1 Spaces: 4 UTF-8 LF

```
@23-22411-013-sys ➜/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        = "arooj-assignment2"
instance_type     = "t3.micro"
public_key        = "~/.ssh/id_ed25519.pub"
private_key       = "~/.ssh/id_ed25519"

@23-22411-013-sys ➜/workspaces/Assignment2 (main) $
```

modules > networking > main.tf

```
8   resource "aws_subnet" "this" {
14    tags = {
15      Name = "${var.env_prefix}-public-subnet"
16    }
17  }
18  resource "aws_internet_gateway" "this" {
19    vpc_id = aws_vpc.this.id
20
21    tags = {
22      Name = "${var.env_prefix}-igw"
23    }
24  }
25  resource "aws_route_table" "this" {
26    vpc_id = aws_vpc.this.id
27
28    route {
29      cidr_block = "0.0.0.0/0"
30      gateway_id = aws_internet_gateway.this.id
31    }
32    tags = {
33      Name = "${var.env_prefix}-public-rt"
34    }
35  }
36  resource "aws_route_table_association" "this" {
37    subnet_id      = aws_subnet.this.id
38    route_table_id = aws_route_table.this.id
```

0 0 Ln 7, Col 2 Spaces: 4 UTF-8 LF

modules > networking > outputs.tf

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

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

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

output "route_table_id" {
  description = "ID of the route table"
  value       = aws_route_table.this.id
}
```

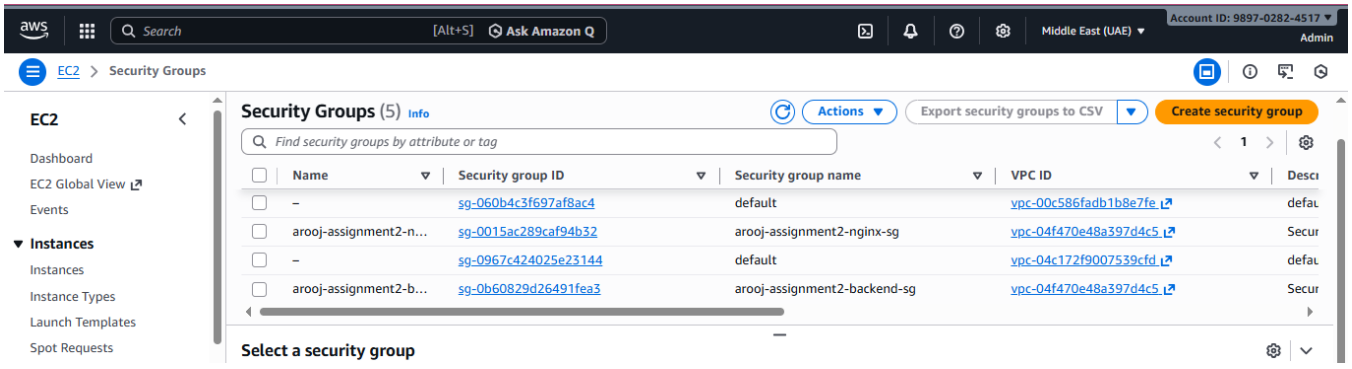
modules > security > main.tf

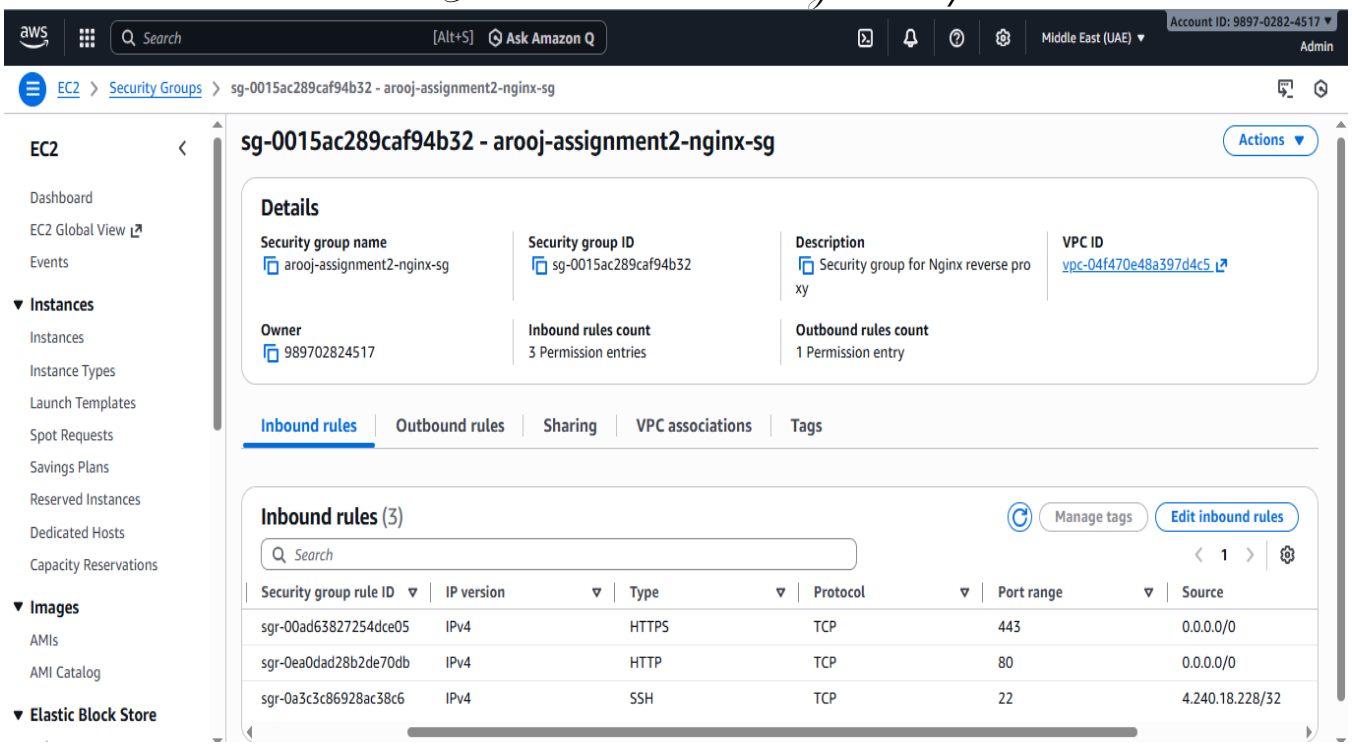
```hcl
resource "aws_security_group" "nginx_sg" {
  name        = "${var.env_prefix}-nginx-sg"
  description = "Security group for Nginx reverse proxy"
  vpc_id      = var.vpc_id

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

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

  ingress {
    description = "HTTPS access from anywhere"
    from_port   = 443
    to_port     = 443
    protocol    = "tcp"
```

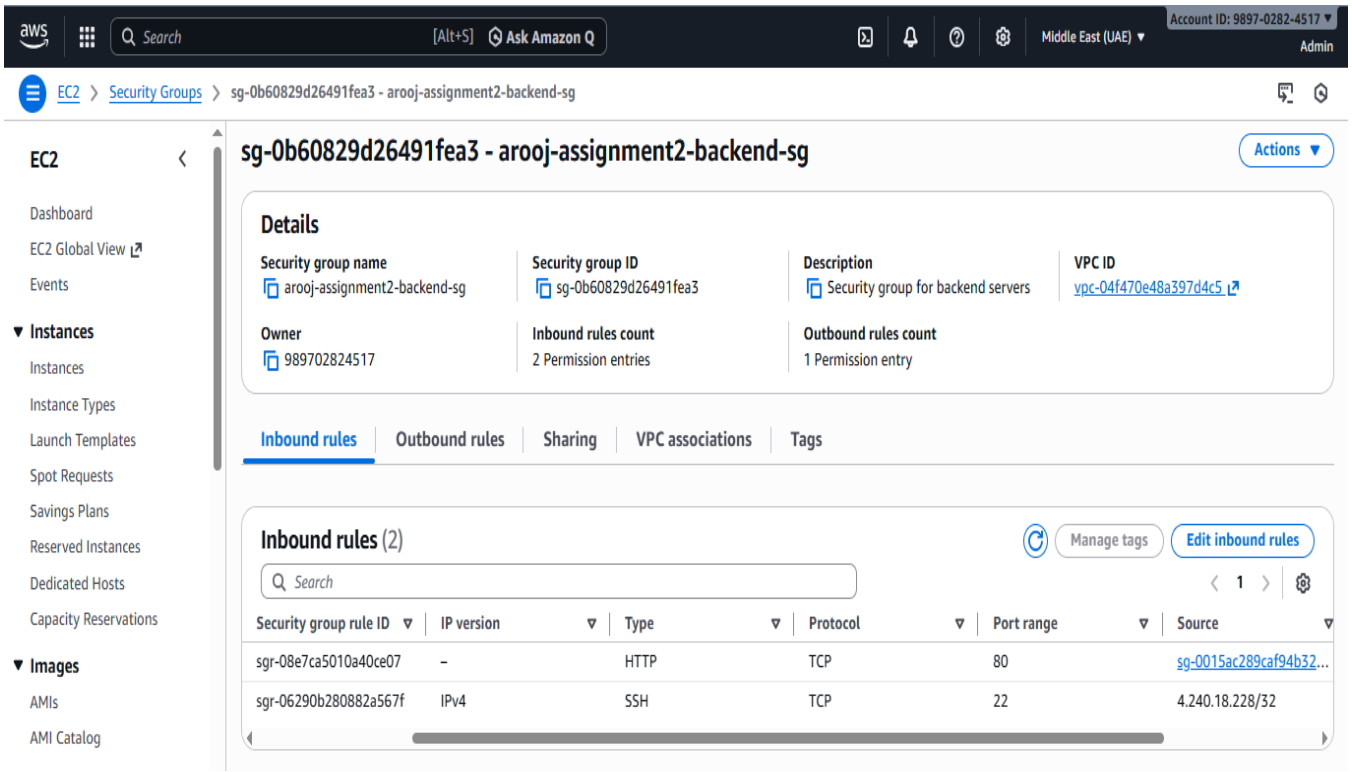
Ln 76, Col 1 Spaces: 4 UTF-8 LF

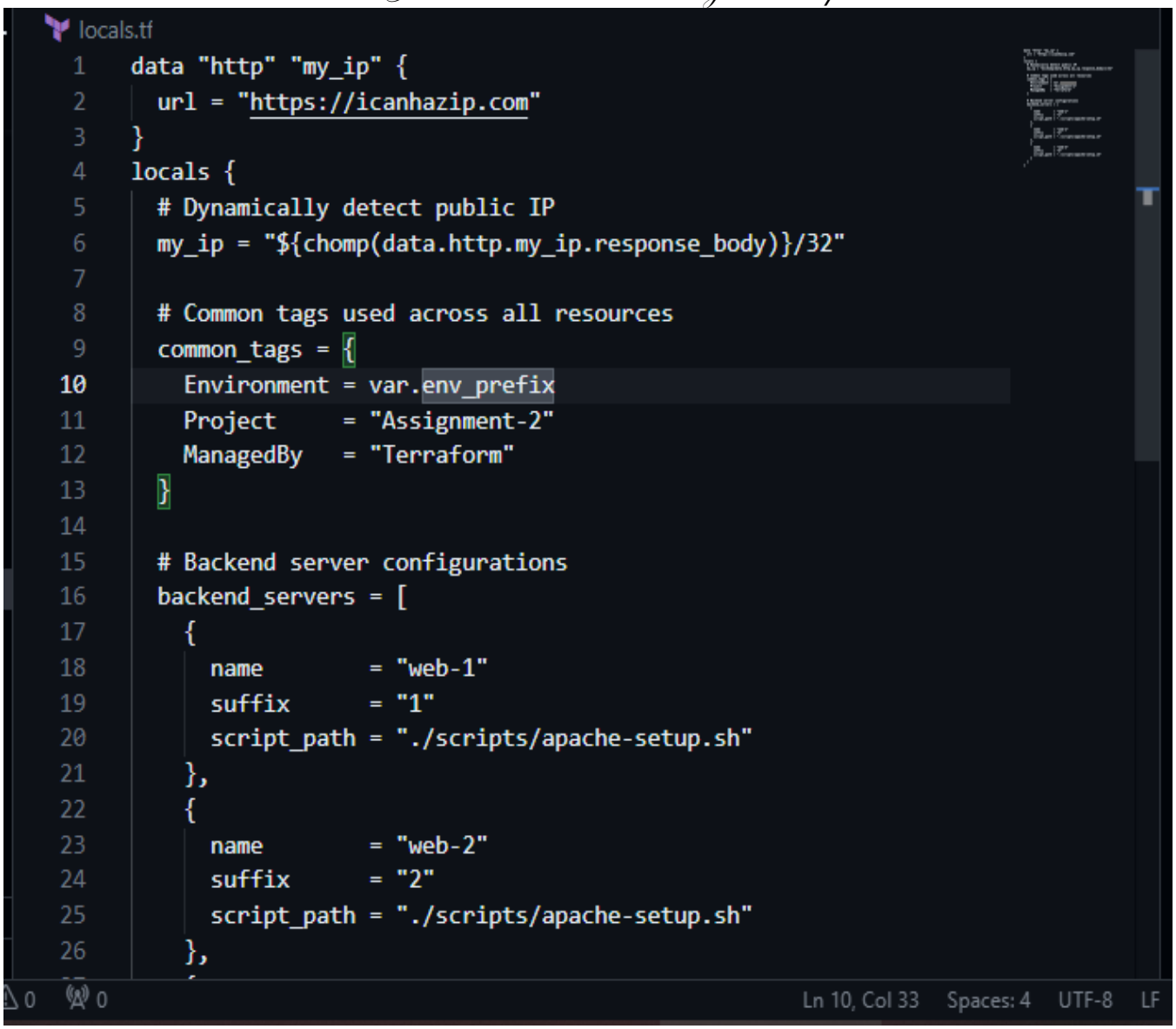
modules > security > main.tf

```
resource "aws_security_group" "nginx_sg" {
  tags = {
    Name = "${var.env_prefix}-nginx-sg"
  }
}

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

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

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

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




Search
[Alt+S]
Ask Amazon Q
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
EC2 > Security Groups > sg-0015ac289caf94b32 - arooj-assignment2-nginx-sg
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
sg-0015ac289caf94b32 - arooj-assignment2-nginx-sg
Actions
Details
Security group name
arooj-assignment2-nginx-sg
Security group ID
sg-0015ac289caf94b32
Description
Security group for Nginx reverse proxy
VPC ID
vpc-04f470e48a397d4c5
Owner
989702824517
Inbound rules count
3 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing
VPC associations
Tags
Inbound rules (3)
Manage tags
Edit inbound rules
Search
1
Security group rule ID
IP version
Type
Protocol
Port range
Source
sgr-00ad63827254dce05
IPv4
HTTPS
TCP
443
0.0.0.0/0
sgr-0ea0dad28b2de70db
IPv4
HTTP
TCP
80
0.0.0.0/0
sgr-0a3c3c86928ac38c6
IPv4
SSH
TCP
22
4.240.18.228/32


Search
[Alt+S]
Ask Amazon Q
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
EC2 > Security Groups > sg-0b60829d26491fea3 - arooj-assignment2-backend-sg
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
sg-0b60829d26491fea3 - arooj-assignment2-backend-sg
Actions
Details
Security group name
arooj-assignment2-backend-sg
Security group ID
sg-0b60829d26491fea3
Description
Security group for backend servers
VPC ID
vpc-04f470e48a397d4c5
Owner
989702824517
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing
VPC associations
Tags
Inbound rules (2)
Manage tags
Edit inbound rules
Search
1
Security group rule ID
IP version
Type
Protocol
Port range
Source
sgr-08e7ca5010a40ce07
-
HTTP
TCP
80
sg-0015ac289caf94b32...
sgr-06290b280882a567f
IPv4
SSH
TCP
22
4.240.18.228/32



3.2 Webserver Module

A reusable webserver module was implemented to deploy EC2 instances for both the Nginx reverse proxy and backend Apache servers. The module accepts configurable parameters such as instance type, security group, and user-data script. The same module was reused for all servers by passing different scripts and security groups, demonstrating modular and scalable infrastructure design.

modules > webserver > variables.tf

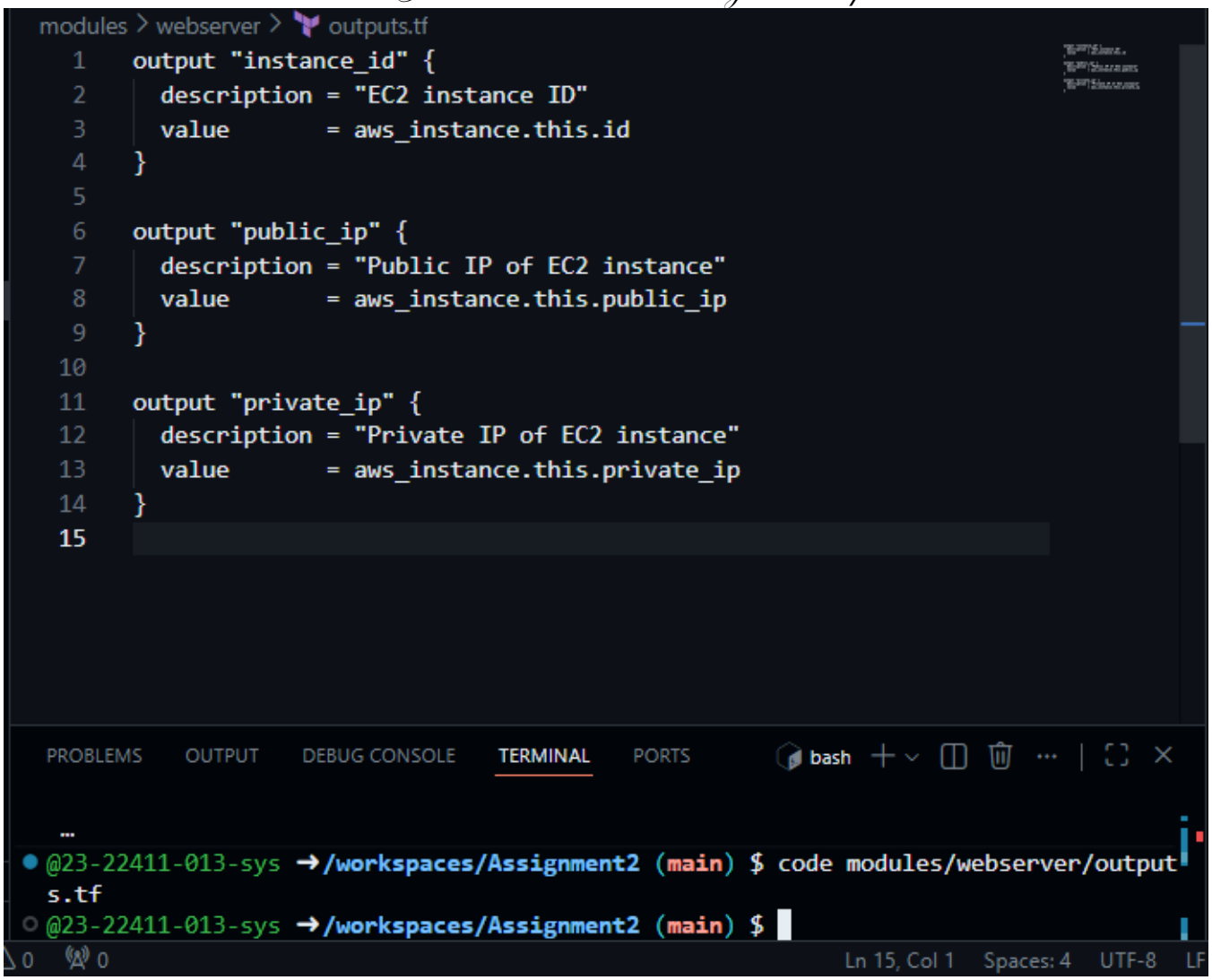
```
}
variable "instance_suffix" {
  description = "Unique suffix for instance and key pair"
  type        = string
}
variable "instance_type" {
  description = "EC2 instance type"
  type        = string
}
variable "availability_zone" {
  description = "Availability zone for EC2"
  type        = string
}
variable "vpc_id" {
  description = "VPC ID"
  type        = string
}
variable "subnet_id" {
  description = "Subnet ID"
  type        = string
}
variable "security_group_id" {
  description = "Security group ID"
  type        = string
}
variable "public_key" {
```

