Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

Cloud Computing Project

Proiect: 03

Submitted to: Sir Waqas

Due Date: 26 Jan 2026

Submitted By

Sadaf Riaz (077)

Zuha Irfan (073)

Rafia Fiaz (051)

Project 3 – High Availability Web Application with Nginx Load Balancing

Table of Contents

Repo link: https://github.com/SadafRiaz077/cc_SadafRiaz_077-Project-3-HA-WebApp

Executive Summary

This project demonstrates the design and implementation of a Highly Available Web Application Infrastructure using AWS cloud services and open-source automation tools strictly within the scope of the course curriculum. The primary goal of the project is to ensure high availability, fault tolerance, secure communication, performance optimization, and automated management of a web application.

The infrastructure was provisioned using Terraform as Infrastructure as Code (IaC), enabling repeatable and consistent deployment of AWS resources. A custom Virtual Private Cloud (VPC) was created with public and private subnets distributed across multiple Availability Zones, ensuring resilience against zone-level failures. An Nginx-based Load Balancer was deployed in a public subnet to distribute incoming client traffic across multiple backend web servers hosted in private subnets.

The backend tier consists of three EC2 instances, where two servers act as primary web servers and the third server is configured as a backup server. The backup server is only utilized when both primary servers become unavailable, ensuring uninterrupted service delivery. Strict security group rules were applied so that backend servers accept HTTP traffic exclusively from the load balancer, enhancing overall system security.

In addition, Ansible played a critical role in automating server configuration and application deployment. It ensured consistent Nginx configuration across all servers, enabled SSL termination with self-signed certificates, configured content caching at the load balancer, and deployed a health monitoring script. This automation reduced manual intervention, improved deployment reliability, and made the infrastructure easy to maintain and scale for future enhancements.

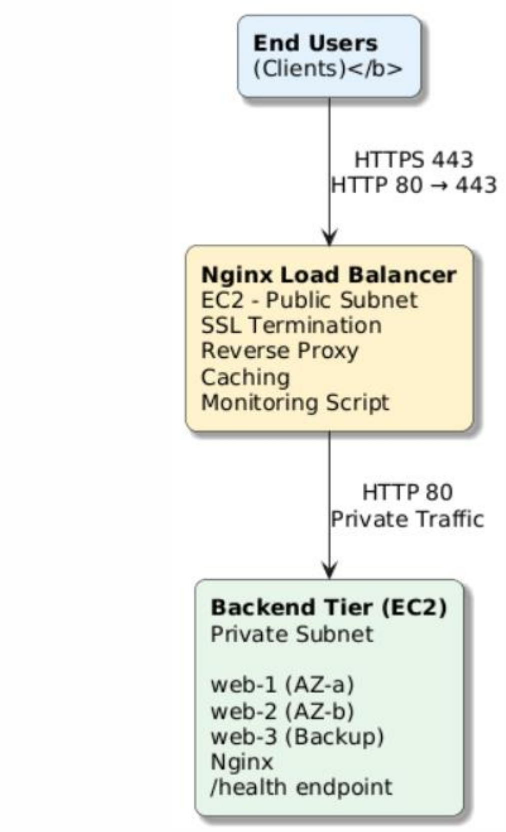
High availability was validated through controlled failover testing, caching behavior was verified using HTTP response headers, SSL functionality was tested through HTTPS redirection and certificate inspection, and monitoring was validated through log analysis. Overall, the project successfully demonstrates practical implementation of high availability principles, automation, and reliability in a cloud-based web application environment.

Architecture Design

1. Overview

The system is designed using a highly available three-tier web architecture.
End users access the application through the internet, where requests are first handled by an Nginx Load Balancer deployed on an EC2 instance in a public subnet.The load balancer distributes traffic to multiple backend EC2 instances located in private subnets across different Availability Zones, ensuring security, scalability, and fault tolerance.

2. Diagram



3.Components

3.1 End Users (Clients)

- Users access the application using web browsers.
- Requests are sent using **HTTPS (443)**.
- **HTTP (80)** requests are automatically redirected to HTTPS for secure communication.

3.2 Nginx Load Balancer (EC2 – Public Subnet)

- Deployed as an **EC2 instance with a public IP**.

- Acts as the **single entry point** for all user traffic.

Functions:

- SSL termination (handles HTTPS traffic)
- Reverse proxy to backend servers
- Load balancing between backend EC2 instances
- Caching to improve performance
- Monitoring backend health using scripts

3.3 Backend Tier (EC2 – Private Subnet)

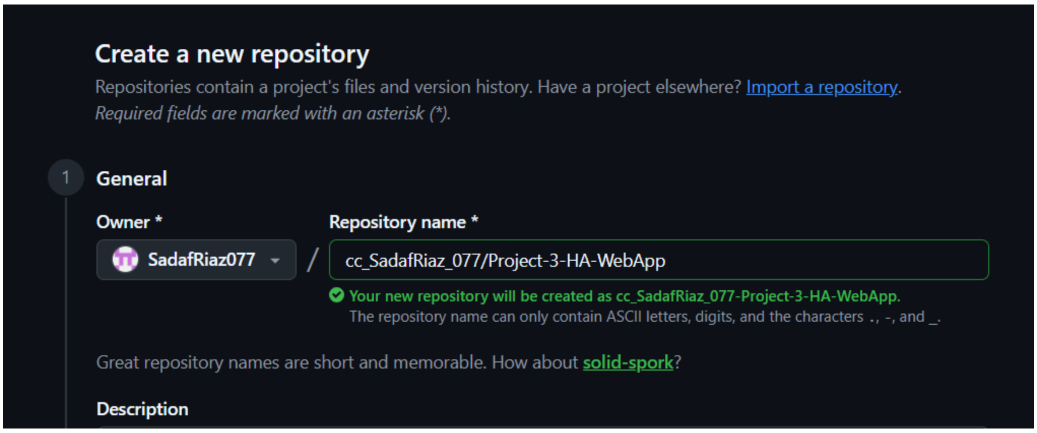
- Consists of multiple backend EC2 instances:
  - web-1 (Availability Zone A)
  - web-2 (Availability Zone B)
  - web-3 (Backup server)
- Each backend server runs:
  - Nginx web server
  - Application content
  - /health endpoint for health checks
- Backend servers are **not directly accessible from the internet**, improving security.

4. Data Flow

1. End users send requests to the application over **HTTPS (443)**.
2. Requests reach the **Nginx Load Balancer** in the public subnet.
3. The load balancer:
   - Terminates SSL
   - Applies caching rules
   - Selects a healthy backend server
4. Requests are forwarded to backend EC2 instances using **HTTP (80)** over the private network.
5. Backend servers process the request and return the response.
6. The load balancer sends the response back to the end user

1.Repository Creation and Initial Project Structure

1.1 Repository Structure



Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.
Required fields are marked with an asterisk (*).

1 General

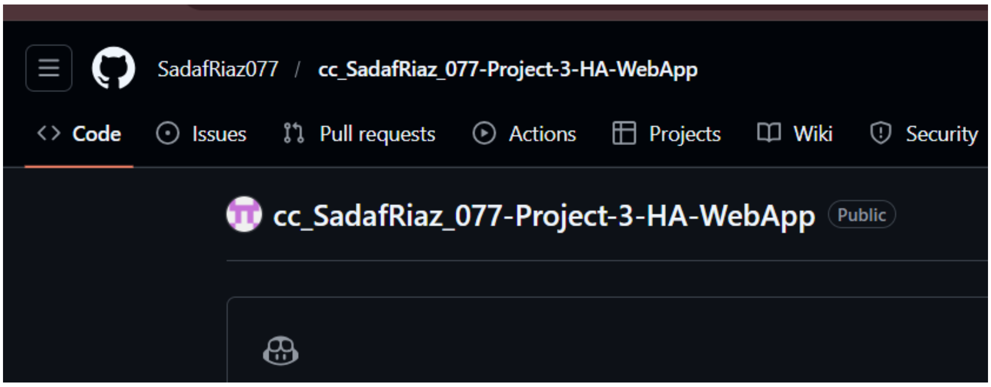
Owner *: SadafRiaz077

Repository name *: cc_SadafRiaz_077/Project-3-HA-WebApp

Your new repository will be created as cc_SadafRiaz_077-Project-3-HA-WebApp.
The repository name can only contain ASCII letters, digits, and the characters ., -, and _.

Great repository names are short and memorable. How about solid-spork?