0 0 Ln 40, Col 2 Spaces: 4 UTF-8 LF

modules > webserver > main.tf

```
resource "aws_key_pair" "this" {
  key_name   = "${var.env_prefix}-${var.instance_name}-${var.instar
  public_key = file(var.public_key)
  tags = merge(
    var.common_tags,
    {
      Name = "${var.env_prefix}-${var.instance_name}-${var.instance
    }
  )
}
resource "aws_instance" "this" {
  ami                    = data.aws_ami.amazon_linux.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.this.key_name
  associate_public_ip_address = true
  user_data = file(var.script_path)
  tags = merge(
    var.common_tags,
    {
      Name = "${var.env_prefix}-${var.instance_name}-${var.instance
    }
  )
}
```

0 0 Ln 29, Col 27 Spaces: 4 UTF-8 LF



For the Nginx server, the module is invoked with the Nginx security group and nginx-setup.sh script, whereas for backend servers, the same module is reused with the backend security group and apache-setup.sh script.

main.tf

```
16  # Nginx Server
17  module "nginx_server" {
18    source            = "./modules/webserver"
19    env_prefix        = var. env_prefix
20    instance_name     = "nginx-proxy"
21    instance_type     = var.instance_type
22    availability_zone = var.availability_zone
23    vpc_id            = module.networking.vpc_id
24    subnet_id         = module.networking.subnet_id
25    security_group_id = module.security. nginx_sg_id
26    public_key        = var.public_key
27    script_path       = "./scripts/nginx-setup.sh"
28    instance_suffix   = "nginx"
29    common_tags       = local.common_tags
30  }
31  # Backend Servers (use for_each or count)
32  module "backend_servers" {
33    for_each = { for idx, server in local.backend_servers : server.na
34    source            = "./modules/webserver"
35    env_prefix        = var.env_prefix
36    instance_name     = each.value.name
37    instance_type     = var.instance_type
38    availability_zone = var.availability_zone
39    vpc_id            = module. networking.vpc_id
40    subnet_id         = module.networking. subnet_id
41    security_group_id = module.security.backend_sg_id
```

0 0 Ln 30, Col 2 Spaces: 4 UTF-8 LF

Proper security group assignment is ensured by passing the correct security group IDs to each webserver module instance. The Nginx server is associated with the Nginx security group, which allows public HTTP and HTTPS access, while the backend servers are associated with the backend security group that only permits HTTP traffic from the Nginx server. All required module variables are explicitly passed from the root configuration using variables, locals, and outputs from other modules, ensuring correct dependencies, flexibility, and reusability of the webserver module.

3.3 Server Scripts

Shell scripts were used to automate server configuration. The Apache setup script installs and configures the Apache web server and displays instance metadata on a custom web page. The Nginx setup script configures the reverse proxy, SSL termination, caching, and load balancing between backend servers. Both scripts were executed using EC2 user-data to ensure automatic configuration during instance launch.

scripts > $ apache-setup.sh

```
7   # Install Apache
8   yum install httpd -y
9
10  # Start and enable Apache
11  systemctl start httpd
12  systemctl enable httpd
13
14  # Get metadata token (IMDSv2)
15  TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" \
16    -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
17
18  # Get instance metadata
19  PRIVATE_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
20    http://169.254.169.254/latest/meta-data/local-ipv4)
21  PUBLIC_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
22    http://169.254.169.254/latest/meta-data/public-ipv4)
23  PUBLIC_DNS=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
24    http://169.254.169.254/latest/meta-data/public-hostname)
25  INSTANCE_ID=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
26    http://169.254.169.254/latest/meta-data/instance-id)
27
28  # Set hostname
29  hostnamectl set-hostname myapp-webserver
30
31  # Create custom HTML page
32  cat > /var/www/html/index.html <<EOF
```

0 0 Ln 74, Col 44 Spaces: 4 UTF-8 LF

```
@23-22411-013-sys ➜ /workspaces/Assignment2 (main) $ code scripts/apache-setup.sh
@23-22411-013-sys ➜ /workspaces/Assignment2 (main) $ chmod +x scripts/apache-setup
.sh
```

Not secure 40.172.215.143

🚀 Backend Web Server - Assignment 2

Hostname: myapp-webserver

Instance ID: i-01189ee444d0e9051

Private IP: 10.0.10.196

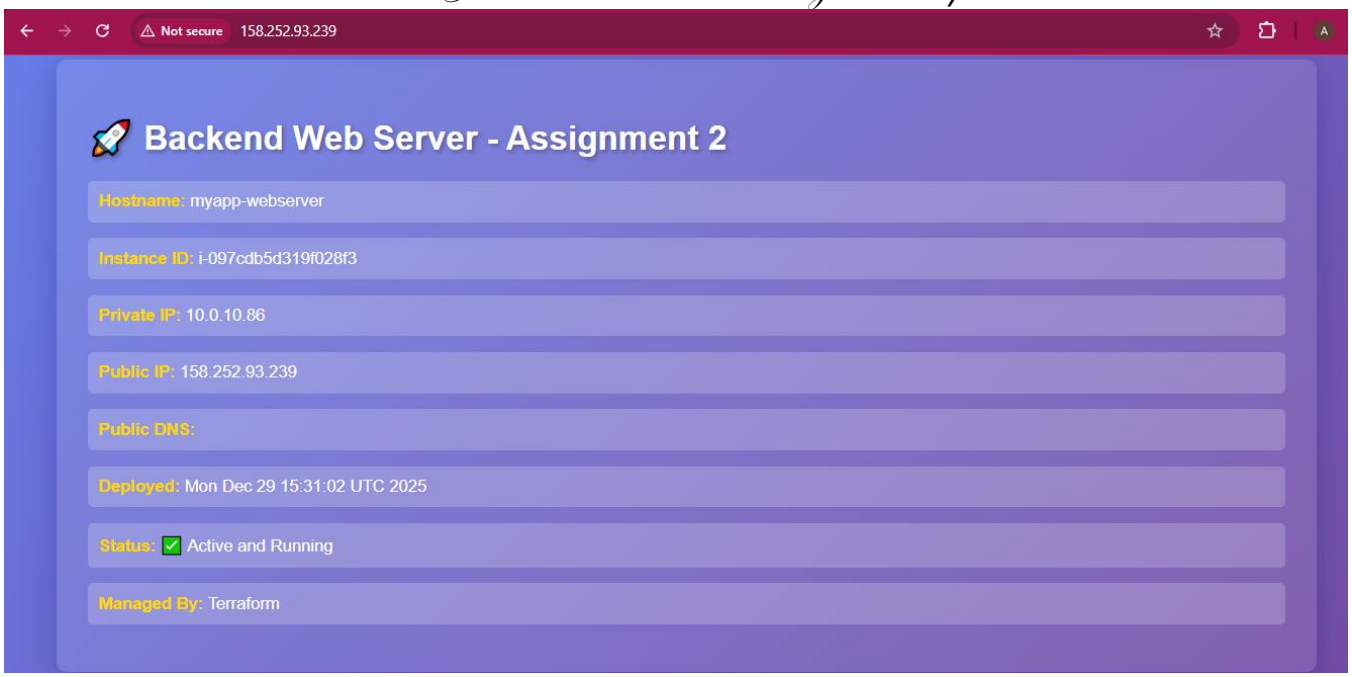
Public IP: 40.172.215.143

Public DNS:

Deployed: Mon Dec 29 15:30:51 UTC 2025

Status: ✅ Active and Running

Managed By: Terraform


Not secure
158.252.93.239
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-097cdb5d319f028f3
Private IP: 10.0.10.86
Public IP: 158.252.93.239
Public DNS:
Deployed: Mon Dec 29 15:31:02 UTC 2025
Status: Active and Running
Managed By: Terraform


Not secure
158.252.79.207
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-0cb7f4968a752c4fd
Private IP: 10.0.10.249
Public IP: 158.252.79.207
Public DNS:
Deployed: Mon Dec 29 15:30:51 UTC 2025
Status: Active and Running
Managed By: Terraform

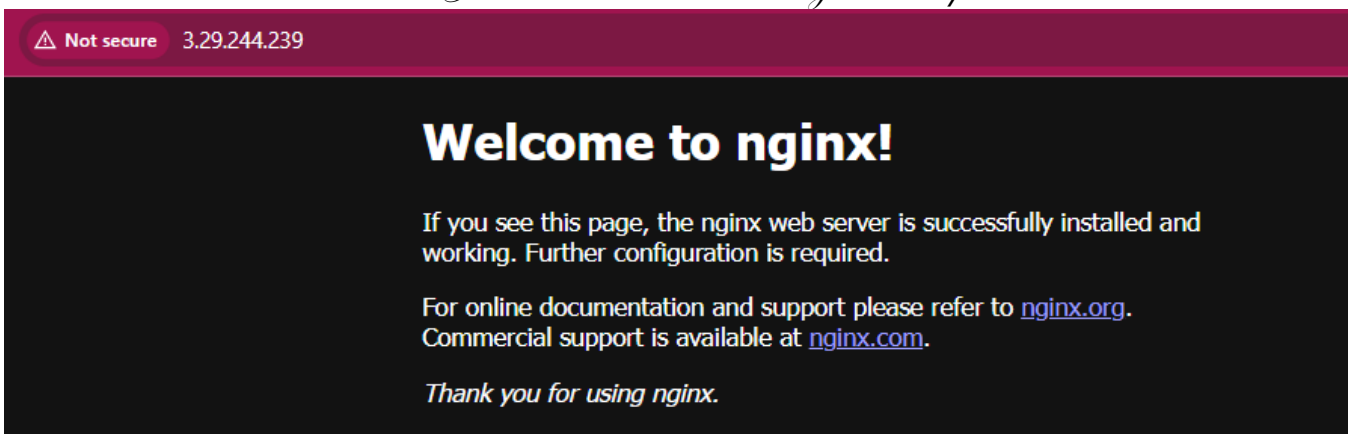
scripts > $ nginx-setup.sh

```
3
4   # Update and install Nginx
5   yum update -y
6   yum install -y nginx openssl
7   systemctl start nginx
8   systemctl enable nginx
9
10  # Create SSL directories
11  mkdir -p /etc/ssl/private
12  mkdir -p /etc/ssl/certs
13
14  # Get metadata token
15  TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" \
16    -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
17
18  # Get public IP
19  PUBLIC_IP=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \
20    http://169.254.169.254/latest/meta-data/public-ipv4)
21
22  # Generate self-signed certificate
23  openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
24    -keyout /etc/ssl/private/selfsigned.key \
25    -out /etc/ssl/certs/selfsigned.crt \
26    -subj "/CN=$PUBLIC_IP" \
27    -addext "subjectAltName=IP:$PUBLIC_IP" \
28    -addext "basicConstraints=CA:FALSE" \
```

0 0 Ln 110, Col 47 Spaces: 4 UTF-8 LF

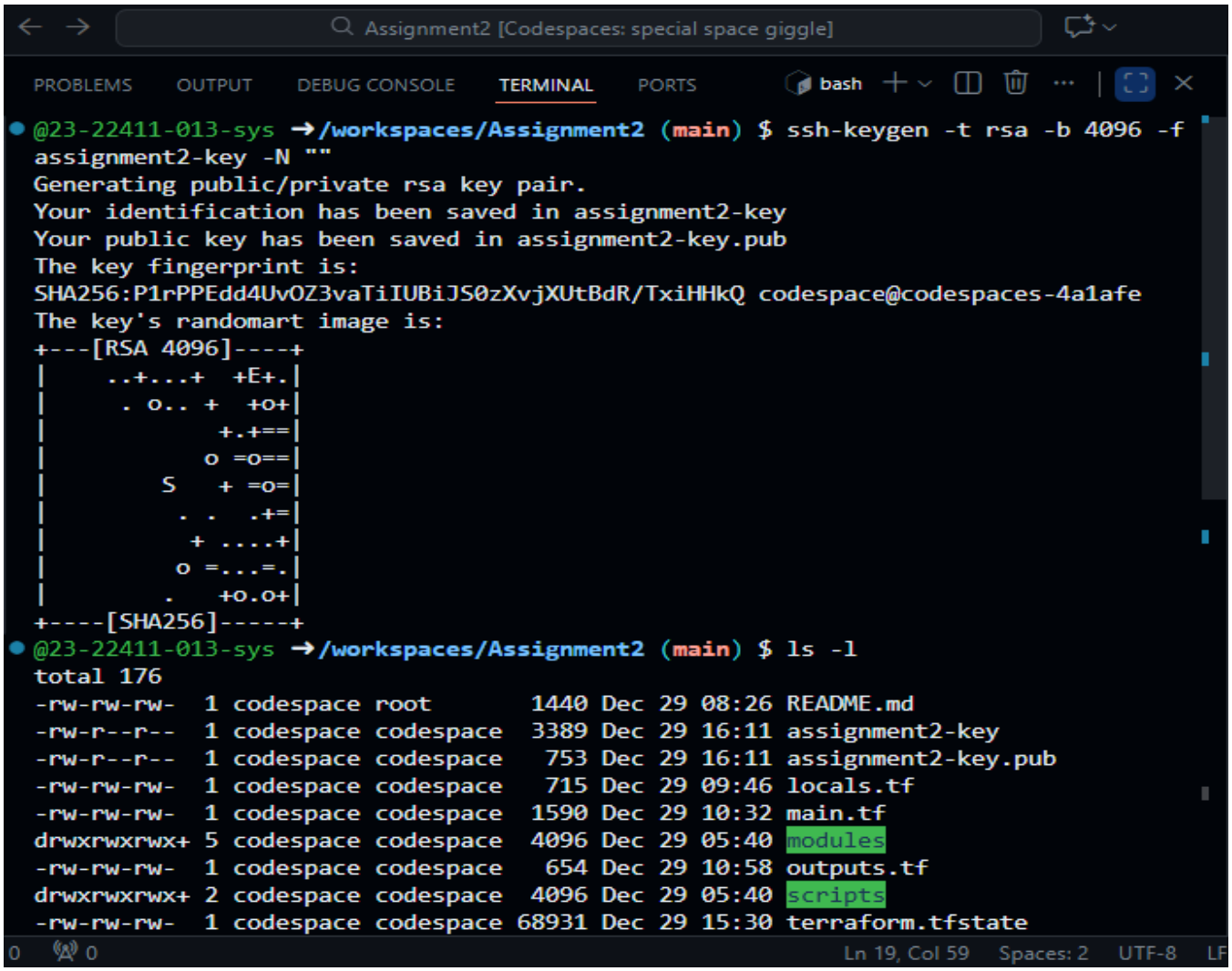
```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ terraform output
backend_private_ips = {
  "web-1" = "10.0.10.214"
  "web-2" = "10.0.10.38"
  "web-3" = "10.0.10.85"
}
backend_public_ips = {
  "web-1" = "158.252.82.34"
  "web-2" = "3.29.63.106"
  "web-3" = "158.252.34.78"
}
nginx_private_ip = "10.0.10.226"
nginx_public_ip = "3.29.244.239"
@23-22411-013-sys →/workspaces/Assignment2 (main) $
```

0 0 Ln 110, Col 47 Spac



3.4 Deployment

The complete infrastructure was deployed using Terraform. Terraform initialization and validation were performed before applying the configuration. After deployment, all resources including EC2 instances, security groups, and networking components were verified using the AWS Management Console to ensure correct provisioning.



```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ terraform init

Initializing the backend...
Initializing modules...

Initializing provider plugins...
- Reusing previous version of hashicorp/http from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0
- Using previously-installed hashicorp/http v3.5.0

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
@23-22411-013-sys →/workspaces/Assignment2 (main) $
```

```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ terraform validate
Success! The configuration is valid.

@23-22411-013-sys →/workspaces/Assignment2 (main) $
```

```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ terraform plan
Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # module.security.aws_security_group.backend_sg will be updated in-place
  ~ resource "aws_security_group" "backend_sg" {
        id                     = "sg-0b60829d26491fea3"
      ~ ingress                = [
          - {
              - cidr_blocks      = [
                  - "0.0.0.0/0",
                ]
              - description      = ""
              - from_port        = 80
              - ipv6_cidr_blocks = []
              - prefix_list_ids  = []
              - protocol         = "tcp"
              - security_groups  = []
              - self             = false
              - to_port          = 80
            },
          + {
              + cidr_blocks      = []
              + description      = "HTTP access from Nginx SG only"
              + from_port        = 80
              + ipv6_cidr_blocks = []
              + prefix_list_ids  = []
```

```
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $ terraform apply -auto-approve
        tags                      = {
            "Name" = "arooj-assignment2-backend-sg"
        }
        # (8 unchanged attributes hidden)
    }

Plan: 0 to add, 1 to change, 0 to destroy.
module.security.aws_security_group.backend_sg: Modifying... [id=sg-0b60829d26491fe
a3]
module.security.aws_security_group.backend_sg: Modifications complete after 2s [id
=sg-0b60829d26491fea3]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

Outputs:

backend_private_ips = {
  "web-1" = "10.0.10.196"
  "web-2" = "10.0.10.86"
  "web-3" = "10.0.10.249"
}
backend_public_ips = {
  "web-1" = "40.172.215.143"
  "web-2" = "158.252.93.239"
  "web-3" = "158.252.79.207"
}
nginx_private_ip = "10.0.10.226"
nginx_public_ip = "51.112.45.241"
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $
```

0 0 Ln 19, Col 59 Spaces: 2 UTF-8 LF

outputs.tf

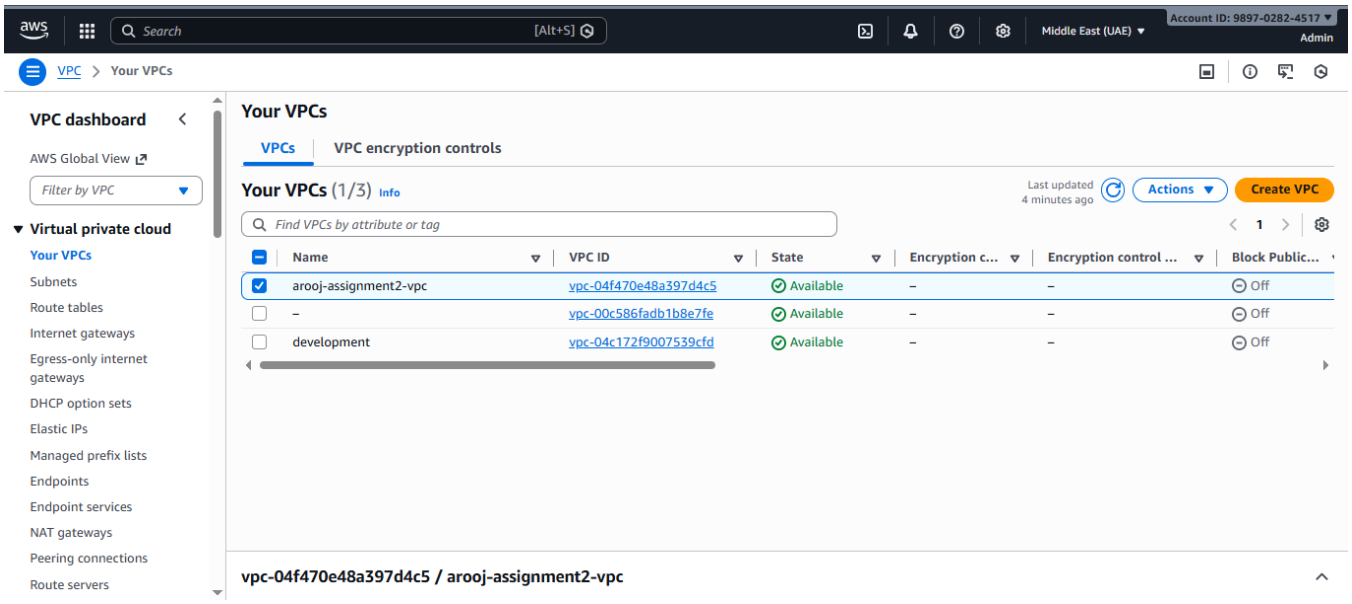
```
23  output "nginx_instance_id" {
26  }
27  # =========================
28  # Backend Servers Outputs
29  # =========================
30  output "backend_public_ips" {
31    description = "Public IPs of backend web servers"
32    value = {
33      for name, mod in module.backend_servers :
34      name => mod.public_ip
35    }
36  }
37  output "backend_private_ips" {
38    description = "Private IPs of backend web servers"
39    value = {
40      for name, mod in module.backend_servers :
41      name => mod.private_ip
42    }
43  }
44  output "backend_servers_info" {
45    description = "Backend servers information"
46    value = {
47      for name, server in module.backend_servers : name => {
48        instance_id = server.instance_id
49        public_ip   = server.public_ip
50        private_ip  = server.private_ip
```

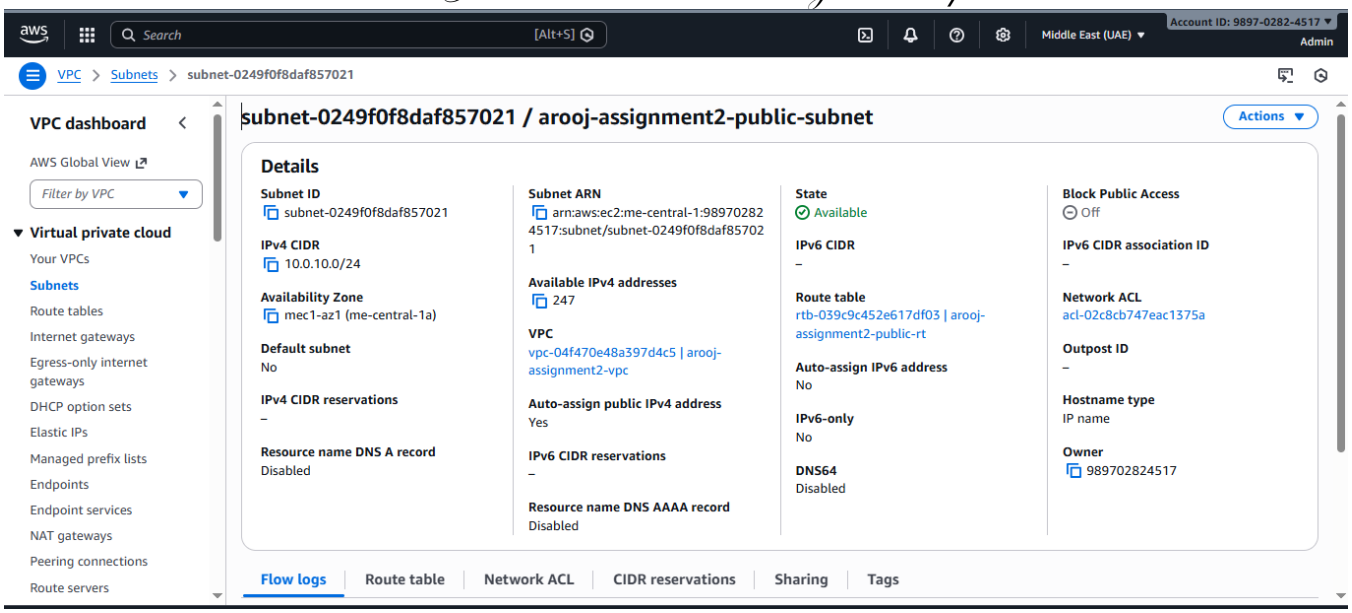
0 0 Ln 78, Col 8 Spaces: 2 UTF-8 LF

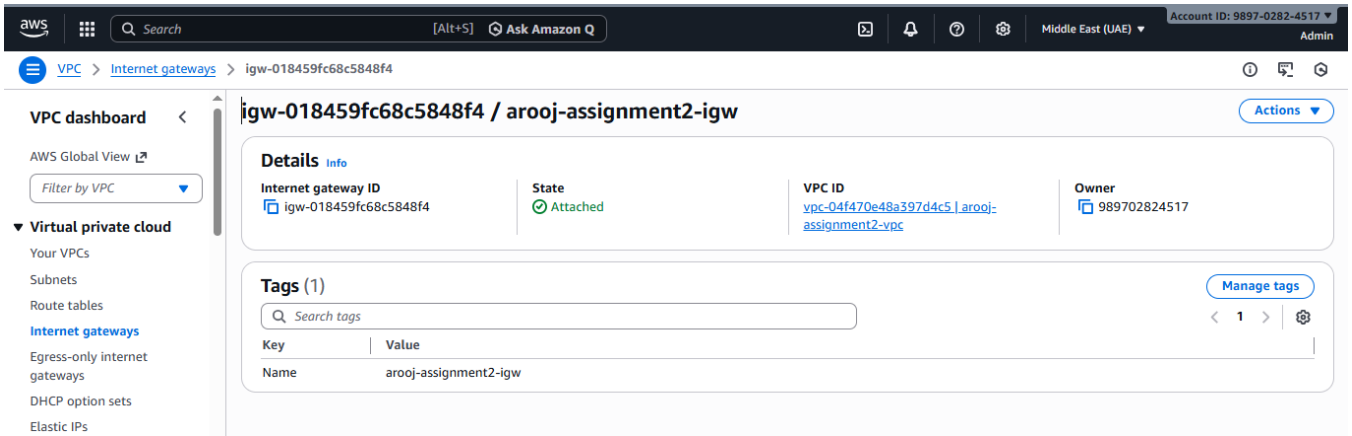
```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ terraform output
backend_private_ips = {
  "web-1" = "10.0.10.196"
  "web-2" = "10.0.10.86"
  "web-3" = "10.0.10.249"
}
backend_public_ips = {
  "web-1" = "40.172.215.143"
  "web-2" = "158.252.93.239"
  "web-3" = "158.252.79.207"
}
backend_servers_info = {
  "web-1" = {
    "instance_id" = "i-01189ee444d0e9051"
    "private_ip" = "10.0.10.196"
    "public_ip" = "40.172.215.143"
  }
  "web-2" = {
    "instance_id" = "i-097cdb5d319f028f3"
    "private_ip" = "10.0.10.86"
    "public_ip" = "158.252.93.239"
  }
  "web-3" = {
    "instance_id" = "i-0cb7f4968a752c4fd"
    "private_ip" = "10.0.10.249"
    "public_ip" = "158.252.79.207"
  }
}
configuration_guide = <<EOT
========================================
```

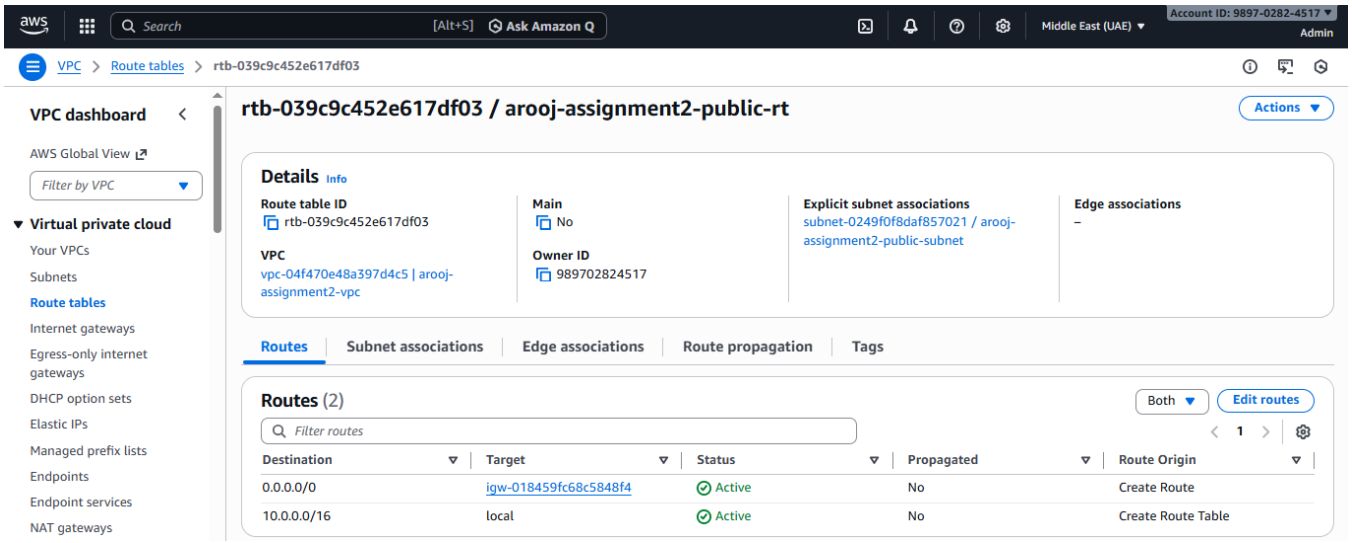
```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ terraform output
========================================
DEPLOYMENT SUCCESSFUL!
========================================
Next Steps:
1. SSH into Nginx server:
   ssh ec2-user@51.112.45.241
2. Edit Nginx config:
   sudo vim /etc/nginx/nginx.conf
3. Update backend IPs in upstream block:
   - BACKEND_IP_1: 10.0.10.196
   - BACKEND_IP_2: 10.0.10.86
   - BACKEND_IP_3: 10.0.10.249
4. Restart Nginx:
   sudo systemctl restart nginx
5. Test:
   https://51.112.45.241
Backend Servers:
- web-1: 40.172.215.143 (private: 10.0.10.196)
    - web-2: 158.252.93.239 (private: 10.0.10.86)
    - web-3: 158.252.79.207 (private: 10.0.10.249)
========================================

EOT
nginx_instance_id = "i-0c1de976fe01c15a2"
nginx_private_ip = "10.0.10.226"
nginx_public_ip = "51.112.45.241"
subnet_id = "subnet-0249f0f8daf857021"
vpc_id = "vpc-04f470e48a397d4c5"
@23-22411-013-sys →/workspaces/Assignment2 (main) $
0  0                                   Ln 78, Col 8  Spaces: 2  UTF-8
```

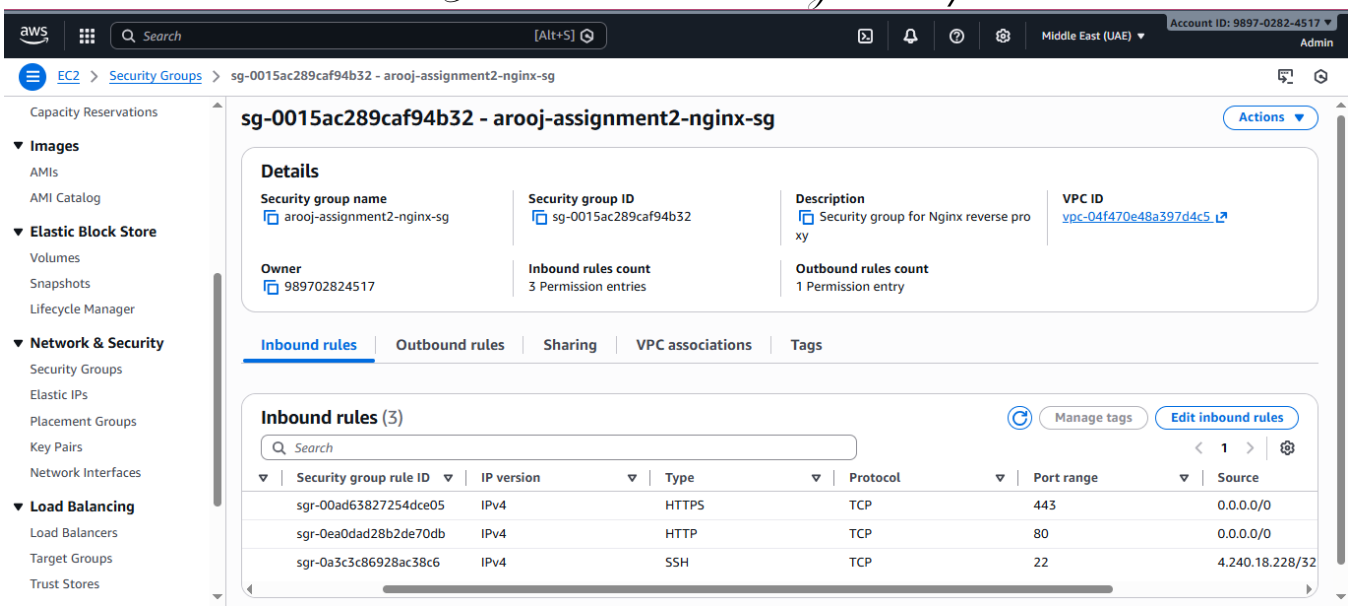
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                bash