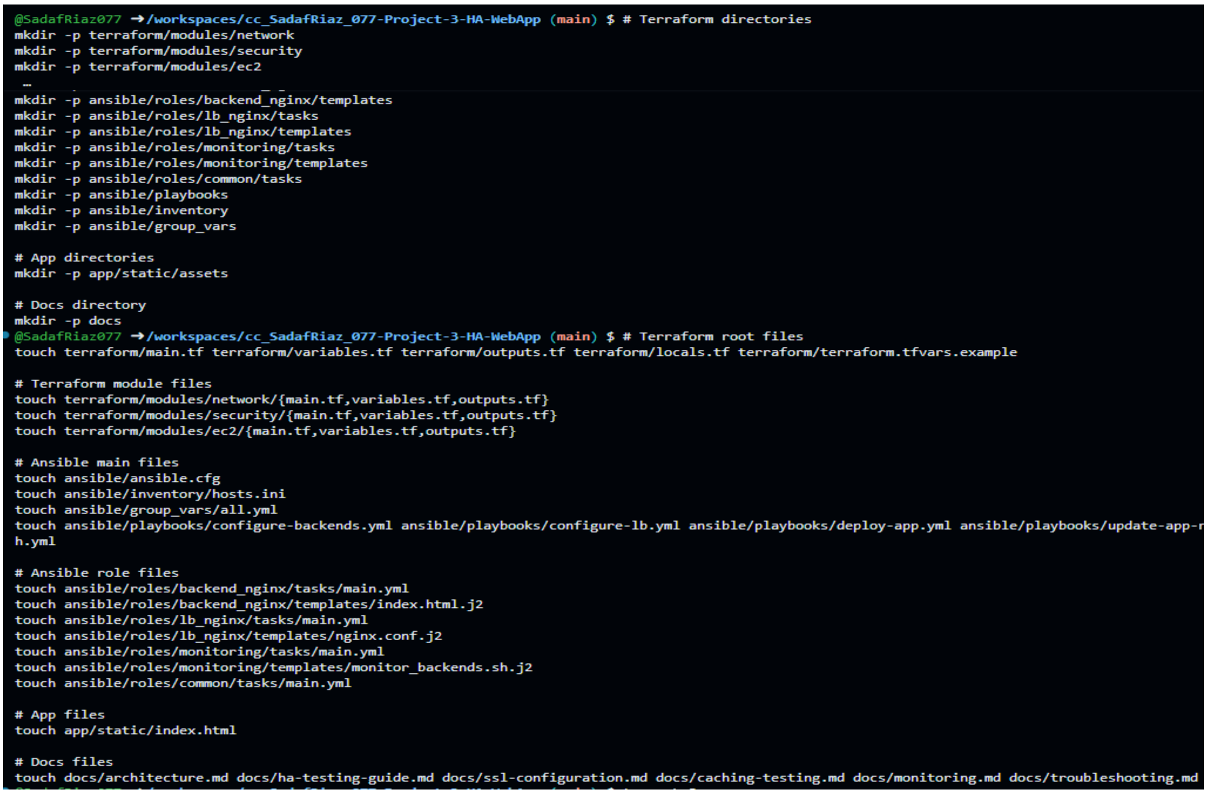
Description



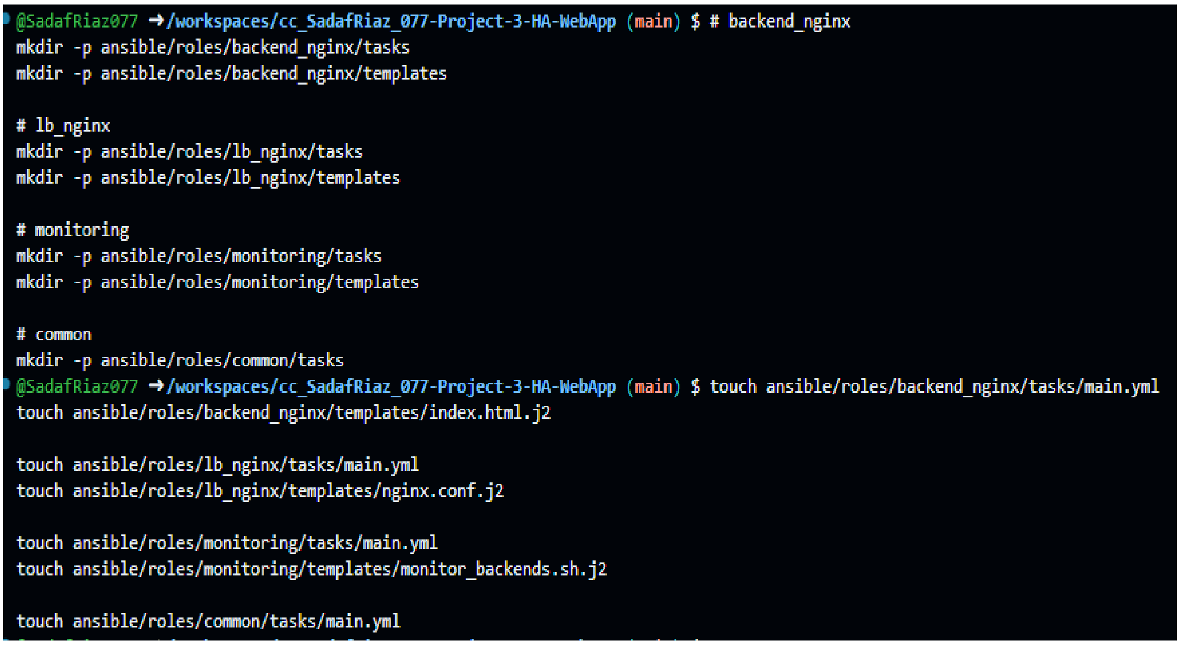
SadafRiaz077 / cc_SadafRiaz_077-Project-3-HA-WebApp

cc_SadafRiaz_077-Project-3-HA-WebApp Public

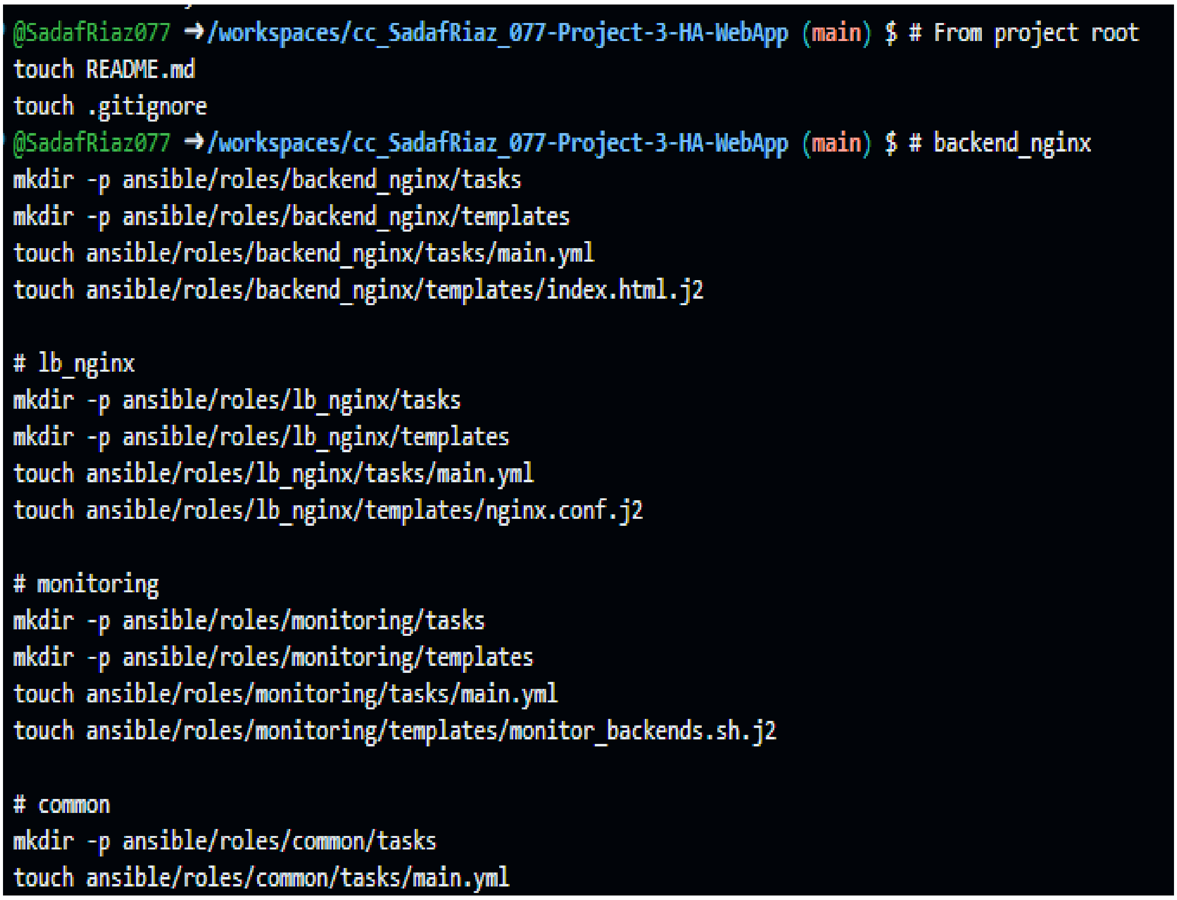
Project Structure



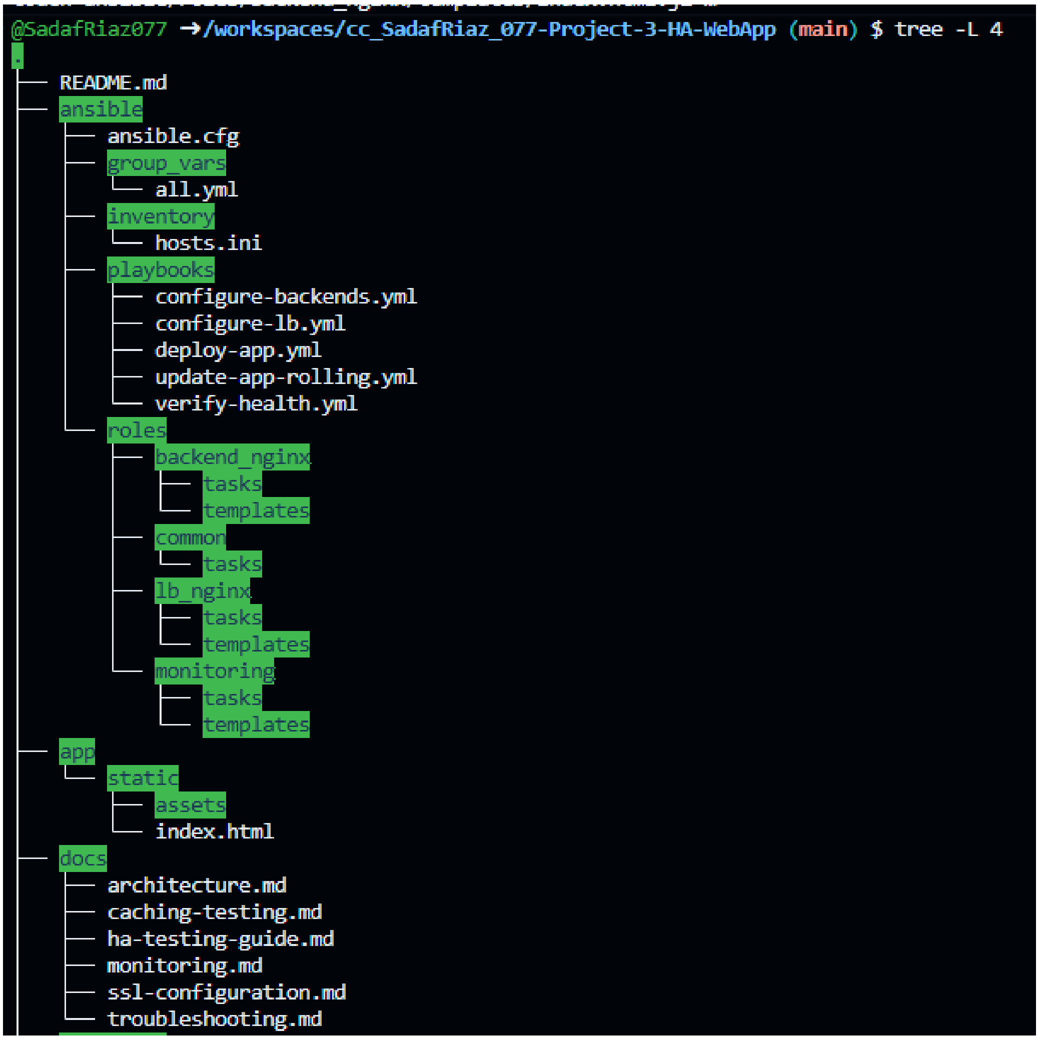
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ # Terraform directories
mkdir -p terraform/modules/network
mkdir -p terraform/modules/security
mkdir -p terraform/modules/ec2
mkdir -p ansible/roles/backend_nginx/templates
mkdir -p ansible/roles/lb_nginx/tasks
mkdir -p ansible/roles/lb_nginx/templates
mkdir -p ansible/roles/monitoring/tasks
mkdir -p ansible/roles/monitoring/templates
mkdir -p ansible/roles/common/tasks
mkdir -p ansible/playbooks
mkdir -p ansible/inventory
mkdir -p ansible/group_vars

# App directories
mkdir -p app/static/assets

# Docs directory
mkdir -p docs
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ # Terraform root files
touch terraform/main.tf terraform/variables.tf terraform/outputs.tf terraform/locals.tf terraform/terraform.tfvars.example

# Terraform module files
touch terraform/modules/network/{main.tf,variables.tf,outputs.tf}
touch terraform/modules/security/{main.tf,variables.tf,outputs.tf}
touch terraform/modules/ec2/{main.tf,variables.tf,outputs.tf}

# Ansible main files
touch ansible/ansible.cfg
touch ansible/inventory/hosts.ini
touch ansible/group_vars/all.yml
touch ansible/playbooks/configure-backends.yml ansible/playbooks/configure-lb.yml ansible/playbooks/deploy-app.yml ansible/playbooks/update-app-r
h.yml

# Ansible role files
touch ansible/roles/backend_nginx/tasks/main.yml
touch ansible/roles/backend_nginx/templates/index.html.j2
touch ansible/roles/lb_nginx/tasks/main.yml
touch ansible/roles/lb_nginx/templates/nginx.conf.j2
touch ansible/roles/monitoring/tasks/main.yml
touch ansible/roles/monitoring/templates/monitor_backends.sh.j2
touch ansible/roles/common/tasks/main.yml

# App files
touch app/static/index.html

# Docs files
touch docs/architecture.md docs/ha-testing-guide.md docs/ssl-configuration.md docs/caching-testing.md docs/monitoring.md docs/troubleshooting.md
```



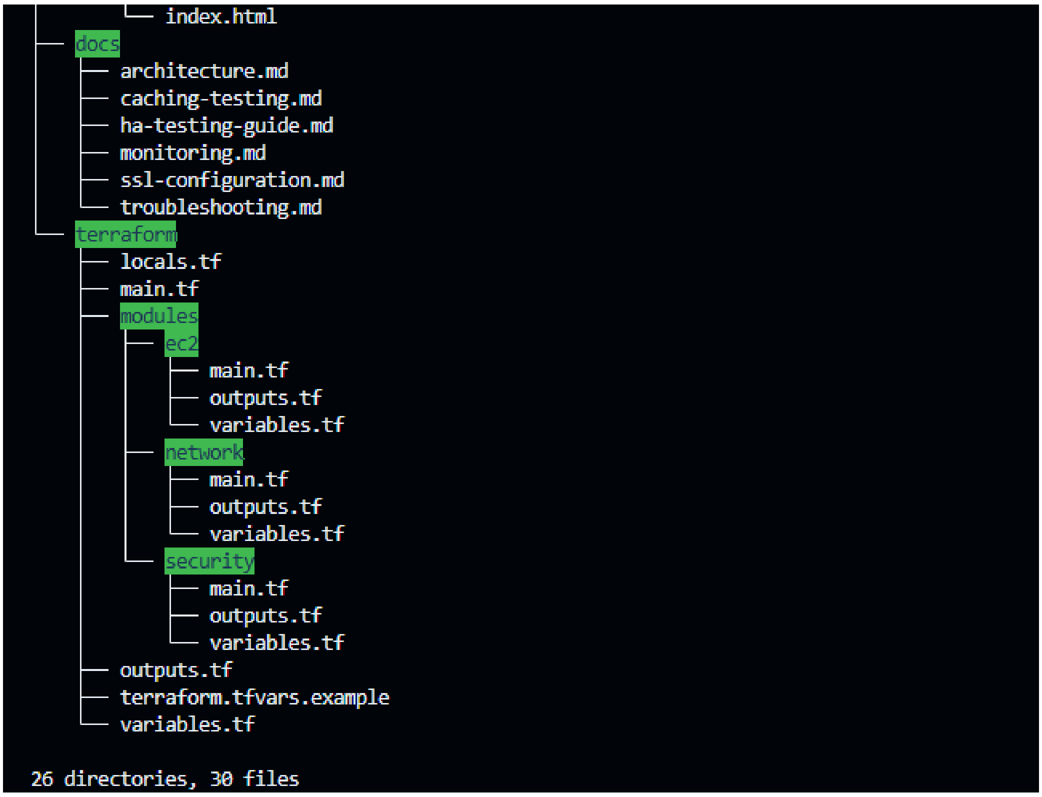
```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ # backend_nginx
mkdir -p ansible/roles/backend_nginx/tasks
mkdir -p ansible/roles/backend_nginx/templates

# lb_nginx
mkdir -p ansible/roles/lb_nginx/tasks
mkdir -p ansible/roles/lb_nginx/templates

# monitoring
mkdir -p ansible/roles/monitoring/tasks
mkdir -p ansible/roles/monitoring/templates

# common
mkdir -p ansible/roles/common/tasks
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ touch ansible/roles/backend_nginx/tasks/main.yml
touch ansible/roles/backend_nginx/templates/index.html.j2

touch ansible/roles/lb_nginx/tasks/main.yml
touch ansible/roles/lb_nginx/templates/nginx.conf.j2

touch ansible/roles/monitoring/tasks/main.yml
touch ansible/roles/monitoring/templates/monitor_backends.sh.j2

touch ansible/roles/common/tasks/main.yml
```



```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ # From project root
touch README.md
touch .gitignore
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ # backend_nginx
mkdir -p ansible/roles/backend_nginx/tasks
mkdir -p ansible/roles/backend_nginx/templates
touch ansible/roles/backend_nginx/tasks/main.yml
touch ansible/roles/backend_nginx/templates/index.html.j2