@23-22411-013-sys →/workspaces/Assignment2 (main) $ terraform output -json > outp
uts.json
@23-22411-013-sys →/workspaces/Assignment2 (main) $ ls
README.md            main.tf       scripts                    variables.tf
assignment2-key      modules       terraform.tfstate
assignment2-key.pub  outputs.json  terraform.tfstate.backup
locals.tf            outputs.tf    terraform.tfvars
@23-22411-013-sys →/workspaces/Assignment2 (main) $ cat outputs.json
{
  "backend_private_ips": {
    "sensitive": false,
    "type": [
      "object",
      {
        "web-1": "string",
        "web-2": "string",
        "web-3": "string"
      }
    ],
    "value": {
      "web-1": "10.0.10.196",
      "web-2": "10.0.10.86",
      "web-3": "10.0.10.249"
    }
  },
  "backend_public_ips": {
    "sensitive": false,
    "type": [
      "object",
      {
```

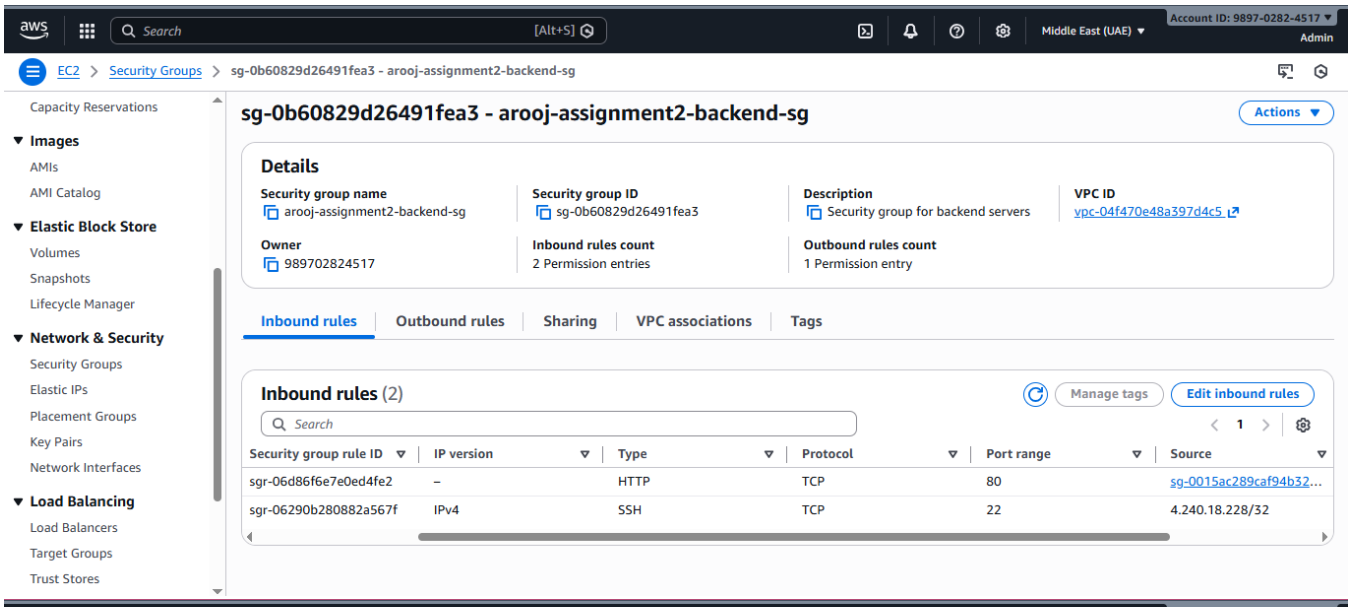


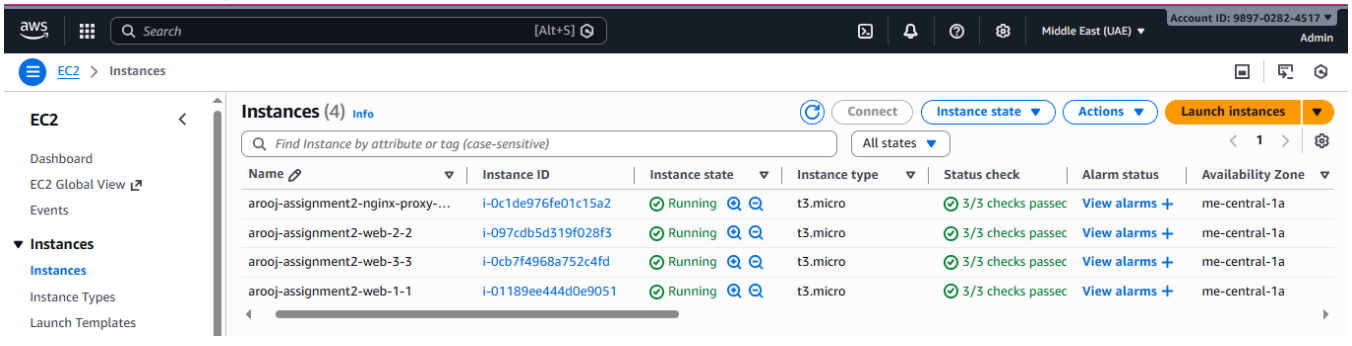

Search [Alt+S]
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
VPC > Subnets > subnet-0249f0f8daf857021
VPC dashboard
AWS Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Route servers
subnet-0249f0f8daf857021 / arooj-assignment2-public-subnet
Actions
Details
Subnet ID
subnet-0249f0f8daf857021
IPv4 CIDR
10.0.10.0/24
Availability Zone
mec1-az1 (me-central-1a)
Default subnet
No
IPv4 CIDR reservations
–
Resource name DNS A record
Disabled
Subnet ARN
arn:aws:ec2:me-central-1:989702824517:subnet/subnet-0249f0f8daf857021
Available IPv4 addresses
247
VPC
vpc-04f470e48a397d4c5 | arooj-assignment2-vpc
Auto-assign public IPv4 address
Yes
IPv6 CIDR reservations
–
Resource name DNS AAAA record
Disabled
State
Available
IPv6 CIDR
–
Route table
rtb-039c9c452e617df03 | arooj-assignment2-public-rt
Auto-assign IPv6 address
No
IPv6-only
No
DNS64
Disabled
Block Public Access
Off
IPv6 CIDR association ID
–
Network ACL
acl-02c8cb747eac1375a
Outpost ID
–
Hostname type
IP name
Owner
989702824517
Flow logs
Route table
Network ACL
CIDR reservations
Sharing
Tags

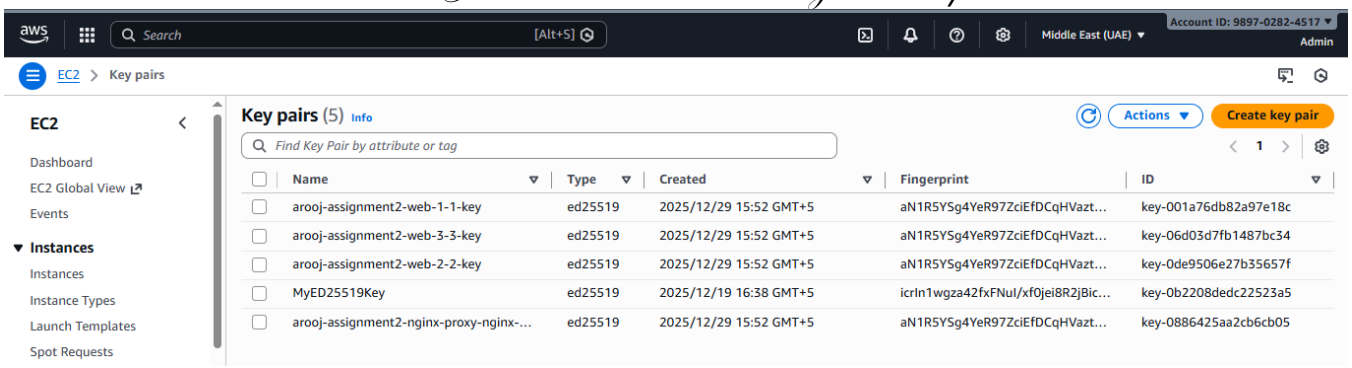

Search [Alt+S] Ask Amazon Q
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
VPC > Internet gateways > igw-018459fc68c5848f4
VPC dashboard
AWS Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
igw-018459fc68c5848f4 / arooj-assignment2-igw
Actions
Details Info
Internet gateway ID
igw-018459fc68c5848f4
State
Attached
VPC ID
vpc-04f470e48a397d4c5 | arooj-assignment2-vpc
Owner
989702824517
Tags (1)
Manage tags
Search tags
Key
Value
Name
arooj-assignment2-igw


Search [Alt+S] Ask Amazon Q
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
VPC > Route tables > rtb-039c9c452e617df03
VPC dashboard
AWS Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
rtb-039c9c452e617df03 / arooj-assignment2-public-rt
Actions
Details Info
Route table ID
rtb-039c9c452e617df03
Main
No
Explicit subnet associations
subnet-0249f0f8daf857021 / arooj-assignment2-public-subnet
Edge associations
–
VPC
vpc-04f470e48a397d4c5 | arooj-assignment2-vpc
Owner ID
989702824517
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (2)
Both
Edit routes
Filter routes
Destination
Target
Status
Propagated
Route Origin
0.0.0.0/0
igw-018459fc68c5848f4
Active
No
Create Route
10.0.0.0/16
local
Active
No
Create Route Table


Search [Alt+S]
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
EC2 > Security Groups > sg-0015ac289caf94b32 - arooj-assignment2-nginx-sg
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
sg-0015ac289caf94b32 - arooj-assignment2-nginx-sg
Actions
Details
Security group name
arooj-assignment2-nginx-sg
Security group ID
sg-0015ac289caf94b32
Description
Security group for Nginx reverse proxy
VPC ID
vpc-04f470e48a397d4c5
Owner
989702824517
Inbound rules count
3 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing
VPC associations
Tags
Inbound rules (3)
Manage tags
Edit inbound rules
Search
Security group rule ID
IP version
Type
Protocol
Port range
Source
sgr-00ad63827254dce05
IPv4
HTTPS
TCP
443
0.0.0.0/0
sgr-0ea0dad28b2de70db
IPv4
HTTP
TCP
80
0.0.0.0/0
sgr-0a3c3c86928ac38c6
IPv4
SSH
TCP
22
4.240.18.228/32


Search [Alt+S]
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
EC2 > Security Groups > sg-0b60829d26491fea3 - arooj-assignment2-backend-sg
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
sg-0b60829d26491fea3 - arooj-assignment2-backend-sg
Actions
Details
Security group name
arooj-assignment2-backend-sg
Security group ID
sg-0b60829d26491fea3
Description
Security group for backend servers
VPC ID
vpc-04f470e48a397d4c5
Owner
989702824517
Inbound rules count
2 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing
VPC associations
Tags
Inbound rules (2)
Manage tags
Edit inbound rules
Search
Security group rule ID
IP version
Type
Protocol
Port range
Source
sgr-06d86f6e7e0ed4fe2
–
HTTP
TCP
80
sg-0015ac289caf94b32...
sgr-06290b280882a567f
IPv4
SSH
TCP
22
4.240.18.228/32


Search [Alt+S]
Middle East (UAE)
Account ID: 9897-0282-4517
Admin
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Instances (4) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
arooj-assignment2-nginx-proxy-...
i-0c1de976fe01c15a2
Running
t3.micro
3/3 checks passed
View alarms
me-central-1a
arooj-assignment2-web-2-2
i-097cdb5d319f028f3
Running
t3.micro
3/3 checks passed
View alarms
me-central-1a
arooj-assignment2-web-3-3
i-0cb7f4968a752c4fd
Running
t3.micro
3/3 checks passed
View alarms
me-central-1a
arooj-assignment2-web-1-1
i-01189ee444d0e9051
Running
t3.micro
3/3 checks passed
View alarms
me-central-1a



3.5 Testing and Verification

3.5.1 Update Nginx Backend Configuration

```
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $ terraform output
Backend Servers:
- web-1: 40.172.187.154 (private: 10.0.10.34)
    - web-2: 51.112.228.94 (private: 10.0.10.205)
    - web-3: 40.172.100.235 (private: 10.0.10.10)
========================================

EOT
nginx_instance_id = "i-068f810807387421c"
nginx_private_ip = "10.0.10.162"
nginx_public_ip = "3.28.131.196"
subnet_id = "subnet-0bb3e6cd3dade5025"
vpc_id = "vpc-069479242b0eb280b"
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $
```

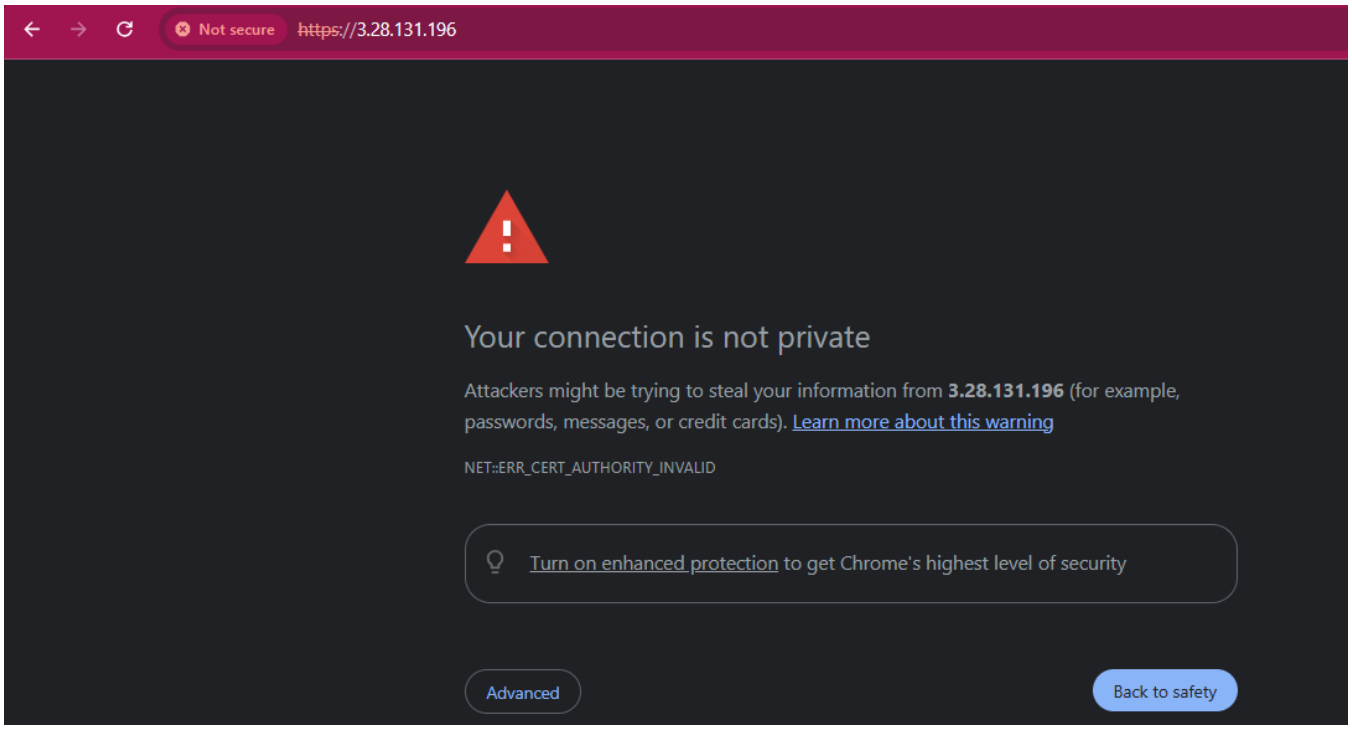
```
ec2-user@myapp-webserver:~

Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$ ssh -i ~/.ssh/assignment2-key ec2-user@3.28.131.196
The authenticity of host '3.28.131.196 (3.28.131.196)' can't be established.
ED25519 key fingerprint is SHA256:cSzxlJ5AqhPW9XxJIVBwNmiy+UlK3dadhvMvIAT2hqY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.28.131.196' (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 ~]$
```

```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ ^C
[ec2-user@myapp-webserver ~]$ sudo ls /etc/nginx/conf.d
backend.conf
[ec2-user@myapp-webserver ~]$ sudo cat /etc/nginx/conf.d/backend.conf
upstream backend_servers {
    server 10.0.10.34:80;
    server 10.0.10.205:80;
    server 10.0.10.10:80 backup;
}

server {
    listen 80;
    server_name _;

    location / {
        proxy_pass http://backend_servers;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

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

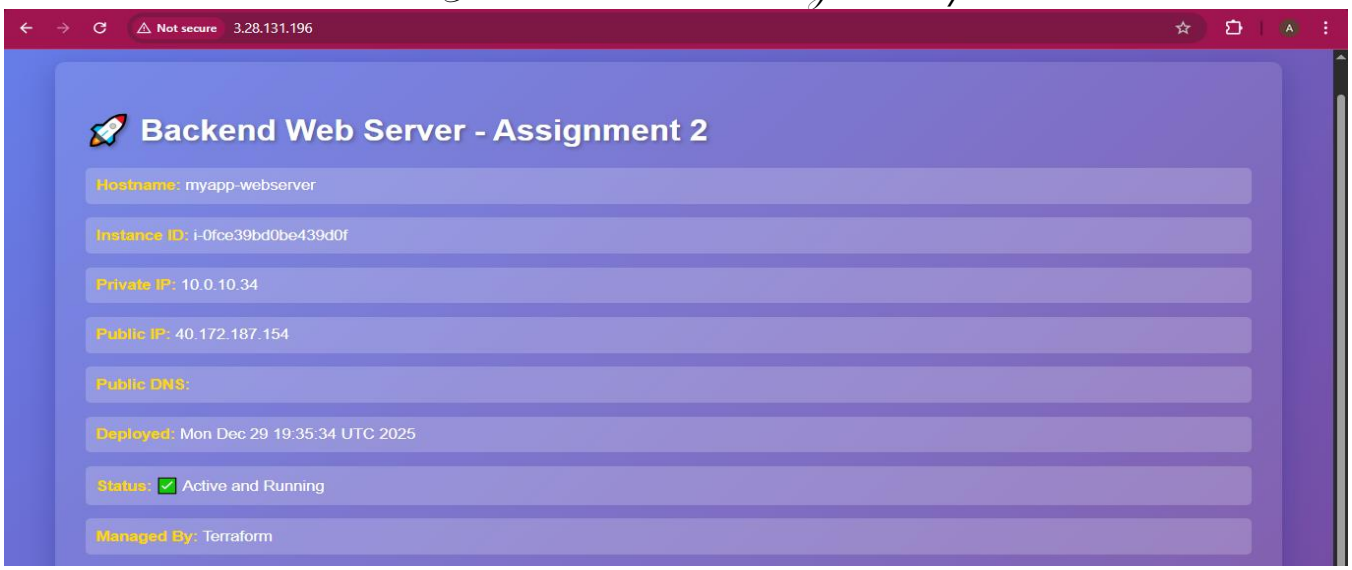
```
[ec2-user@myapp-webserver ~]$ sudo nginx -t
nginx: [warn] conflicting server name "_" on 0.0.0.0:80, ignored
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@myapp-webserver ~]$
```