# lb_nginx
mkdir -p ansible/roles/lb_nginx/tasks
mkdir -p ansible/roles/lb_nginx/templates
touch ansible/roles/lb_nginx/tasks/main.yml
touch ansible/roles/lb_nginx/templates/nginx.conf.j2

# monitoring
mkdir -p ansible/roles/monitoring/tasks
mkdir -p ansible/roles/monitoring/templates
touch ansible/roles/monitoring/tasks/main.yml
touch ansible/roles/monitoring/templates/monitor_backends.sh.j2

# common
mkdir -p ansible/roles/common/tasks
touch ansible/roles/common/tasks/main.yml
```

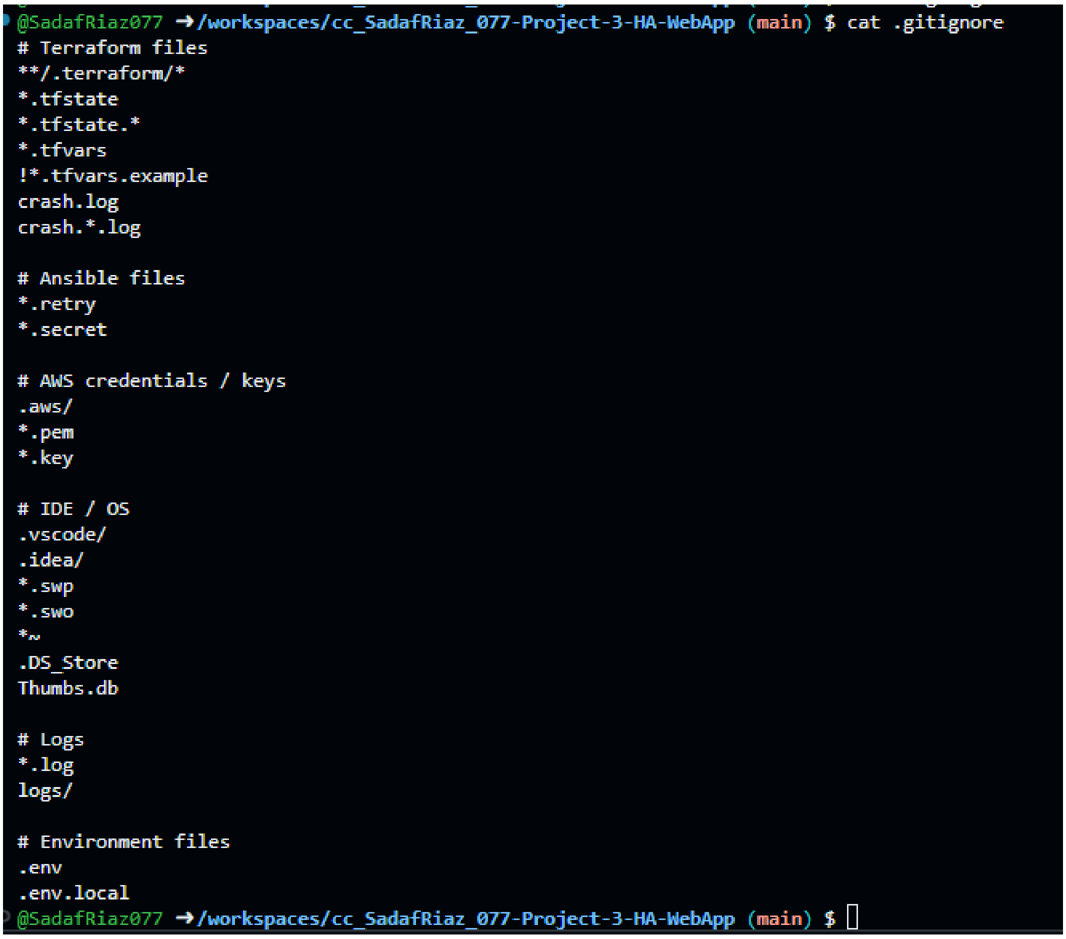


```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ tree -L 4
.
├── README.md
├── ansible
│   ├── ansible.cfg
│   ├── group_vars
│   │   └── all.yml
│   ├── inventory
│   │   └── hosts.ini
│   ├── playbooks
│   │   ├── configure-backends.yml
│   │   ├── configure-lb.yml
│   │   ├── deploy-app.yml
│   │   ├── update-app-rolling.yml
│   │   └── verify-health.yml
│   └── roles
│       ├── backend_nginx
│       │   ├── tasks
│       │   └── templates
│       ├── common
│       │   └── tasks
│       ├── lb_nginx
│       │   ├── tasks
│       │   └── templates
│       └── monitoring
│           ├── tasks
│           └── templates
├── app
│   └── static
│       ├── assets
│       └── index.html
├── docs
│   ├── architecture.md
│   ├── caching-testing.md
│   ├── ha-testing-guide.md
│   ├── monitoring.md
│   ├── ssl-configuration.md
│   └── troubleshooting.md
```

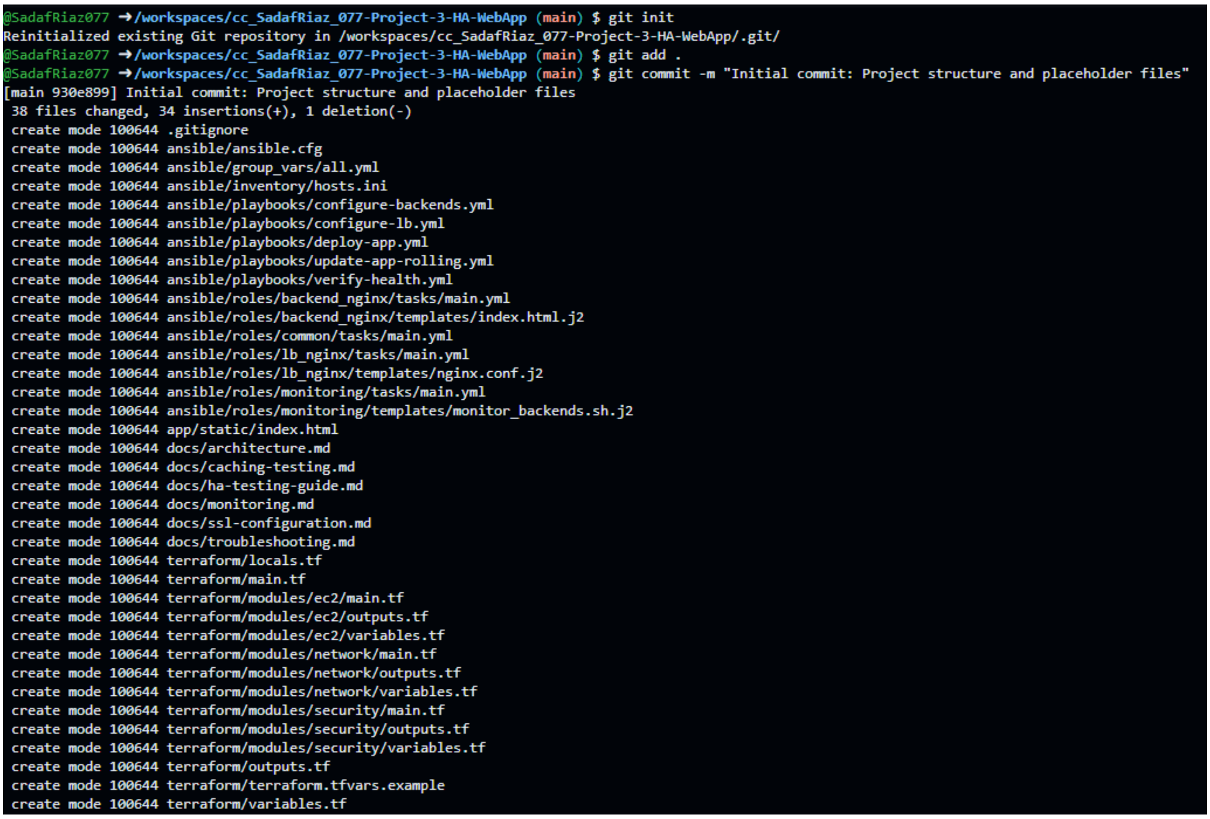


```
│       └── index.html
├── docs
│   ├── architecture.md
│   ├── caching-testing.md
│   ├── ha-testing-guide.md
│   ├── monitoring.md
│   ├── ssl-configuration.md
│   └── troubleshooting.md
└── terraform
    ├── locals.tf
    ├── main.tf
    ├── modules
    │   ├── ec2
    │   │   ├── main.tf
    │   │   ├── outputs.tf
    │   │   └── variables.tf
    │   ├── network
    │   │   ├── main.tf
    │   │   ├── outputs.tf
    │   │   └── variables.tf
    │   └── security
    │       ├── main.tf
    │       ├── outputs.tf
    │       └── variables.tf
    ├── outputs.tf
    ├── terraform.tfvars.example
    └── variables.tf

26 directories, 30 files
```

.gitignore Configuration for Terraform, Ansible, and AWS Files(project3_part1_gitignore.png)

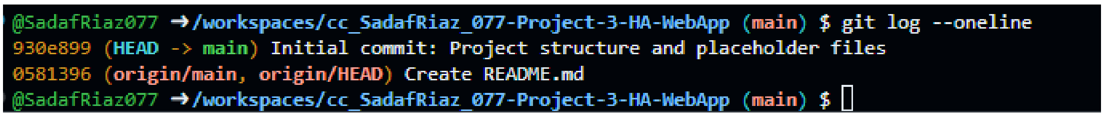


```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat .gitignore
# Terraform files
**/.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
!*.tfvars.example
crash.log
crash.*.log

# Ansible files
*.retry
*.secret

# AWS credentials / keys
.aws/
*.pem
*.key

# IDE / OS
.vscode/
.idea/
*.swp
*.swo
*~
.DS_Store
Thumbs.db

# Logs
*.log
logs/

# Environment files
.env
.env.local
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ 
```



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ git init
Reinitialized existing Git repository in /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/.git/
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ git add .
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ git commit -m "Initial commit: Project structure and placeholder files"
[main 930e899] Initial commit: Project structure and placeholder files
 38 files changed, 34 insertions(+), 1 deletion(-)
 create mode 100644 .gitignore
 create mode 100644 ansible/ansible.cfg
 create mode 100644 ansible/group_vars/all.yml
 create mode 100644 ansible/inventory/hosts.ini
 create mode 100644 ansible/playbooks/configure-backends.yml
 create mode 100644 ansible/playbooks/configure-lb.yml
 create mode 100644 ansible/playbooks/deploy-app.yml
 create mode 100644 ansible/playbooks/update-app-rolling.yml
 create mode 100644 ansible/playbooks/verify-health.yml
 create mode 100644 ansible/roles/backend_nginx/tasks/main.yml
 create mode 100644 ansible/roles/backend_nginx/templates/index.html.j2
 create mode 100644 ansible/roles/common/tasks/main.yml
 create mode 100644 ansible/roles/lb_nginx/tasks/main.yml
 create mode 100644 ansible/roles/lb_nginx/templates/nginx.conf.j2
 create mode 100644 ansible/roles/monitoring/tasks/main.yml
 create mode 100644 ansible/roles/monitoring/templates/monitor_backends.sh.j2
 create mode 100644 app/static/index.html
 create mode 100644 docs/architecture.md
 create mode 100644 docs/caching-testing.md
 create mode 100644 docs/ha-testing-guide.md
 create mode 100644 docs/monitoring.md
 create mode 100644 docs/ssl-configuration.md
 create mode 100644 docs/troubleshooting.md
 create mode 100644 terraform/locals.tf
 create mode 100644 terraform/main.tf
 create mode 100644 terraform/modules/ec2/main.tf
 create mode 100644 terraform/modules/ec2/outputs.tf
 create mode 100644 terraform/modules/ec2/variables.tf
 create mode 100644 terraform/modules/network/main.tf
 create mode 100644 terraform/modules/network/outputs.tf
 create mode 100644 terraform/modules/network/variables.tf
 create mode 100644 terraform/modules/security/main.tf
 create mode 100644 terraform/modules/security/outputs.tf
 create mode 100644 terraform/modules/security/variables.tf
 create mode 100644 terraform/outputs.tf
 create mode 100644 terraform/terraform.tfvars.example
 create mode 100644 terraform/variables.tf
```

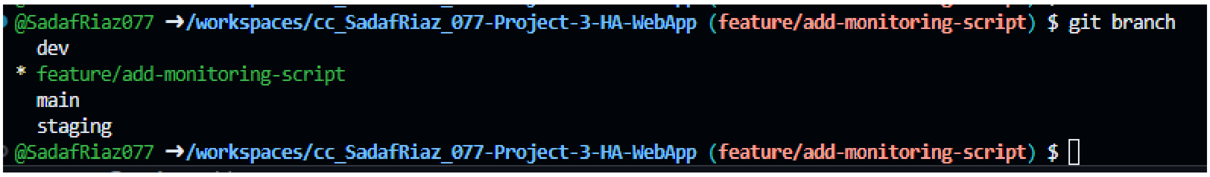
project3_part1_initial_commit.png



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ git log --oneline
930e899 (HEAD -> main) Initial commit: Project structure and placeholder files
0581396 (origin/main, origin/HEAD) Create README.md
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ 
```

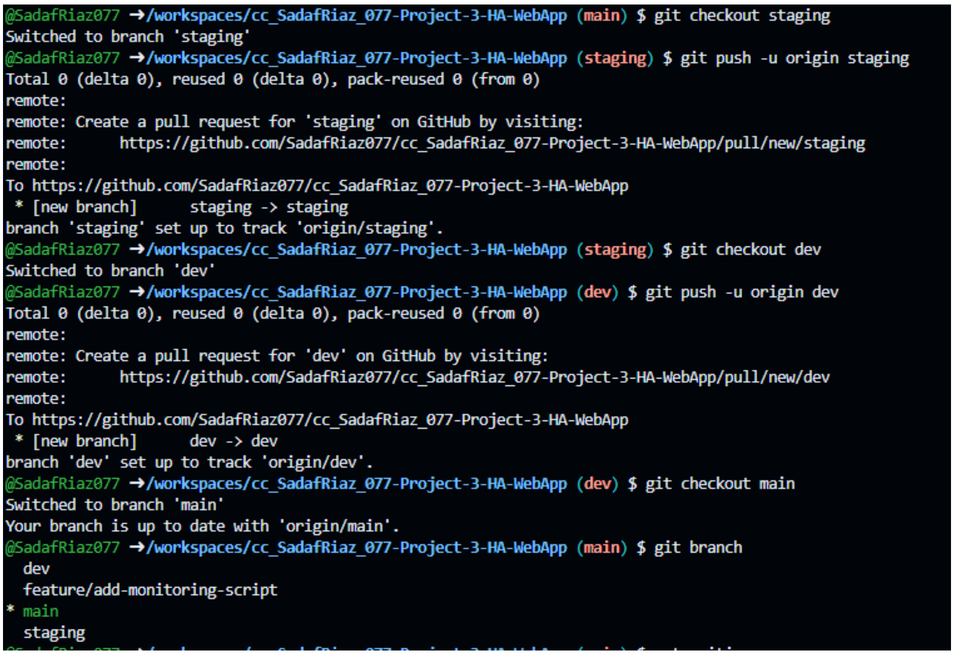
1.2 Git Branching Strategy

project3_part1_git_branches.png

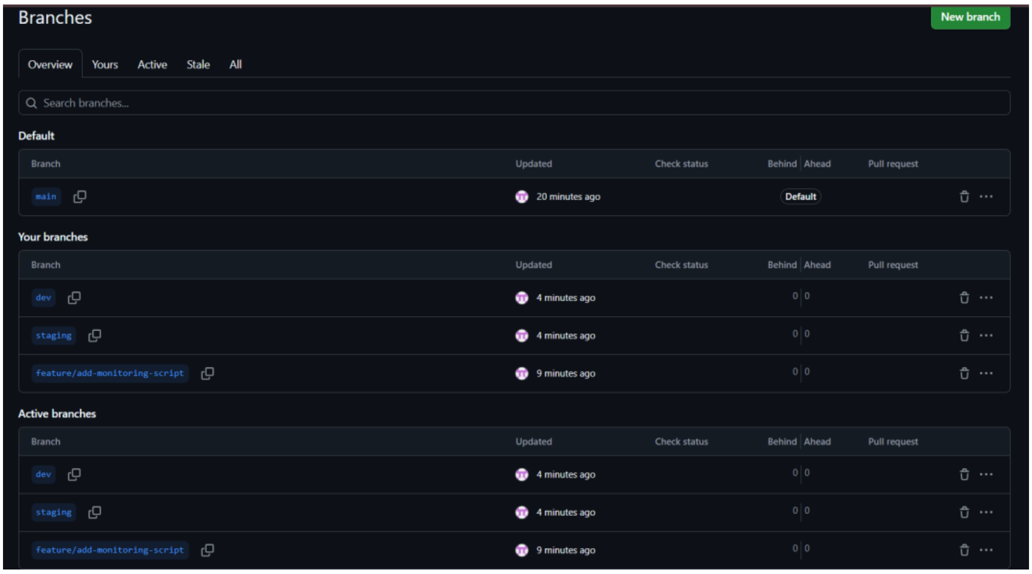