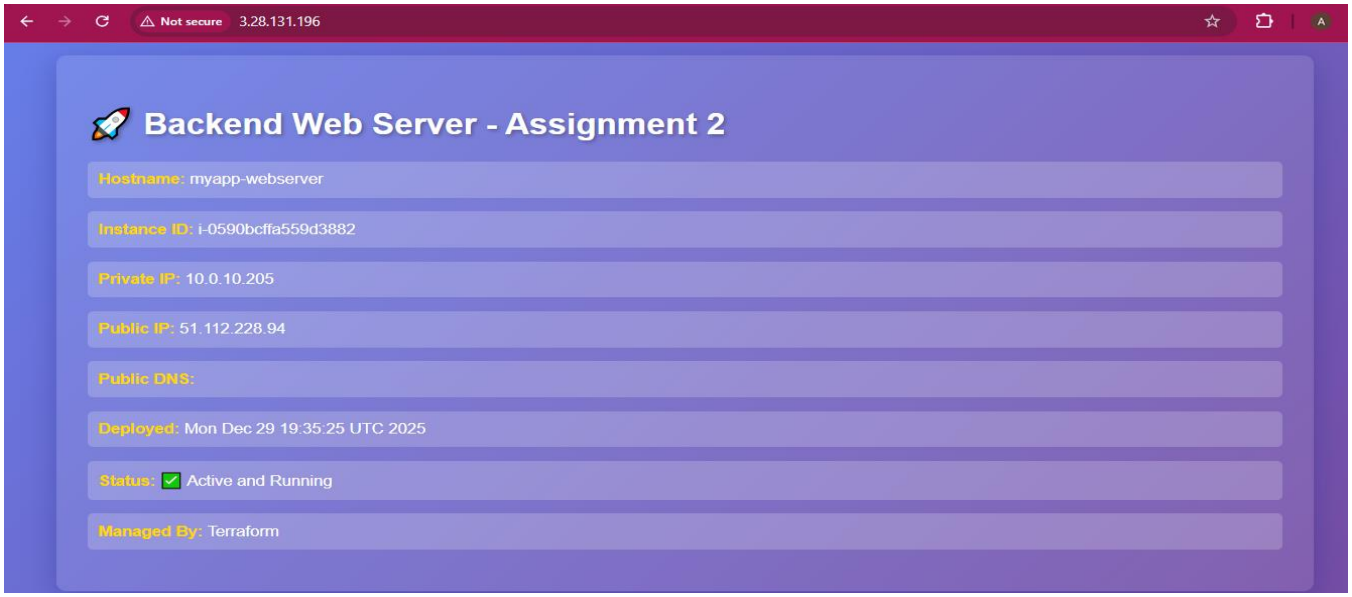
```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo systemctl restart nginx
[ec2-user@myapp-webserver ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: di>
     Active: active (running) since Mon 2025-12-29 20:48:55 UTC; 4s ago
    Process: 70229 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, sta>
    Process: 70230 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCE>
    Process: 70231 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 70232 (nginx)
      Tasks: 3 (limit: 1065)
     Memory: 3.2M
        CPU: 56ms
     CGroup: /system.slice/nginx.service
             ├─70232 "nginx: master process /usr/sbin/nginx"
             ├─70233 "nginx: worker process"
             └─70234 "nginx: worker process"

Dec 29 20:48:54 myapp-webserver systemd[1]: Starting nginx.service - The nginx >
Dec 29 20:48:54 myapp-webserver nginx[70230]: nginx: [warn] conflicting server >
Dec 29 20:48:54 myapp-webserver nginx[70230]: nginx: the configuration file /et>
Dec 29 20:48:54 myapp-webserver nginx[70230]: nginx: configuration file /etc/ng>
Dec 29 20:48:55 myapp-webserver nginx[70231]: nginx: [warn] conflicting server >
```

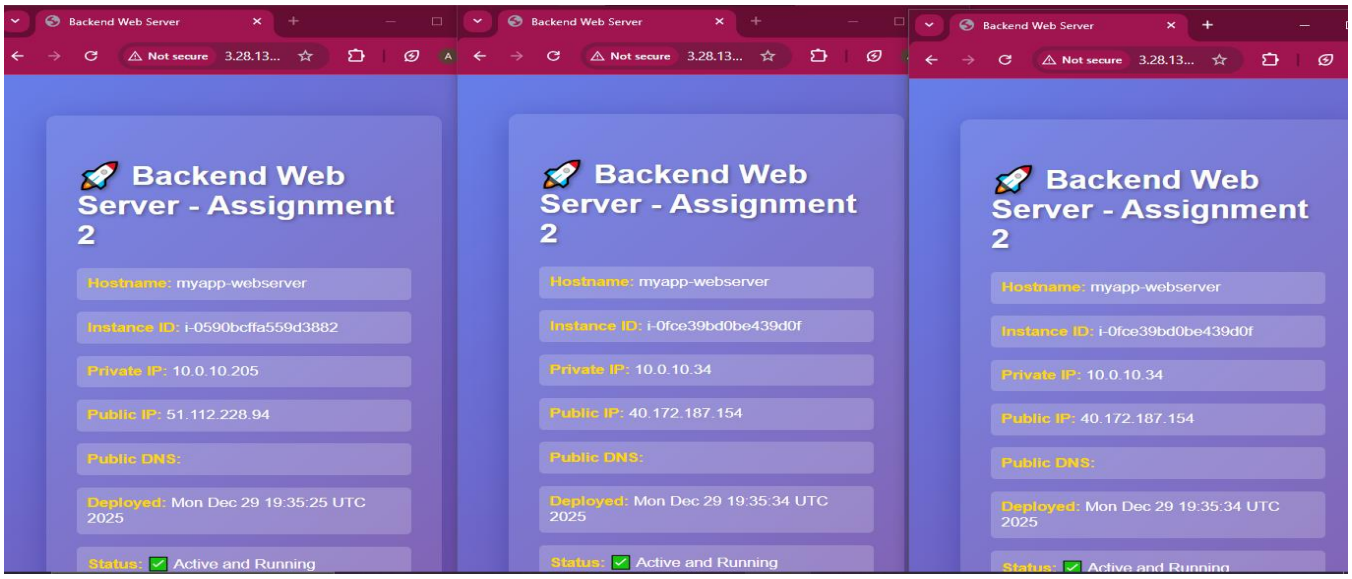
3.5.2 Test Load Balancing




```
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $ terraform output
Backend Servers:
- web-1: 40.172.187.154 (private: 10.0.10.34)
    - web-2: 51.112.228.94 (private: 10.0.10.205)
    - web-3: 40.172.100.235 (private: 10.0.10.10)
=====================================

EOT
nginx_instance_id = "i-068f810807387421c"
nginx_private_ip = "10.0.10.162"
nginx_public_ip = "3.28.131.196"
subnet_id = "subnet-0bb3e6cd3dade5025"
vpc_id = "vpc-069479242b0eb280b"
@23-22411-013-sys ➜/workspaces/Assignment2 (main) $
```

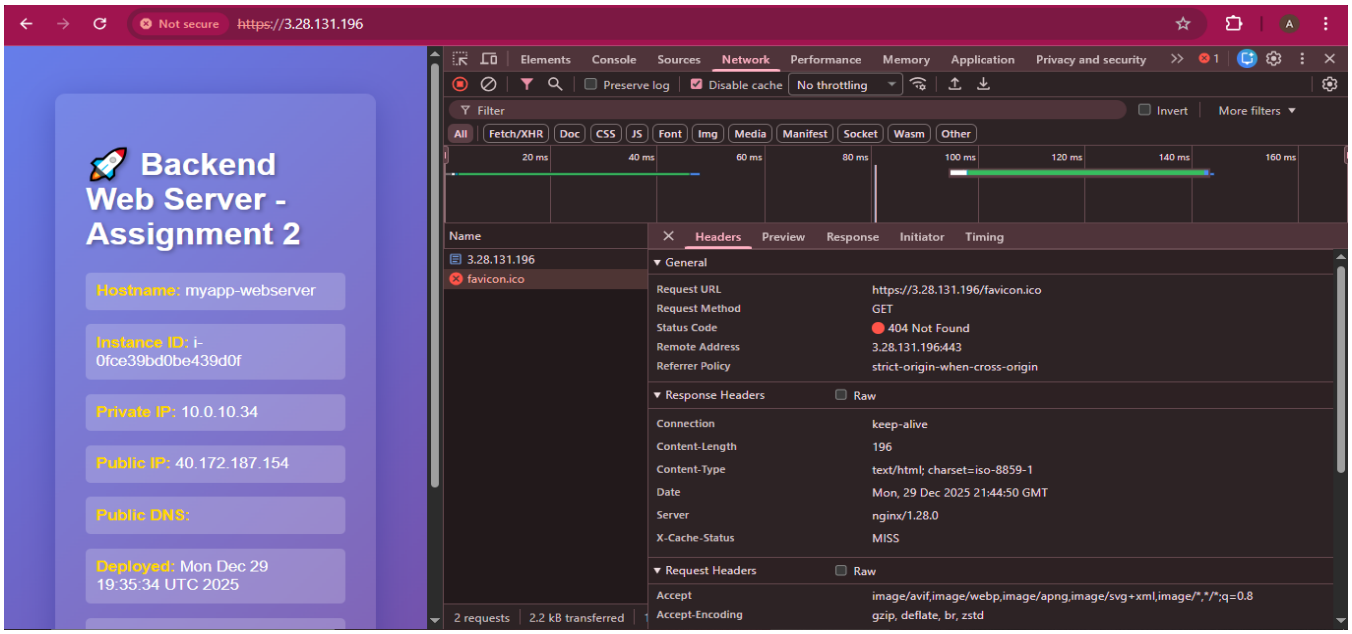

Not secure 3.28.131.196
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-0fce39bd0be439d0f
Private IP: 10.0.10.34
Public IP: 40.172.187.154
Public DNS:
Deployed: Mon Dec 29 19:35:34 UTC 2025
Status: Active and Running
Managed By: Terraform

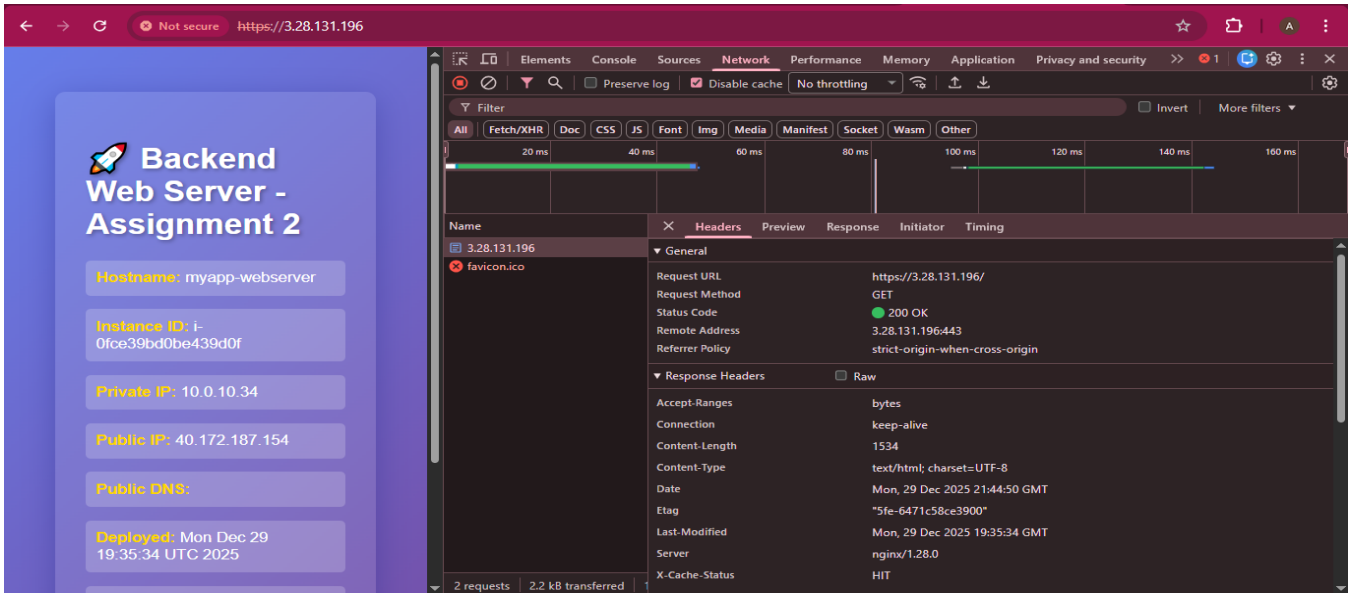

Not secure 3.28.131.196
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-0590bcffa559d3882
Private IP: 10.0.10.205
Public IP: 51.112.228.94
Public DNS:
Deployed: Mon Dec 29 19:35:25 UTC 2025
Status: Active and Running
Managed By: Terraform


Backend Web Server
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-0590bcffa559d3882
Private IP: 10.0.10.205
Public IP: 51.112.228.94
Public DNS:
Deployed: Mon Dec 29 19:35:25 UTC 2025
Status: Active and Running
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-0fce39bd0be439d0f
Private IP: 10.0.10.34
Public IP: 40.172.187.154
Public DNS:
Deployed: Mon Dec 29 19:35:34 UTC 2025
Status: Active and Running
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-0fce39bd0be439d0f
Private IP: 10.0.10.34
Public IP: 40.172.187.154
Public DNS:
Deployed: Mon Dec 29 19:35:34 UTC 2025

The successful display of backend server metadata, including hostname, instance ID, and private IP address, confirms that the Nginx reverse proxy and upstream load-balancing configuration are functioning correctly. Repeated page refreshes return responses from different backend instances, demonstrating effective request distribution among the primary backend servers using a round-robin load-balancing strategy. The third backend server is configured as a backup node within the Nginx upstream block. According to Nginx load-balancing behavior, backup servers do not receive traffic during normal operation and are only utilized when all primary backend servers become unavailable. As a result, the backup server's IP address does not appear during normal page reloads, confirming correct backup server configuration.

3.5.3 Test Cache Functionality

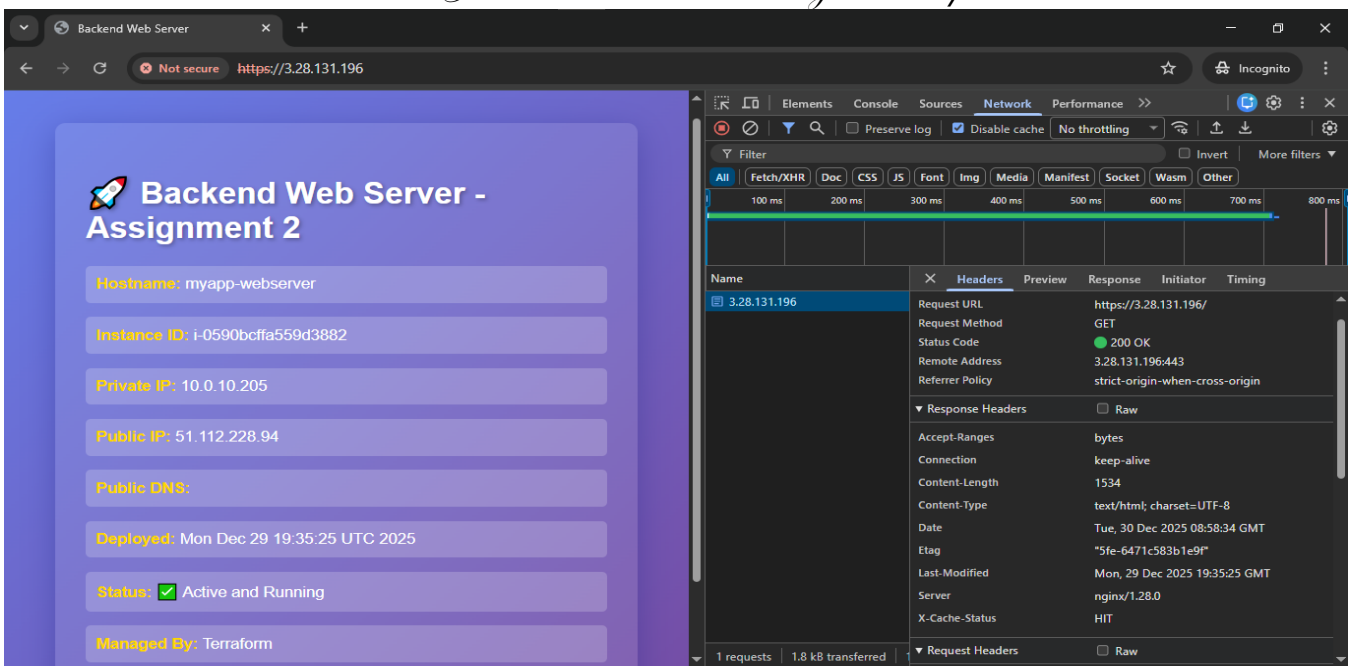




```
[ec2-user@myapp-webserver ~]$ ls -la /var/cache/nginx/
total 0
drwxr-xr-x.  3 nginx nginx  15 Dec 29 21:58 .
drwxr-xr-x. 10 root  root  107 Dec 29 21:38 ..
drwx------.  3 nginx nginx  16 Dec 29 21:58 f
[ec2-user@myapp-webserver ~]$
```

```
[ec2-user@myapp-webserver ~]$ sudo tail -f /var/log/nginx/access.log
141.98.11.140 - - [30/Dec/2025:08:00:33 +0000] "GET / HTTP/1.1" 301 169 "-" "-"
"-"
93.174.93.12 - - [30/Dec/2025:08:06:30 +0000] "\x16\x03\x02\x01o\x01\x00\x01k\x0
3\x02RH\xC5\x1A#\xF7:N\xDF\xE2\xB4\x82/\xFF\x09T\x9F\xA7\xC4y\xB0h\xC6\x13\x8C\x
A4\x1C=\x22\xE1\x1A\x98 \x84\xB4,\x85\xAFn\xE3Y\xBBbhl\xFF(=':\xA9\x82\xD9o\xC8\
xA2\xD7\x93\x98\xB4\xEF\x80\xE5\xB9\x90\x00(\xC0" 400 157 "-" "-" "-"
198.235.24.96 - - [30/Dec/2025:08:13:36 +0000] "\x16\x03\x01\x00\xCA\x01\x00\x00
\xC6\x03\x03\x9A\x92X\x86\x04\xE0\xAE\xC8~A\x9B\x97t\xDA" 400 157 "-" "-" "-"
198.235.24.96 - - [30/Dec/2025:08:13:36 +0000] "\x16\x03\x01\x00\xEE\x01\x00\x00
\xEA\x03\x03T\xF5\xC7\xB3\xBDPd\xF2t#`\x17\x86\xEA\xDC\xF2\xA6\x84n{\xB3f:\x0C\x
06\xB51\x8AH\x92\xC3^ \xBC\x14\xB37\x83_0'x\xBBW\xCA\xBA\xB3g\x8C\xEA\xBF\x81\xF
C?\x1A*\xEB\xBFDSV\xC7\x07\xE2\xFA\x00&\xC0+\xC0/\xC0,\xC00\xCC\xA9\xCC\xA8\xC0\
x09\xC0\x13\xC0" 400 157 "-" "-" "-"
3.134.148.59 - - [30/Dec/2025:08:20:27 +0000] "GET / HTTP/1.1" 301 169 "-" "cype
x.ai/scanning Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) Chrome/126.0.0.0 S
afari/537.36" "-"
3.134.148.59 - - [30/Dec/2025:08:24:41 +0000] "\x16\x03\x01\x00{\x01\x00\x00w\x0
3\x03\xDF\xC1Q2\x06\xF2\xF5F\xAF'\x02\xB8\xC5R\xB6\xB4\xB82\x5CR\xE7Q#\xB2W\x19\
xB4De\xEB\xFDd\x00\x00\x1A\xC0/\xC0+\xC0\x11\xC0\x07\xC0\x13\xC0\x09\xC0\x14\xC0
" 400 157 "-" "-" "-"
3.134.148.59 - - [30/Dec/2025:08:25:39 +0000] "SSH-2.0-Go" 400 157 "-" "-" "-"
3.134.148.59 - - [30/Dec/2025:08:26:03 +0000] "\x16\x03\x01\x00{\x01\x00\x00w\x0
3\x03X \xE6" 400 157 "-" "-" "-"
3.134.148.59 - - [30/Dec/2025:08:28:49 +0000] "" 400 0 "-" "-" "-"
```

```
ec2-user@myapp-webserver:~
3.134.148.59 - - [30/Dec/2025:08:28:49 +0000] "" 400 0 "-" "-" "-"
40.119.24.130 - - [30/Dec/2025:08:31:07 +0000] "GET /owa/auth/logon.aspx HTTP/1.
1" 404 196 "-" "Mozilla/5.0 zgrab/0.x" Cache:MISS

165.232.107.247 - - [30/Dec/2025:08:49:20 +0000] "GET /+CSCOL+/Java.jar HTTP/1.1
" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHT
ML, like Gecko) Chrome/108.0.0.0 Safari/537.36" Cache:MISS
165.232.107.247 - - [30/Dec/2025:08:49:21 +0000] "GET /+CSCOE+/logon_forms.js HT
TP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.3
6 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36" Cache:MISS
165.232.107.247 - - [30/Dec/2025:08:49:22 +0000] "GET /+CSCOL+/a1.jar HTTP/1.1"
404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
, like Gecko) Chrome/108.0.0.0 Safari/537.36" Cache:MISS
165.232.107.247 - - [30/Dec/2025:08:49:23 +0000] "GET /+CSCOE+/transfer.js HTTP/
1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36" Cache:MISS
149.40.209.201 - - [30/Dec/2025:08:49:50 +0000] "GET / HTTP/1.1" 200 1534 "-" "M
ozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36" Cache:MISS
149.40.209.201 - - [30/Dec/2025:08:49:52 +0000] "GET /favicon.ico HTTP/1.1" 404
196 "https://3.28.131.196/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebK
it/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36" Cache:MISS
149.40.209.201 - - [30/Dec/2025:08:50:01 +0000] "GET / HTTP/1.1" 200 1534 "-" "M
ozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/143.0.0.0 Safari/537.36" Cache:HIT
^C
[ec2-user@myapp-webserver ~]$
```



Cache MISS

```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo grep "Cache:" /var/log/nginx/access.log | tail -20
40.119.24.130 - - [30/Dec/2025:08:31:07 +0000] "GET /owa/auth/logon.aspx HTTP/1.1" 404 196 "-" "
Mozilla/5.0 zgrab/0.x" Cache:MISS
```

Cache HIT

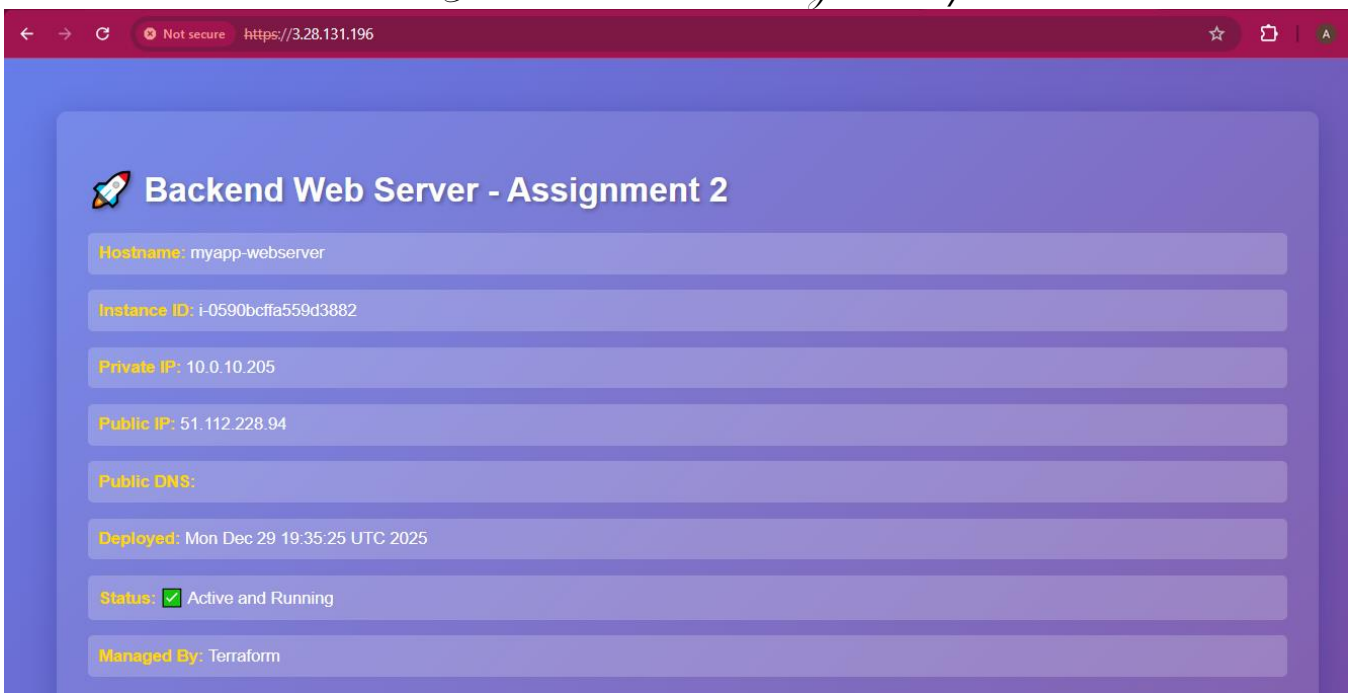
```
149.40.209.201 - - [30/Dec/2025:08:50:01 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
204.76.203.87 - - [30/Dec/2025:08:54:54 +0000] "CONNECT www.roblox.com:443 HTTP/1.1" 400 157 "-"
 "-" Cache:-
```

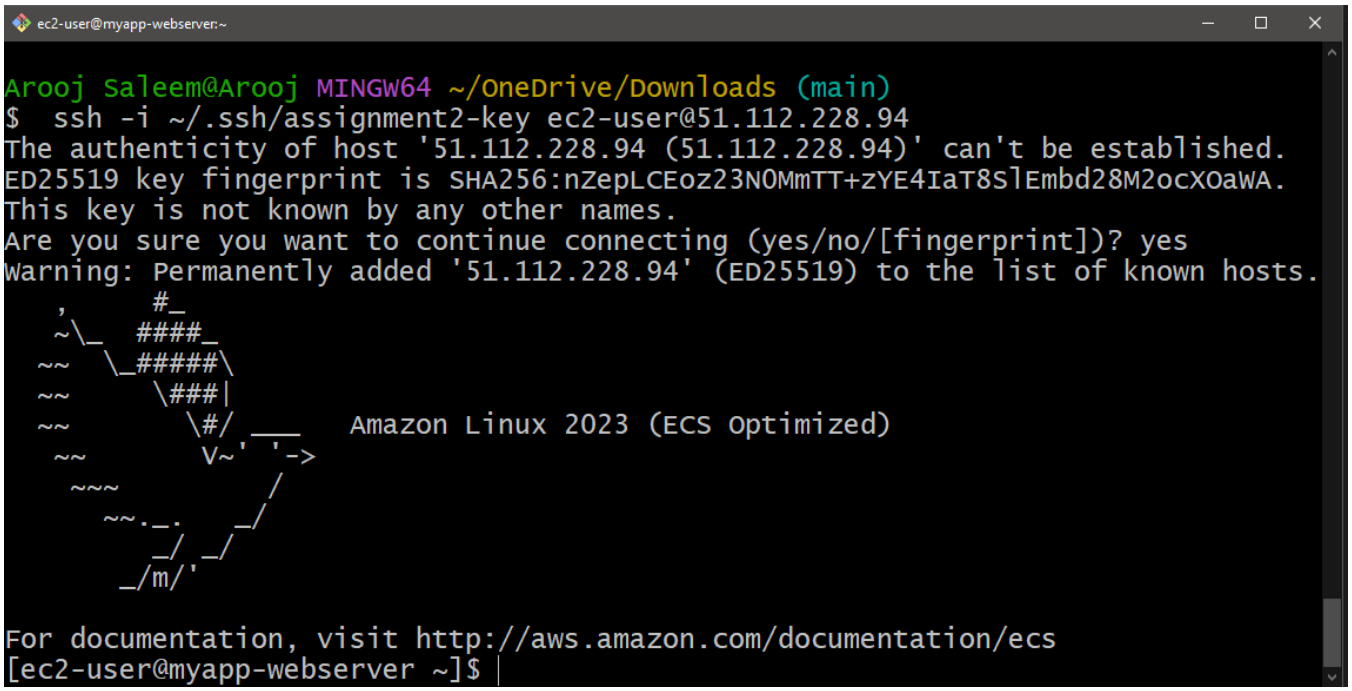
```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo grep "Cache:" /var/log/nginx/access.log | tail -20
40.119.24.130 - - [30/Dec/2025:08:31:07 +0000] "GET /owa/auth/logon.aspx HTTP/1.1" 404 196 "-" "
Mozilla/5.0 zgrab/0.x" Cache:MISS
165.232.107.247 - - [30/Dec/2025:08:49:20 +0000] "GET /+CSCOL+/Java.jar HTTP/1.1" 404 196 "-" "M
ozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0
 Safari/537.36" Cache:MISS
165.232.107.247 - - [30/Dec/2025:08:49:21 +0000] "GET /+CSCOE+/logon_forms.js HTTP/1.1" 404 196
"-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108
.0.0.0 Safari/537.36" Cache:MISS
165.232.107.247 - - [30/Dec/2025:08:49:22 +0000] "GET /+CSCOL+/a1.jar HTTP/1.1" 404 196 "-" "Moz
illa/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 S
afari/537.36" Cache:MISS
165.232.107.247 - - [30/Dec/2025:08:49:23 +0000] "GET /+CSCOE+/transfer.js HTTP/1.1" 404 196 "-"
 "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.
0.0 Safari/537.36" Cache:MISS
149.40.209.201 - - [30/Dec/2025:08:49:50 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:MISS
149.40.209.201 - - [30/Dec/2025:08:49:52 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "https://3.2
8.131.196/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Ch
rome/143.0.0.0 Safari/537.36" Cache:MISS
149.40.209.201 - - [30/Dec/2025:08:50:01 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
204.76.203.87 - - [30/Dec/2025:08:54:54 +0000] "CONNECT www.roblox.com:443 HTTP/1.1" 400 157 "-"
 "-" Cache:-
103.203.57.3 - - [30/Dec/2025:08:57:50 +0000] "GET / HTTP/1.1" 301 169 "-" "Mozilla/5.0 zgrab/0.
```

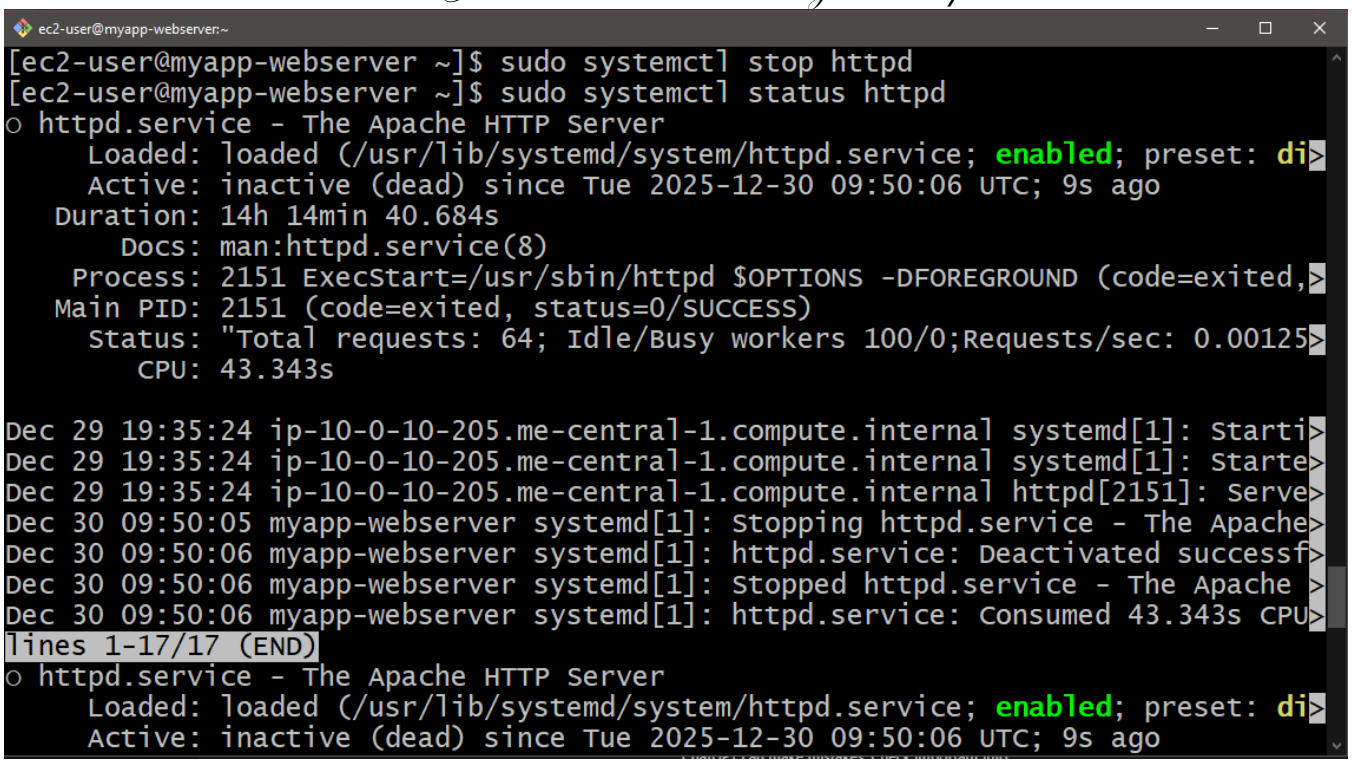
3.5.4 Test High Availability (Backup Server)

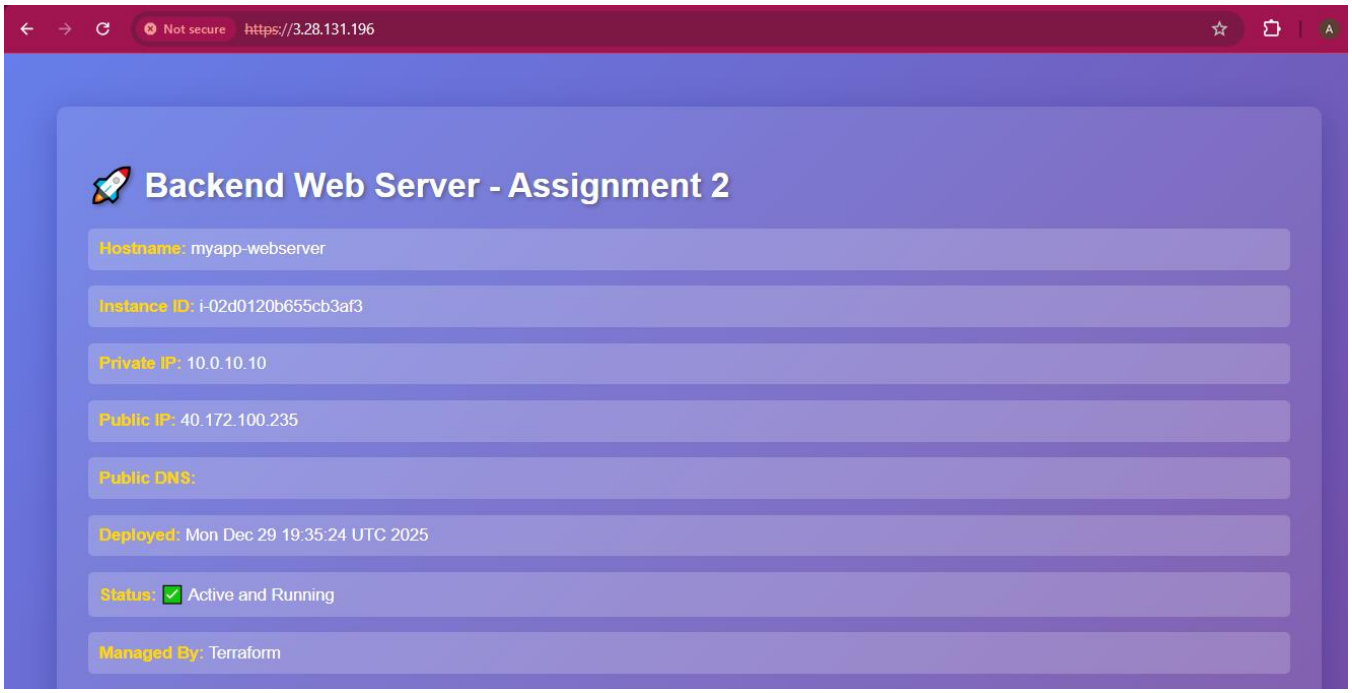
```
ec2-user@myapp-webserver:~

Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$  ssh -i ~/.ssh/assignment2-key ec2-user@40.172.187.154
The authenticity of host '40.172.187.154 (40.172.187.154)' can't be established.
ED25519 key fingerprint is SHA256:z6tgyqdKpyXfUEZkn1qzOZD2roTm+TcLdLxUFdTL9mA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '40.172.187.154' (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:~
[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: di>
     Active: inactive (dead) since Tue 2025-12-30 09:40:57 UTC; 12s ago
   Duration: 14h 5min 21.579s
       Docs: man:httpd.service(8)
lines 1-5...skipping...
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di>
     Active: inactive (dead) since Tue 2025-12-30 09:40:57 UTC; 12s ago
   Duration: 14h 5min 21.579s
       Docs: man:httpd.service(8)
    Process: 2160 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited,>
   Main PID: 2160 (code=exited, status=0/SUCCESS)
     Status: "Total requests: 70; Idle/Busy workers 100/0;Requests/sec: 0.00138>
        CPU: 47.496s

Dec 29 19:35:34 ip-10-0-10-34.me-central-1.compute.internal systemd[1]: Startin>
Dec 29 19:35:34 ip-10-0-10-34.me-central-1.compute.internal systemd[1]: Started>
Dec 29 19:35:34 ip-10-0-10-34.me-central-1.compute.internal httpd[2160]: Server>
lines 1-13...skipping...
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di>
     Active: inactive (dead) since Tue 2025-12-30 09:40:57 UTC; 12s ago
   Duration: 14h 5min 21.579s
       Docs: man:httpd.service(8)
```