```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (feature/add-monitoring-script) $ git branch
  dev
* feature/add-monitoring-script
  main
  staging
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (feature/add-monitoring-script) $ 
```

Push to main



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ git checkout staging
Switched to branch 'staging'
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (staging) $ git push -u origin staging
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'staging' on GitHub by visiting:
remote:      https://github.com/SadafRiaz077/cc_SadafRiaz_077-Project-3-HA-WebApp/pull/new/staging
remote:
To https://github.com/SadafRiaz077/cc_SadafRiaz_077-Project-3-HA-WebApp
 * [new branch]      staging -> staging
branch 'staging' set up to track 'origin/staging'.
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (staging) $ git checkout dev
Switched to branch 'dev'
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (dev) $ git push -u origin dev
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:      https://github.com/SadafRiaz077/cc_SadafRiaz_077-Project-3-HA-WebApp/pull/new/dev
remote:
To https://github.com/SadafRiaz077/cc_SadafRiaz_077-Project-3-HA-WebApp
 * [new branch]      dev -> dev
branch 'dev' set up to track 'origin/dev'.
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (dev) $ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ git branch
  dev
  feature/add-monitoring-script
* main
  staging
```



Branches

Default

Branch	Updated	Check status	Behind | Ahead	Pull request
main	20 minutes ago		Default	

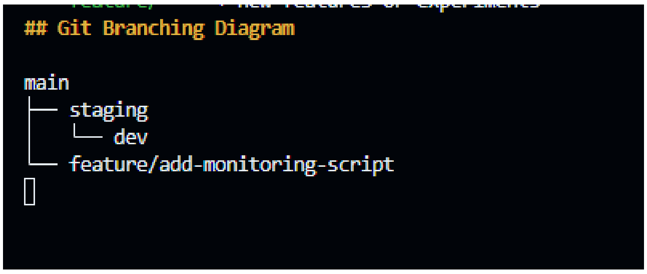
Your branches

Branch	Updated	Check status	Behind | Ahead	Pull request
dev	4 minutes ago		0 | 0	
staging	4 minutes ago		0 | 0	
feature/add-monitoring-script	9 minutes ago		0 | 0	

Active branches

Branch	Updated	Check status	Behind | Ahead	Pull request
dev	4 minutes ago		0 | 0	
staging	4 minutes ago		0 | 0	
feature/add-monitoring-script	9 minutes ago		0 | 0	

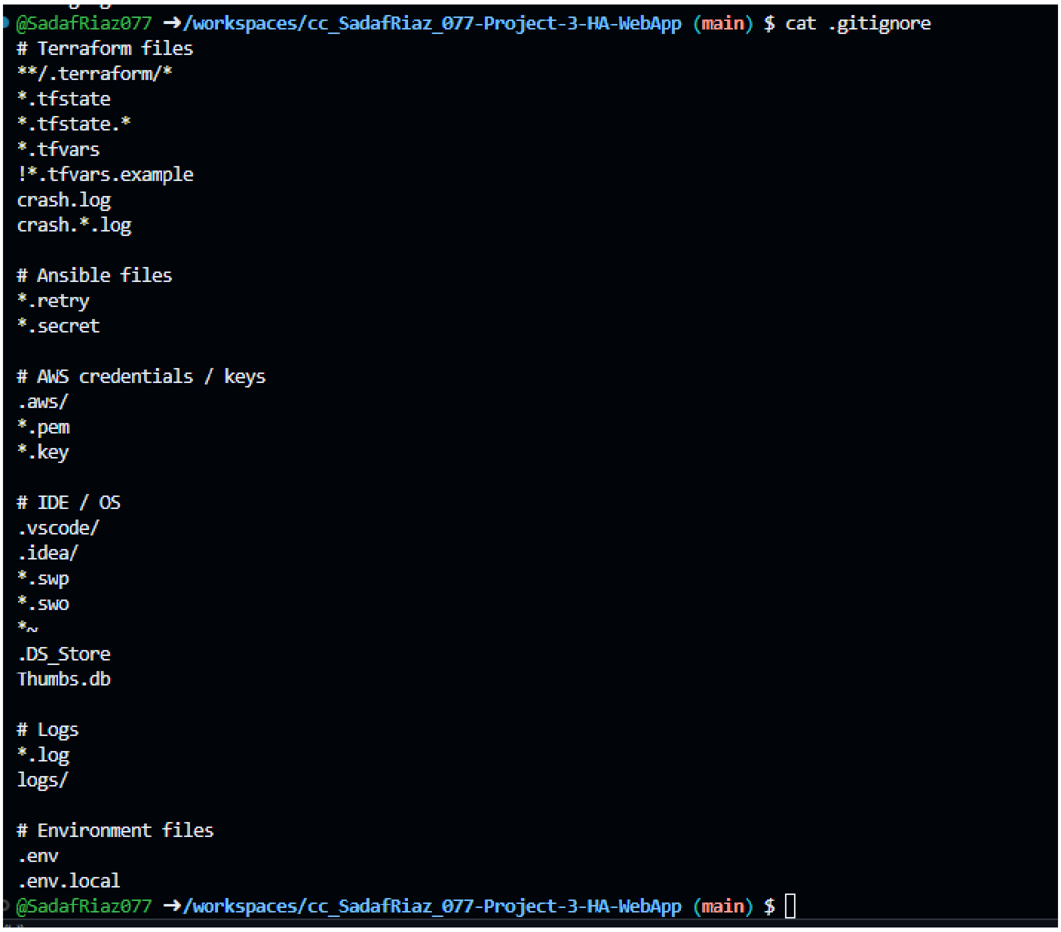
project3_part1_branching_diagram.png



```
## Git Branching Diagram

main
├── staging
│   └── dev
└── feature/add-monitoring-script
```

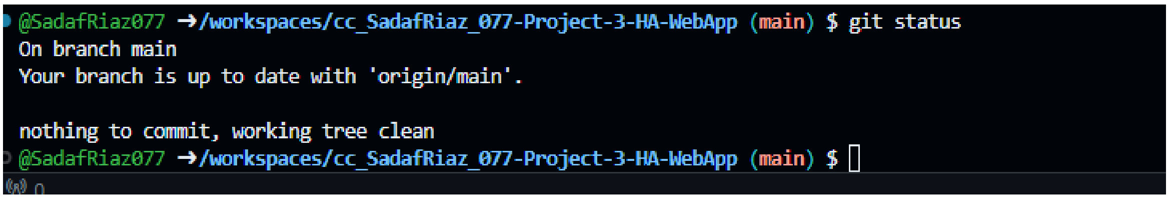
1.3 .gitignore Configuration

project3_part1_gitignore_content.png



```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat .gitignore
# Terraform files
**/.terraform/*
*.tfstate
*.tfstate.*
*.tfvars
!*.tfvars.example
crash.log
crash.*.log

# Ansible files
*.retry
*.secret

# AWS credentials / keys
.aws/
*.pem
*.key

# IDE / OS
.vscode/
.idea/
*.swp
*.swo
*~
.DS_Store
Thumbs.db

# Logs
*.log
logs/

# Environment files
.env
.env.local
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $
```

project3_part1_git_status_clean.png

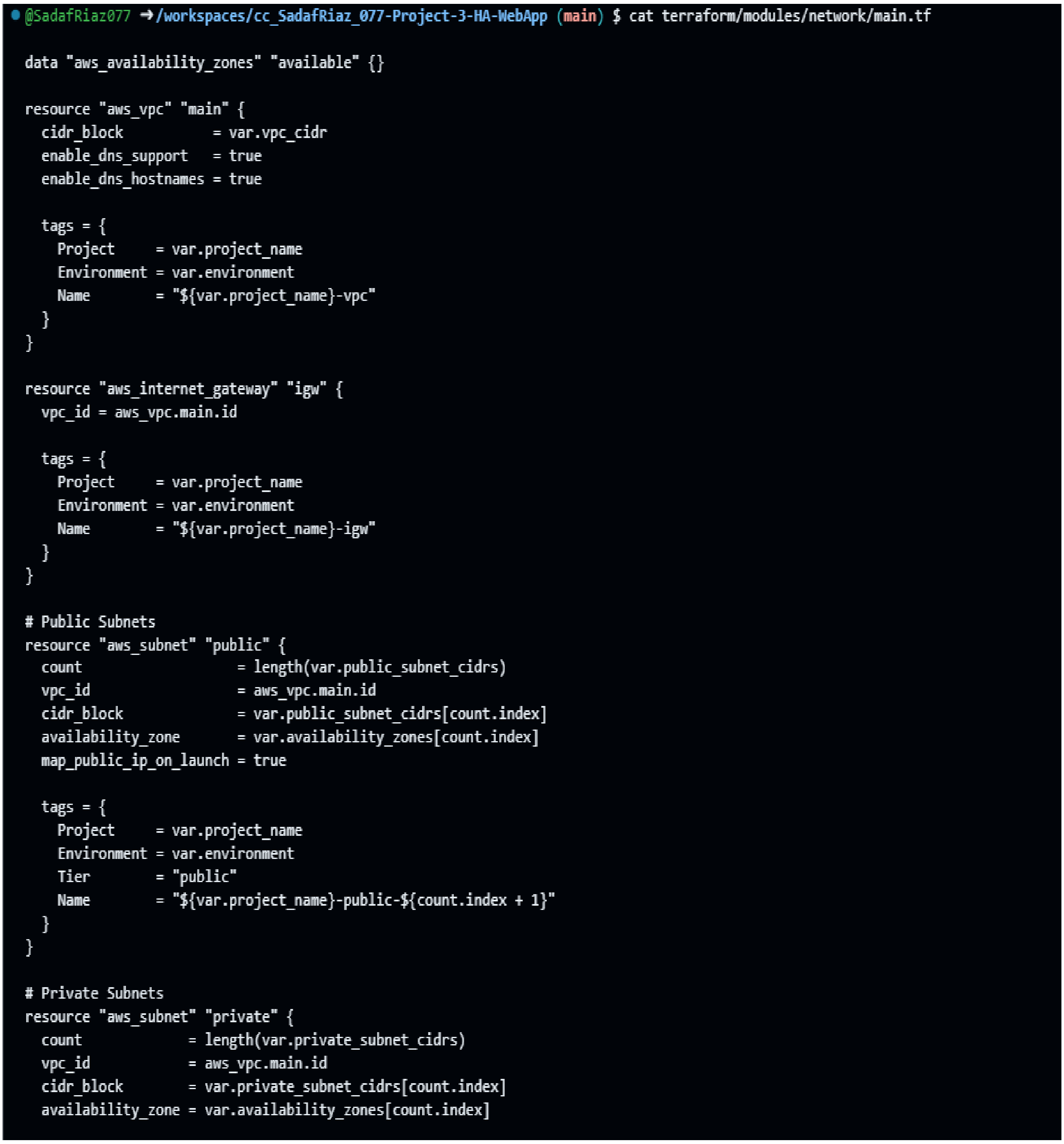


```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $
```