Not secure
https://3.28.131.196
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-0590bcffa559d3882
Private IP: 10.0.10.205
Public IP: 51.112.228.94
Public DNS:
Deployed: Mon Dec 29 19:35:25 UTC 2025
Status: Active and Running
Managed By: Terraform


ec2-user@myapp-webserver:~
Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$ ssh -i ~/.ssh/assignment2-key ec2-user@51.112.228.94
The authenticity of host '51.112.228.94 (51.112.228.94)' can't be established.
ED25519 key fingerprint is SHA256:nZepLCEoz23NOMmTT+zYE4IaT8SlEmbd28M2ocXOaWA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '51.112.228.94' (ED25519) to the list of known hosts.
Amazon Linux 2023 (ECS Optimized)
For documentation, visit http://aws.amazon.com/documentation/ecs
[ec2-user@myapp-webserver ~]$


ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo systemctl stop httpd
[ec2-user@myapp-webserver ~]$ sudo systemctl status httpd
○ httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di
Active: inactive (dead) since Tue 2025-12-30 09:50:06 UTC; 9s ago
Duration: 14h 14min 40.684s
Docs: man:httpd.service(8)
Process: 2151 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited,
Main PID: 2151 (code=exited, status=0/SUCCESS)
Status: "Total requests: 64; Idle/Busy workers 100/0;Requests/sec: 0.00125
CPU: 43.343s
Dec 29 19:35:24 ip-10-0-10-205.me-central-1.compute.internal systemd[1]: Starti
Dec 29 19:35:24 ip-10-0-10-205.me-central-1.compute.internal systemd[1]: Starte
Dec 29 19:35:24 ip-10-0-10-205.me-central-1.compute.internal httpd[2151]: Serve
Dec 30 09:50:05 myapp-webserver systemd[1]: Stopping httpd.service - The Apache
Dec 30 09:50:06 myapp-webserver systemd[1]: httpd.service: Deactivated successf
Dec 30 09:50:06 myapp-webserver systemd[1]: Stopped httpd.service - The Apache
Dec 30 09:50:06 myapp-webserver systemd[1]: httpd.service: Consumed 43.343s CPU
lines 1-17/17 (END)
○ httpd.service - The Apache HTTP Server
Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di
Active: inactive (dead) since Tue 2025-12-30 09:50:06 UTC; 9s ago

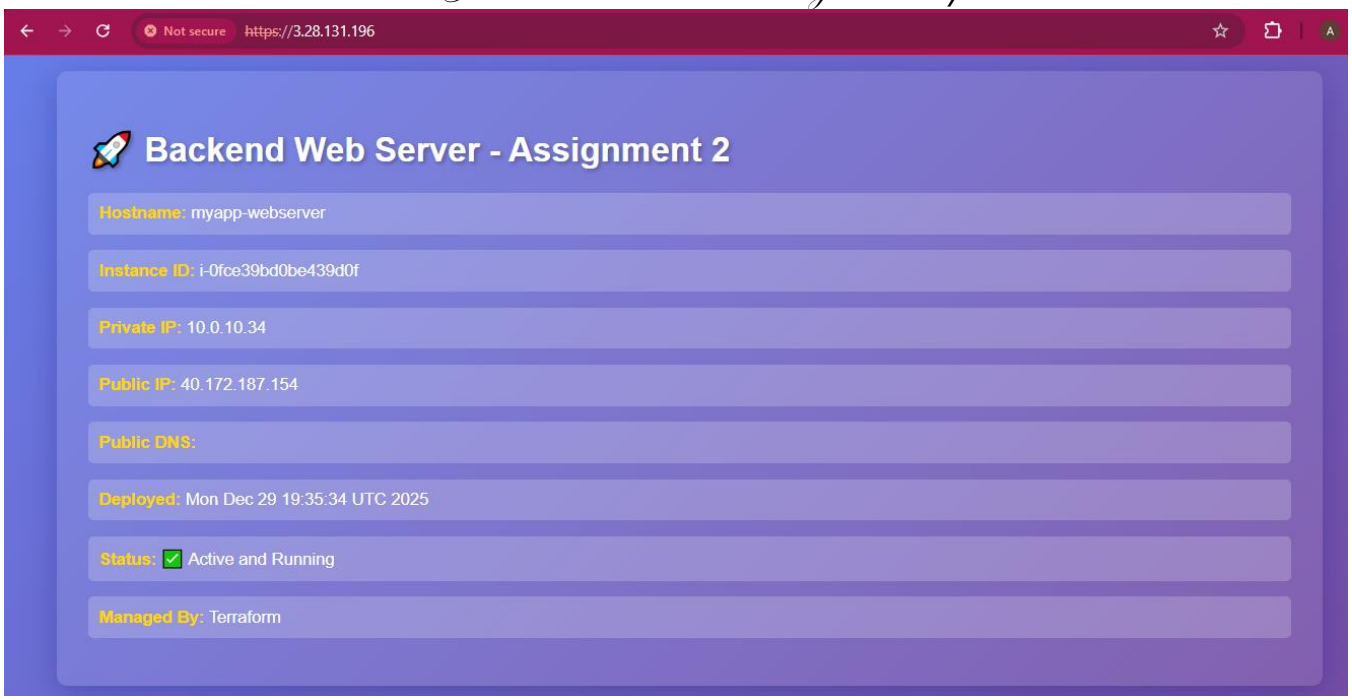

Not secure
https://3.28.131.196
Backend Web Server - Assignment 2
Hostname: myapp-webserver
Instance ID: i-02d0120b655cb3af3
Private IP: 10.0.10.10
Public IP: 40.172.100.235
Public DNS:
Deployed: Mon Dec 29 19:35:24 UTC 2025
Status: Active and Running
Managed By: Terraform

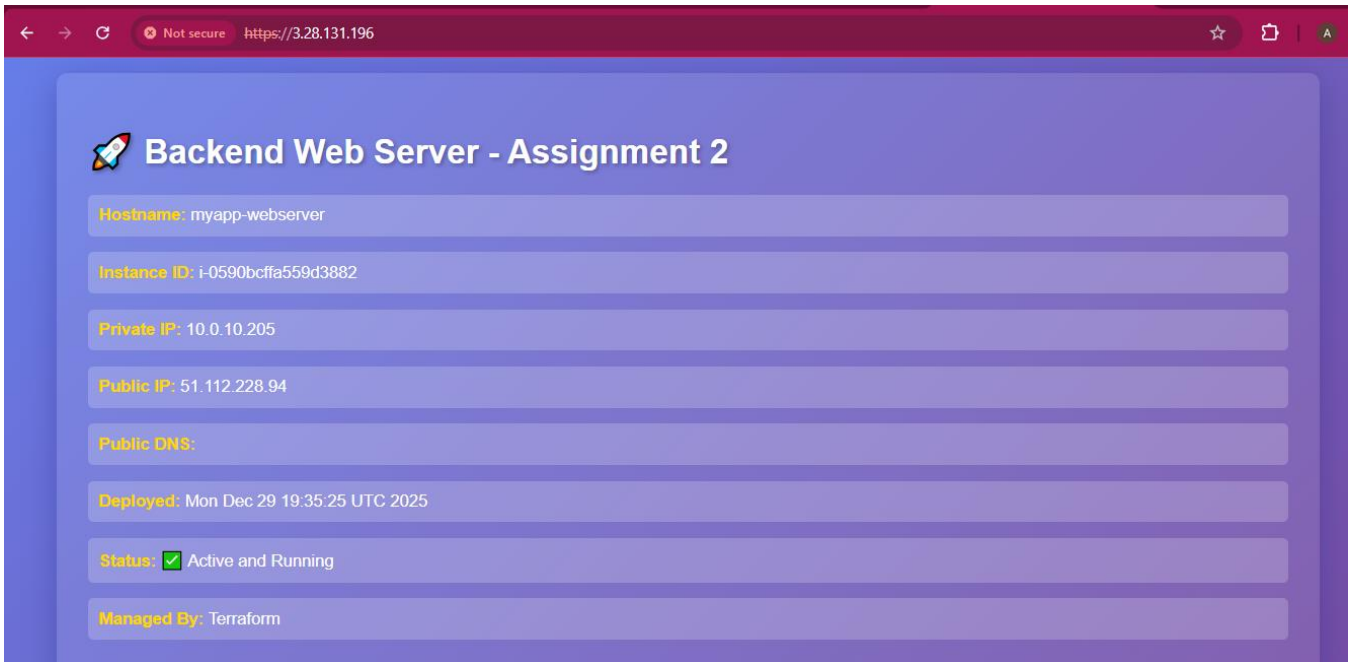
```
ec2-user@myapp-webserver:~
Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$  ssh -i ~/.ssh/assignment2-key ec2-user@3.28.131.196
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Tue Dec 30 09:27:10 2025 from 149.40.209.201
```

```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo tail -f /var/log/nginx/error.log
2025/12/30 08:48:01 [notice] 115021#115021: signal 17 (SIGCHLD) received from 70
4522
2025/12/30 08:48:01 [notice] 115021#115021: signal 17 (SIGCHLD) received from 70
4521
2025/12/30 08:48:01 [notice] 115021#115021: worker process 704521 exited with co
de 0
2025/12/30 08:48:01 [notice] 115021#115021: signal 29 (SIGIO) received
2025/12/30 08:48:42 [crit] 712750#712750: unlink() "/var/cache/nginx/f/ea/4f8a18
cefd3be6707aadd63f30185eaf" failed (2: No such file or directory)
2025/12/30 09:01:53 [crit] 712748#712748: *347 SSL_do_handshake() failed (SSL: e
rror:0A000172:SSL routines::wrong signature type) while SSL handshaking, client:
 50.116.56.80, server: 0.0.0.0:443
2025/12/30 09:55:42 [error] 712749#712749: *397 connect() failed (111: Connectio
n refused) while connecting to upstream, client: 149.40.209.201, server: _, requ
est: "GET / HTTP/1.1", upstream: "http://10.0.10.34:80/", host: "3.28.131.196"
2025/12/30 09:55:42 [warn] 712749#712749: *397 upstream server temporarily disab
led while connecting to upstream, client: 149.40.209.201, server: _, request: "G
ET / HTTP/1.1", upstream: "http://10.0.10.34:80/", host: "3.28.131.196"
2025/12/30 09:55:42 [error] 712749#712749: *397 connect() failed (111: Connectio
n refused) while connecting to upstream, client: 149.40.209.201, server: _, requ
est: "GET / HTTP/1.1", upstream: "http://10.0.10.205:80/", host: "3.28.131.196"
2025/12/30 09:55:42 [warn] 712749#712749: *397 upstream server temporarily disab
led while connecting to upstream, client: 149.40.209.201, server: _, request: "G
ET / HTTP/1.1", upstream: "http://10.0.10.205:80/", host: "3.28.131.196"
^C
[ec2-user@myapp-webserver ~]$
```

```
MINGW64:/c/Users/ALAM-PC/OneDrive/Downloads
Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$ ssh -i ~/.ssh/assignment2-key ec2-user@40.172.187.154
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Tue Dec 30 09:35:41 2025 from 149.40.209.201
[ec2-user@myapp-webserver ~]$ sudo systemctl start httpd
[ec2-user@myapp-webserver ~]$ exit
logout
Connection to 40.172.187.154 closed.
```

```
MINGW64:/c/Users/ALAM-PC/OneDrive/Downloads

Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$ ssh -i ~/.ssh/assignment2-key ec2-user@51.112.228.94
   ,     #_
   ~\_  ####_
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   Amazon Linux 2023 (ECS Optimized)
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

For documentation, visit http://aws.amazon.com/documentation/ecs
Last login: Tue Dec 30 09:46:58 2025 from 149.40.209.201
[ec2-user@myapp-webserver ~]$ sudo systemctl start httpd
[ec2-user@myapp-webserver ~]$ exit
logout
Connection to 51.112.228.94 closed.

Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
```





3.5.5 Security & Performance Analysis

```
MINGW64:/c/Users/ALAM-PC/OneDrive/Downloads
Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$ openssl s_client -connect 3.28.131.196:443 -showcerts
Connecting to 3.28.131.196
CONNECTED(00000138)
Can't use SSL_get_servername
depth=0 C=PK, ST=Punjab, L=Gujarkhan, O=Default Company Ltd, CN=Arooj
verify error:num=18:self-signed certificate
verify return:1
depth=0 C=PK, ST=Punjab, L=Gujarkhan, O=Default Company Ltd, CN=Arooj
verify return:1
---
Certificate chain
 0 s:C=PK, ST=Punjab, L=Gujarkhan, O=Default Company Ltd, CN=Arooj
   i:C=PK, ST=Punjab, L=Gujarkhan, O=Default Company Ltd, CN=Arooj
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Dec 29 21:38:32 2025 GMT; NotAfter: Dec 29 21:38:32 2026 GMT
-----BEGIN CERTIFICATE-----
MIIDoTCCAomgAwIBAgIUPgrSfelPcrd8wU1sAyFM8x9nxkYwDQYJKoZIhvcNAQEL
BQAwYDELMAkGA1UEBhMCUExDzANBgNVBAgMBlB1bmphYjESMBAGA1UEBwwJR3Vq
YXJraGFuMRwwGgYDVQQKDBNEZWZhdWx0IENvbXBhbnkgTHRkMQ4wDAYDVQQDDAVB
cm9vajAeFw0yNTEyMjkyMTM4MzJaFw0yNjEyMjkyMTM4MzJaMGAxCzAJBgNVBAYT
AlBLMQ8wDQYDVQQIDAZQdW5qYWIxEjAQBgNVBAcMCUd1amFya2hhbjEcMBoGA1UE
CgwTRGVmYXVsdCBDb21wYW55IEx0ZDEOMAwGA1UEAwwFQXJvb2owggEiMA0GCSqG
SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCLONIhD4yuGlD/pPTTMkEHb59TC7yK2Cuz
B+eFdzmuhde5Is2QvXu3tlPlOnRA3+3SoSh1VxiTsrwmEnXaBglpHW6ZSsrtoxJi
ufscepw20aIEK8HPC/Df9KkwDsW6MZ8Ea33JIJH1dmf2C+hPQerUcbYptT1+3XGn
HKvc8kv0/xOwCaaA8kkMu+WIy7PZgrCIbx83WE30DERy4T4bOWDvSOPdx8USR0AA
```

```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo openssl x509 -in /etc/nginx/ssl/nginx.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            3e:0a:d2:7d:e9:4f:72:b7:7c:c1:4d:6c:03:21:4c:f3:1f:67:c6:46
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=PK, ST=Punjab, L=Gujarkhan, O=Default Company Ltd, CN=Arooj
        Validity
            Not Before: Dec 29 21:38:32 2025 GMT
            Not After : Dec 29 21:38:32 2026 GMT
        Subject: C=PK, ST=Punjab, L=Gujarkhan, O=Default Company Ltd, CN=Arooj
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:8b:38:d2:21:0f:8c:ae:1a:50:ff:a4:f4:d3:32:
                    41:07:6f:9f:53:0b:bc:8a:d8:2b:b3:07:e7:85:77:
                    39:ae:85:d7:b9:22:cd:90:bd:7b:b7:b6:53:e5:3a:
                    74:40:df:ed:d2:a1:28:75:57:18:93:b2:bc:26:12:
                    75:da:06:09:69:1d:6e:99:4a:ca:ed:a3:12:62:b9:
                    fb:1c:7a:9c:36:d1:a2:04:2b:c1:cf:0b:f0:df:f4:
                    a9:30:0e:c5:ba:31:9f:04:6b:7d:c9:20:91:f5:76:
                    67:f6:0b:e8:4f:41:ea:d4:71:b6:29:b5:3d:7e:dd:
                    71:a7:1c:ab:dc:f2:4b:f4:ff:13:b0:09:a6:80:f2:
                    49:0c:bb:e5:88:cb:b3:d9:82:b0:88:6f:1f:37:58:
                    4d:f4:0c:44:72:e1:3e:1b:39:60:ef:48:e3:dd:c7:
```

```
Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$ curl -I -k https://3.28.131.196
HTTP/1.1 200 OK
Server: nginx/1.28.0
Date: Tue, 30 Dec 2025 10:55:27 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1534
Connection: keep-alive
Last-Modified: Mon, 29 Dec 2025 19:35:34 GMT
ETag: "5fe-6471c58ce3900"
Accept-Ranges: bytes
Strict-Transport-Security: max-age=31536000; includeSubDomains
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block


Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$
```

```
Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$ curl -I http://3.28.131.196
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.0
Date: Tue, 30 Dec 2025 10:56:59 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://3.28.131.196/


Arooj Saleem@Arooj MINGW64 ~/OneDrive/Downloads (main)
$
```