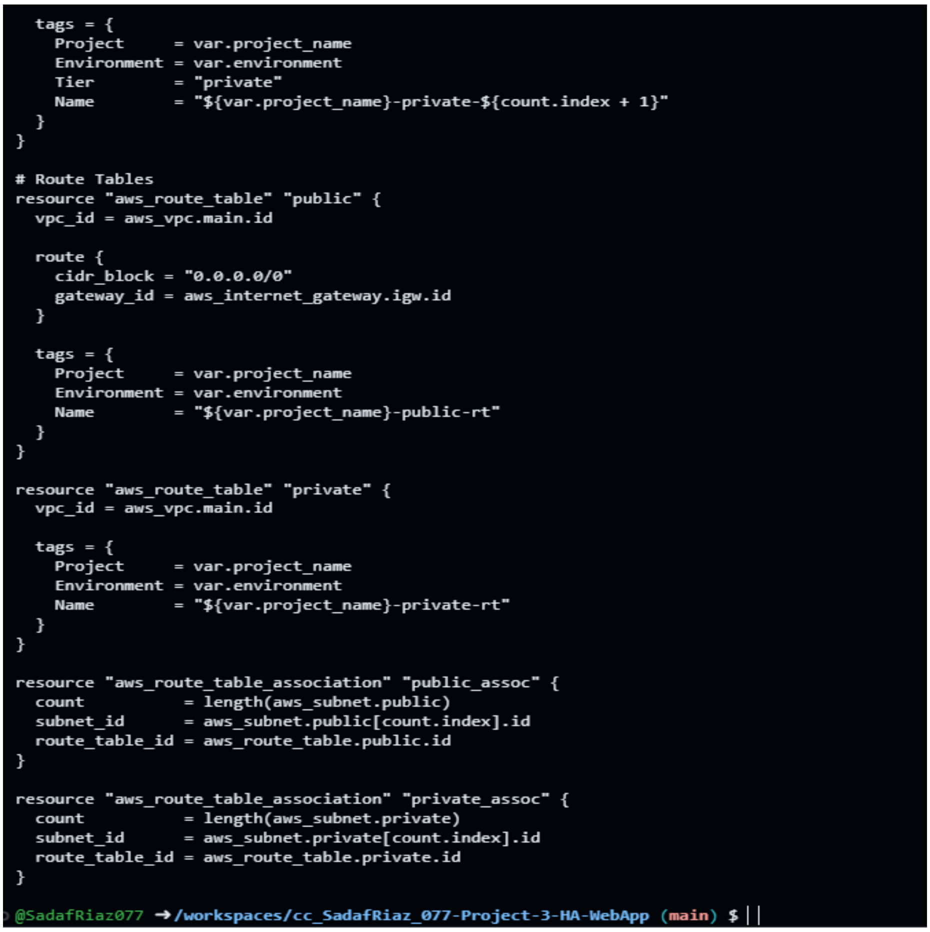
2. Terraform – Network, Security, EC2

2.1 Network Module – VPC & Multi-AZ Subnets

nano terraform/modules/network/main.tf

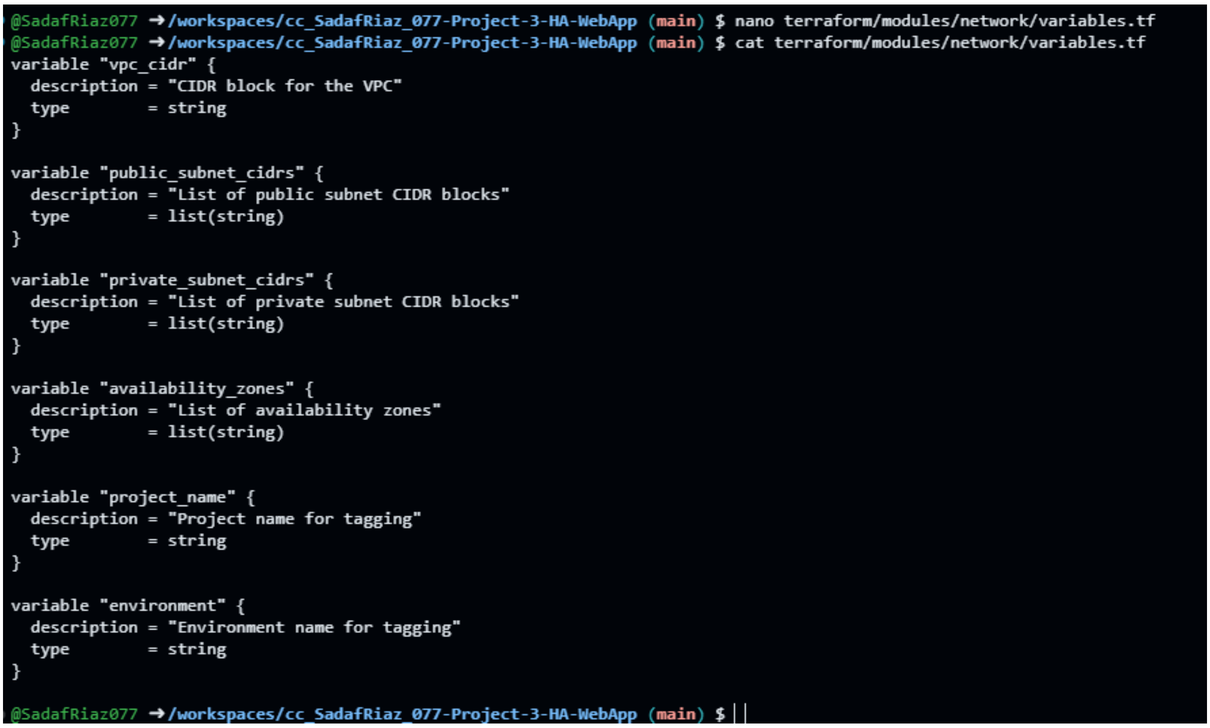


```
● @SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/network/main.tf

data "aws_availability_zones" "available" {}

resource "aws_vpc" "main" {
  cidr_block           = var.vpc_cidr
  enable_dns_support   = true
  enable_dns_hostnames = true

  tags = {
    Project     = var.project_name
    Environment = var.environment
    Name        = "${var.project_name}-vpc"
  }
}

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id

  tags = {
    Project     = var.project_name
    Environment = var.environment
    Name        = "${var.project_name}-igw"
  }
}

# Public Subnets
resource "aws_subnet" "public" {
  count                   = length(var.public_subnet_cidrs)
  vpc_id                  = aws_vpc.main.id
  cidr_block              = var.public_subnet_cidrs[count.index]
  availability_zone       = var.availability_zones[count.index]
  map_public_ip_on_launch = true

  tags = {
    Project     = var.project_name
    Environment = var.environment
    Tier        = "public"
    Name        = "${var.project_name}-public-${count.index + 1}"
  }
}

# Private Subnets
resource "aws_subnet" "private" {
  count             = length(var.private_subnet_cidrs)
  vpc_id            = aws_vpc.main.id
  cidr_block        = var.private_subnet_cidrs[count.index]
  availability_zone = var.availability_zones[count.index]
```