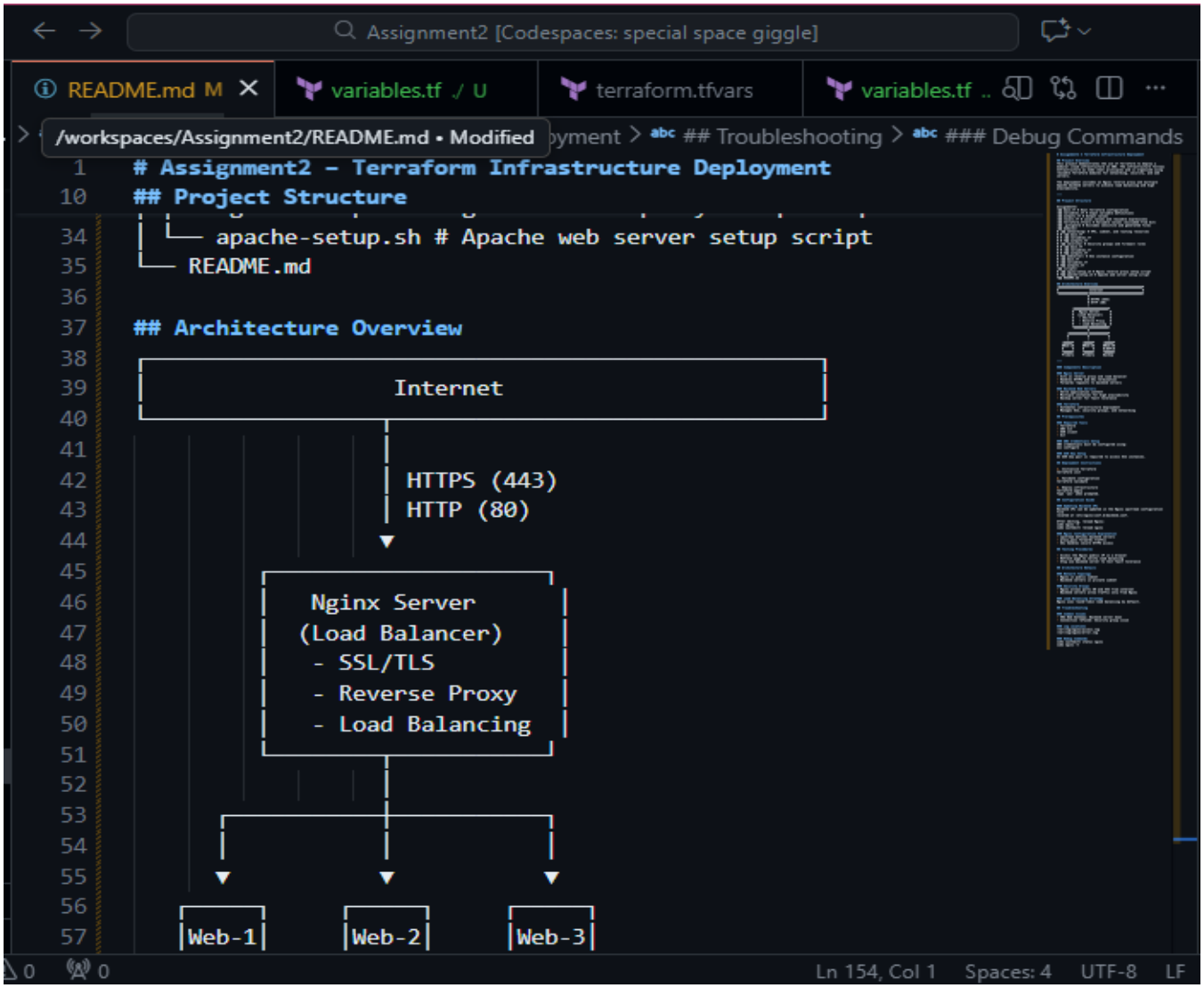
```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo tail -50 /var/log/nginx/error.log
2025/12/30 10:52:03 [warn] 823340#823340: conflicting server name "_" on 0.0.0.0
:80, ignored
2025/12/30 10:52:03 [warn] 823346#823346: conflicting server name "_" on 0.0.0.0
:80, ignored
2025/12/30 10:52:03 [notice] 823346#823346: signal process started
2025/12/30 10:52:03 [notice] 115021#115021: signal 1 (SIGHUP) received from 8233
46, reconfiguring
2025/12/30 10:52:03 [notice] 115021#115021: reconfiguring
2025/12/30 10:52:03 [warn] 115021#115021: conflicting server name "_" on 0.0.0.0
:80, ignored
2025/12/30 10:52:03 [notice] 115021#115021: using the "epoll" event method
2025/12/30 10:52:03 [notice] 115021#115021: start worker processes
2025/12/30 10:52:03 [notice] 115021#115021: start worker process 823347
2025/12/30 10:52:03 [notice] 115021#115021: start worker process 823348
2025/12/30 10:52:03 [notice] 115021#115021: start cache manager process 823349
2025/12/30 10:52:03 [notice] 809294#809294: gracefully shutting down
2025/12/30 10:52:03 [notice] 809293#809293: gracefully shutting down
2025/12/30 10:52:03 [notice] 809294#809294: exiting
2025/12/30 10:52:03 [notice] 809293#809293: exiting
2025/12/30 10:52:03 [notice] 809293#809293: exit
2025/12/30 10:52:03 [notice] 809294#809294: exit
2025/12/30 10:52:03 [notice] 809295#809295: exiting
2025/12/30 10:52:03 [notice] 115021#115021: signal 17 (SIGCHLD) received from 80
9294
2025/12/30 10:52:03 [notice] 115021#115021: worker process 809293 exited with co
de 0
```

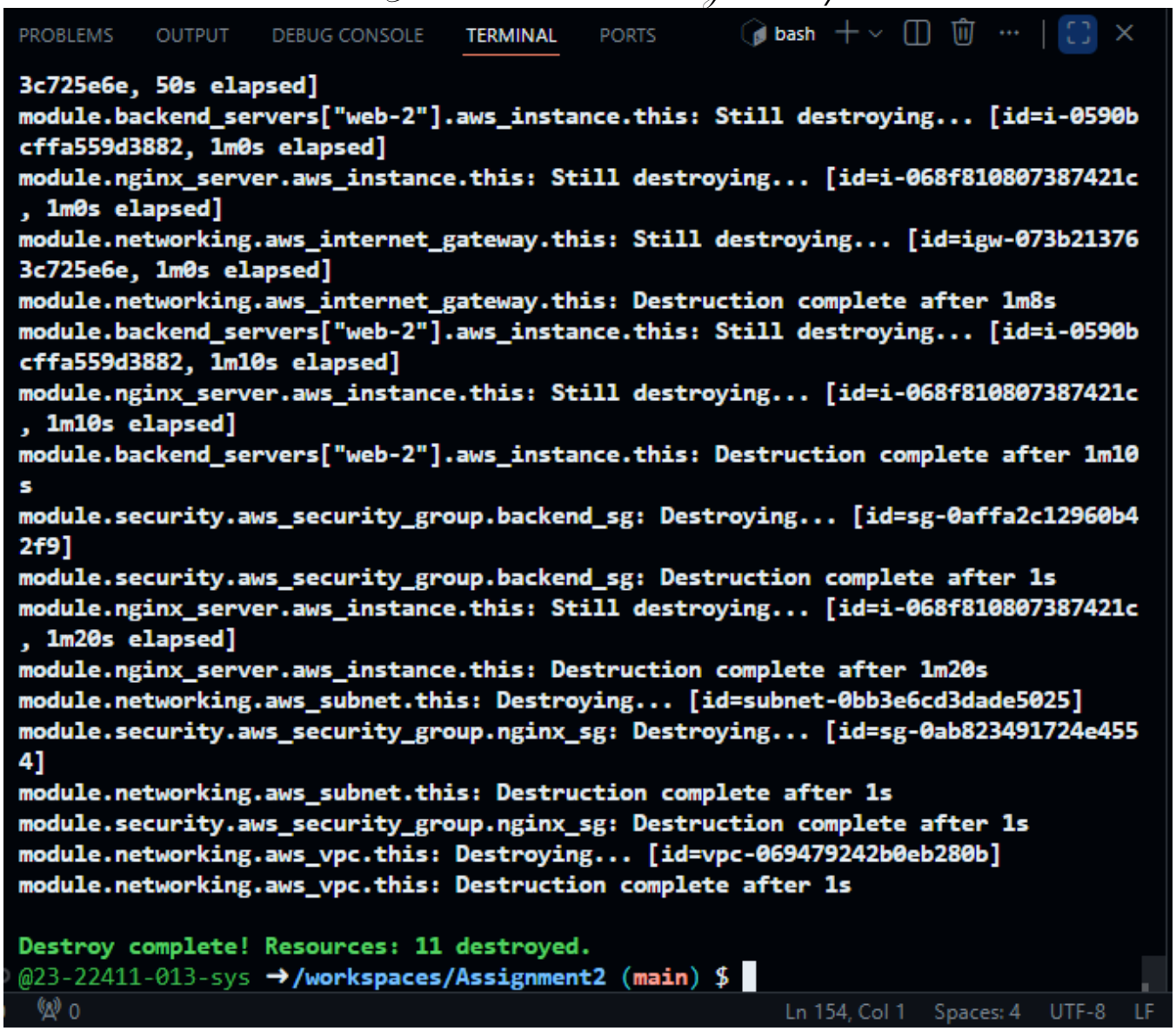
```
ec2-user@myapp-webserver:~
[ec2-user@myapp-webserver ~]$ sudo tail -50 /var/log/nginx/access.log
149.40.209.201 - - [30/Dec/2025:09:21:41 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:21:53 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:22:04 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:22:05 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:22:06 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:23:09 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:23:10 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:25:36 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
Cache:HIT
149.40.209.201 - - [30/Dec/2025:09:25:38 +0000] "GET / HTTP/1.1" 200 1534 "-" "Mozilla/5.0 (Wind
ows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36"
```

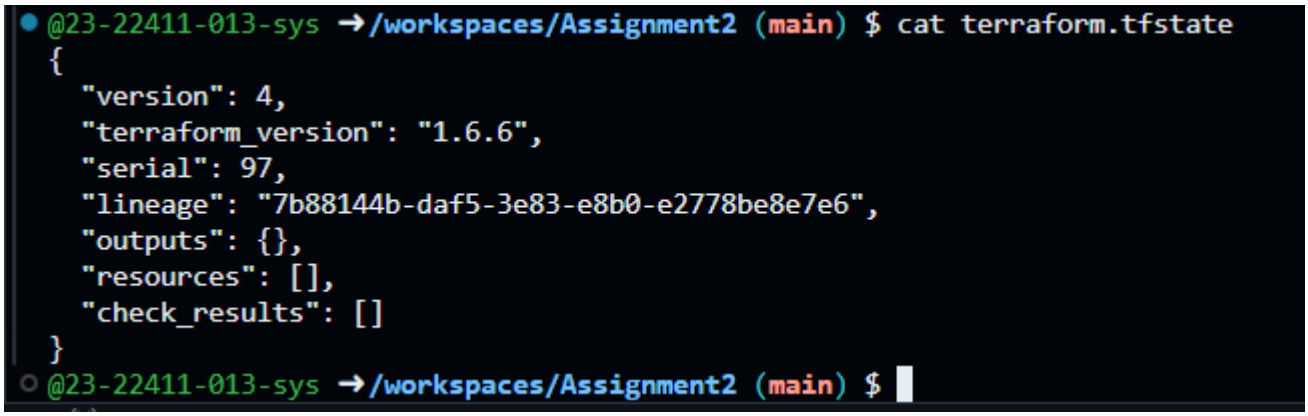
```
[ec2-user@myapp-webserver ~]$ ps aux | grep nginx
root      115021  0.0  0.8  28144  7708 ?        Ss   Dec29   0:00 nginx: master process /usr/sb
in/nginx
nginx     826094  0.0  0.6  28148  6244 ?        S    10:55   0:00 nginx: worker process
nginx     826095  0.0  0.6  28148  6312 ?        S    10:55   0:00 nginx: worker process
nginx     826096  0.0  0.4  28148  3756 ?        S    10:55   0:00 nginx: cache manager process
ec2-user  832863  0.0  0.2 222328  2120 pts/2    S+   11:02   0:00 grep --color=auto nginx
[ec2-user@myapp-webserver ~]$
```

3.6 Cleanup

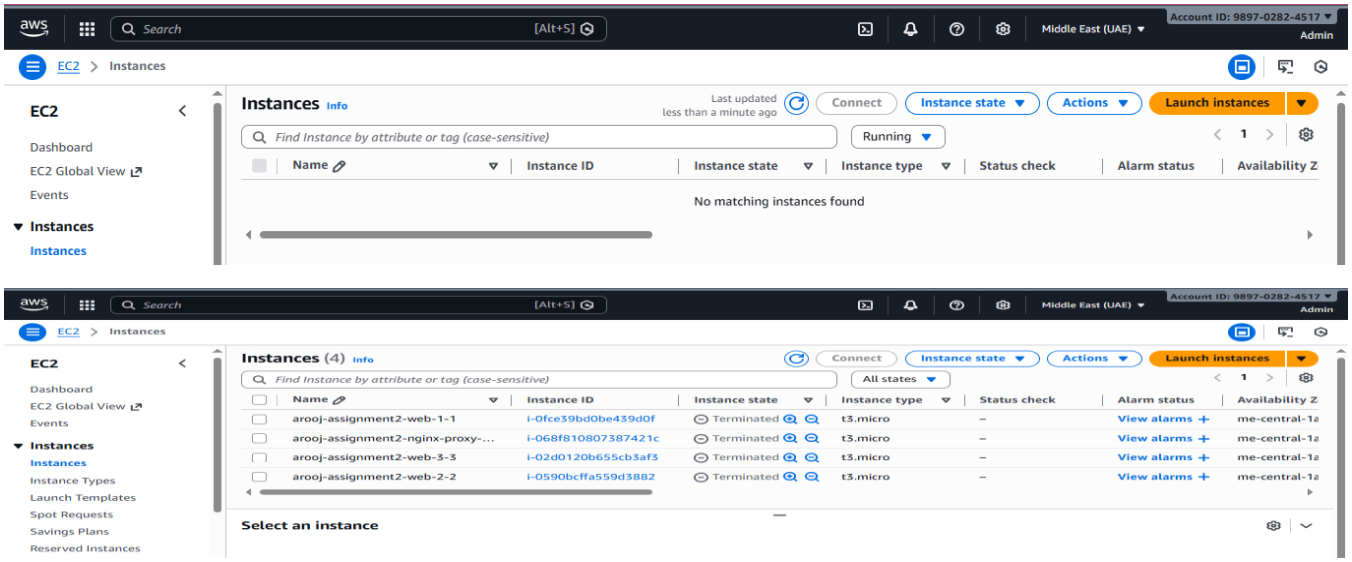
After completing all testing, the infrastructure was safely destroyed using the terraform destroy command. Resource deletion was verified through the AWS EC2 console and AWS CLI commands. The Terraform state file confirmed that no resources remained, ensuring no unnecessary AWS costs.




PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
bash
3c725e6e, 50s elapsed]
module.backend_servers["web-2"].aws_instance.this: Still destroying... [id=i-0590b
cffa559d3882, 1m0s elapsed]
module.nginx_server.aws_instance.this: Still destroying... [id=i-068f810807387421c
, 1m0s elapsed]
module.networking.aws_internet_gateway.this: Still destroying... [id=igw-073b21376
3c725e6e, 1m0s elapsed]
module.networking.aws_internet_gateway.this: Destruction complete after 1m8s
module.backend_servers["web-2"].aws_instance.this: Still destroying... [id=i-0590b
cffa559d3882, 1m10s elapsed]
module.nginx_server.aws_instance.this: Still destroying... [id=i-068f810807387421c
, 1m10s elapsed]
module.backend_servers["web-2"].aws_instance.this: Destruction complete after 1m10
s
module.security.aws_security_group.backend_sg: Destroying... [id=sg-0affa2c12960b4
2f9]
module.security.aws_security_group.backend_sg: Destruction complete after 1s
module.nginx_server.aws_instance.this: Still destroying... [id=i-068f810807387421c
, 1m20s elapsed]
module.nginx_server.aws_instance.this: Destruction complete after 1m20s
module.networking.aws_subnet.this: Destroying... [id=subnet-0bb3e6cd3dade5025]
module.security.aws_security_group.nginx_sg: Destroying... [id=sg-0ab823491724e455
4]
module.networking.aws_subnet.this: Destruction complete after 1s
module.security.aws_security_group.nginx_sg: Destruction complete after 1s
module.networking.aws_vpc.this: Destroying... [id=vpc-069479242b0eb280b]
module.networking.aws_vpc.this: Destruction complete after 1s
Destroy complete! Resources: 11 destroyed.
@23-22411-013-sys →/workspaces/Assignment2 (main) $
0
Ln 154, Col 1
Spaces: 4
UTF-8
LF


@23-22411-013-sys →/workspaces/Assignment2 (main) $ cat terraform.tfstate
{
"version": 4,
"terraform_version": "1.6.6",
"serial": 97,
"lineage": "7b88144b-daf5-3e83-e8b0-e2778be8e7e6",
"outputs": {},
"resources": [],
"check_results": []
}
@23-22411-013-sys →/workspaces/Assignment2 (main) $

```
@23-22411-013-sys →/workspaces/Assignment2 (main) $ aws ec2 describe-instances \
--filters "Name=tag:Project,Values=Assignment-2" \
          "Name=instance-state-name,Values=running,stopped" \
--query "Reservations[].Instances[].InstanceId"
[]
@23-22411-013-sys →/workspaces/Assignment2 (main) $
```



4. Testing Results

4.1 Load Balancing Tests

Load balancing results confirm that Nginx distributes requests across multiple backend servers. Different backend responses were observed during repeated access, validating correct request distribution.

4.2 Cache Performance Tests

Cache testing showed initial cache MISS responses followed by cache HIT responses for repeated requests. This confirms reduced backend load and improved response efficiency.

4.3 High Availability Tests

High availability was achieved using multiple backend servers with a backup node. The backup server remained inactive during normal operation, confirming correct failover configuration.

4.4 Security Tests

Security testing verified that backend servers are not publicly accessible. Only the Nginx reverse proxy accepts external traffic, ensuring controlled internal access.

4.5 Performance Metrics

The system demonstrated stable performance with consistent response times. Load distribution and caching improved overall request handling efficiency.

5. Challenges & Solutions

5.1 Problems Encountered

Challenges included Terraform module dependencies, backend connectivity, and load balancing verification.

5.2 Solutions Implemented

These issues were resolved using proper module outputs, refined security group rules, and instance-based response identification.

5.3 Lessons Learned

The project strengthened skills in Terraform automation, cloud security, and scalable infrastructure design.

6. Conclusion

This assignment successfully implemented a secure and scalable multi-tier web infrastructure using Terraform and AWS. Key skills developed include Infrastructure as Code, server automation, and system testing. Future improvements may include auto-scaling, monitoring, and advanced load balancing services.

7. Appendices

Appendix A: Code Listings

All Terraform and script files are available in the GitHub repository.

Appendix B: Configuration Files

Nginx and Apache configuration files are documented in the project repository.

Appendix C: Additional Screenshots

Supporting screenshots are included to demonstrate deployment, testing, and cleanup.

Appendix D: References

Terraform Documentation, AWS EC2 Documentation, Nginx Documentation.

End of Document