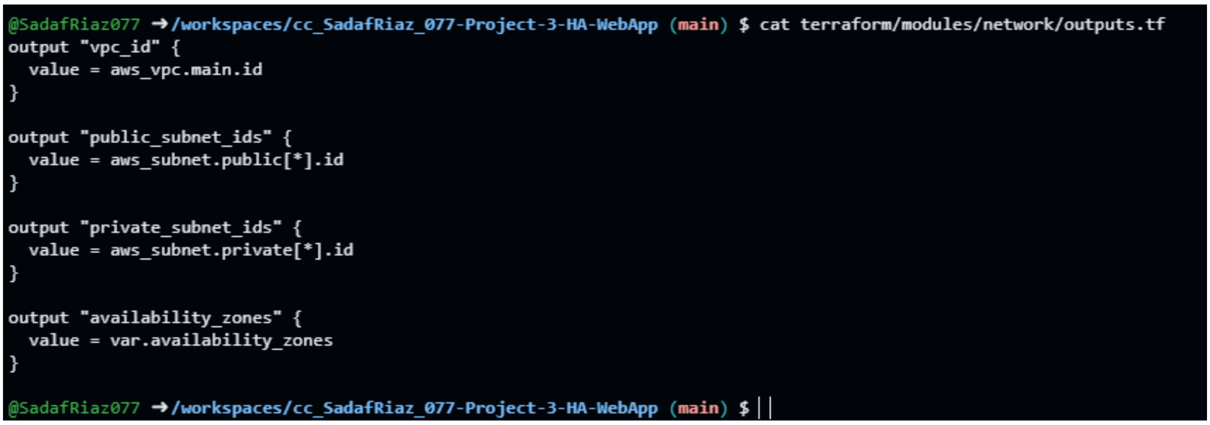
```
  tags = {
    Project     = var.project_name
    Environment = var.environment
    Tier        = "private"
    Name        = "${var.project_name}-private-${count.index + 1}"
  }
}

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

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

  tags = {
    Project     = var.project_name
    Environment = var.environment
    Name        = "${var.project_name}-public-rt"
  }
}

resource "aws_route_table" "private" {
  vpc_id = aws_vpc.main.id

  tags = {
    Project     = var.project_name
    Environment = var.environment
    Name        = "${var.project_name}-private-rt"
  }
}

resource "aws_route_table_association" "public_assoc" {
  count          = length(aws_subnet.public)
  subnet_id      = aws_subnet.public[count.index].id
  route_table_id = aws_route_table.public.id
}

resource "aws_route_table_association" "private_assoc" {
  count          = length(aws_subnet.private)
  subnet_id      = aws_subnet.private[count.index].id
  route_table_id = aws_route_table.private.id
}

@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ |
```

nano terraform/modules/network/variables.tf



```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ nano terraform/modules/network/variables.tf
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/network/variables.tf
variable "vpc_cidr" {
  description = "CIDR block for the VPC"
  type        = string
}

variable "public_subnet_cidrs" {
  description = "List of public subnet CIDR blocks"
  type        = list(string)
}

variable "private_subnet_cidrs" {
  description = "List of private subnet CIDR blocks"
  type        = list(string)
}

variable "availability_zones" {
  description = "List of availability zones"
  type        = list(string)
}

variable "project_name" {
  description = "Project name for tagging"
  type        = string
}

variable "environment" {
  description = "Environment name for tagging"
  type        = string
}

@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ |
```

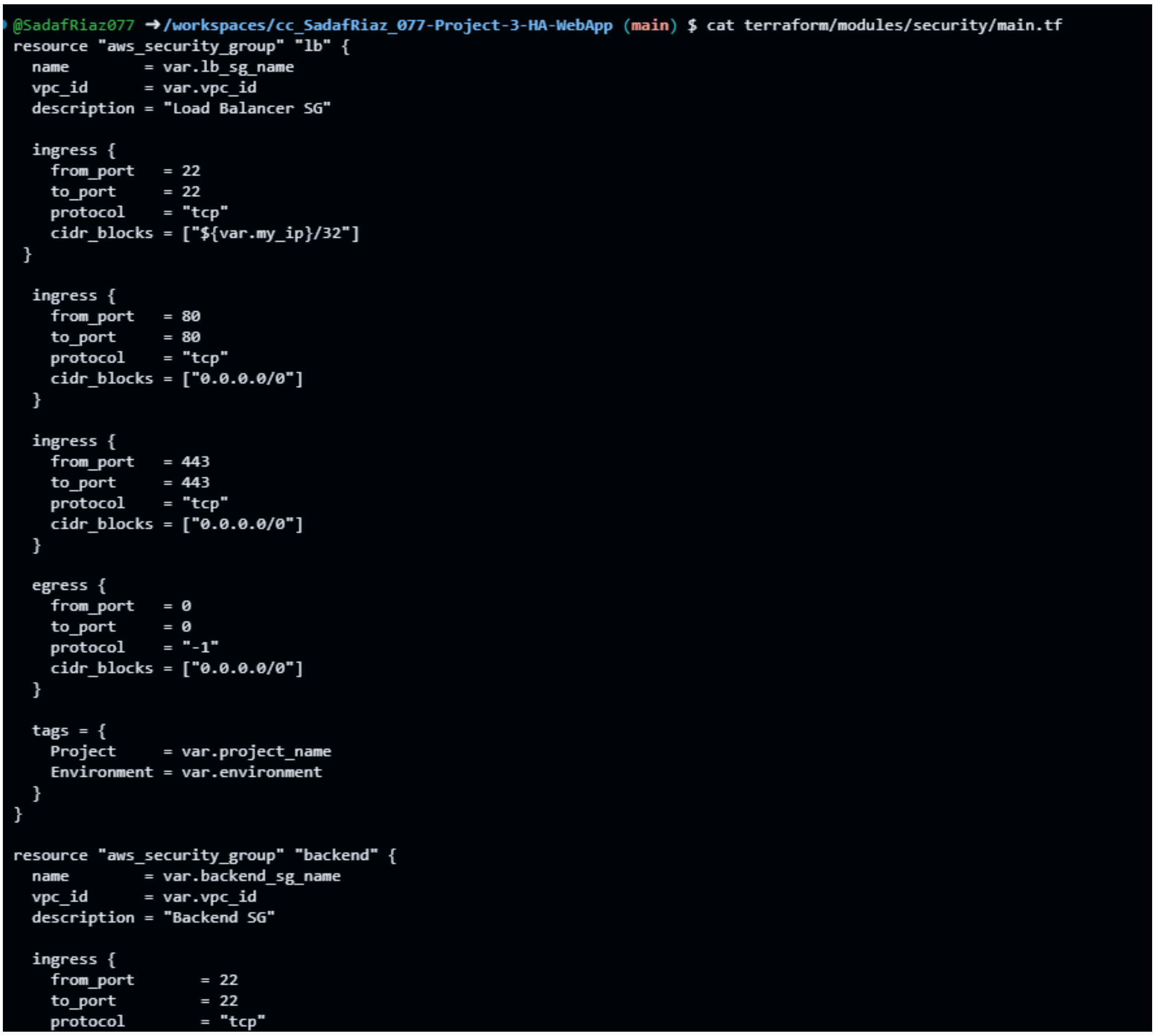
nano terraform/modules/network/outputs.tf



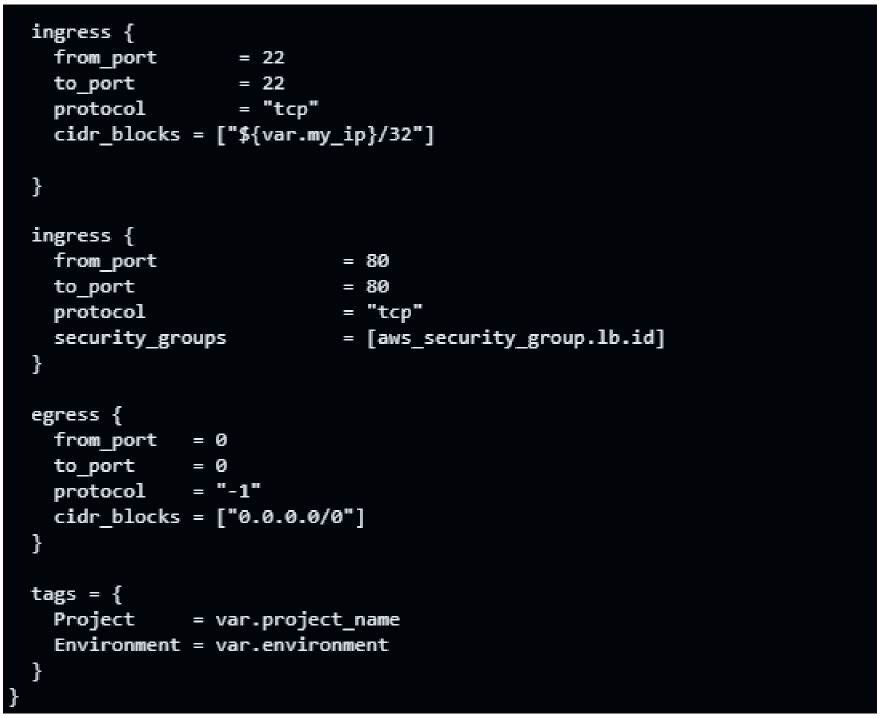
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/network/outputs.tf
output "vpc_id" {
  value = aws_vpc.main.id
}

output "public_subnet_ids" {
  value = aws_subnet.public[*].id
}

output "private_subnet_ids" {
  value = aws_subnet.private[*].id
}

output "availability_zones" {
  value = var.availability_zones
}

@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ ||
```

2.2 — SECURITY MODULE

nano terraform/modules/security/main.tf

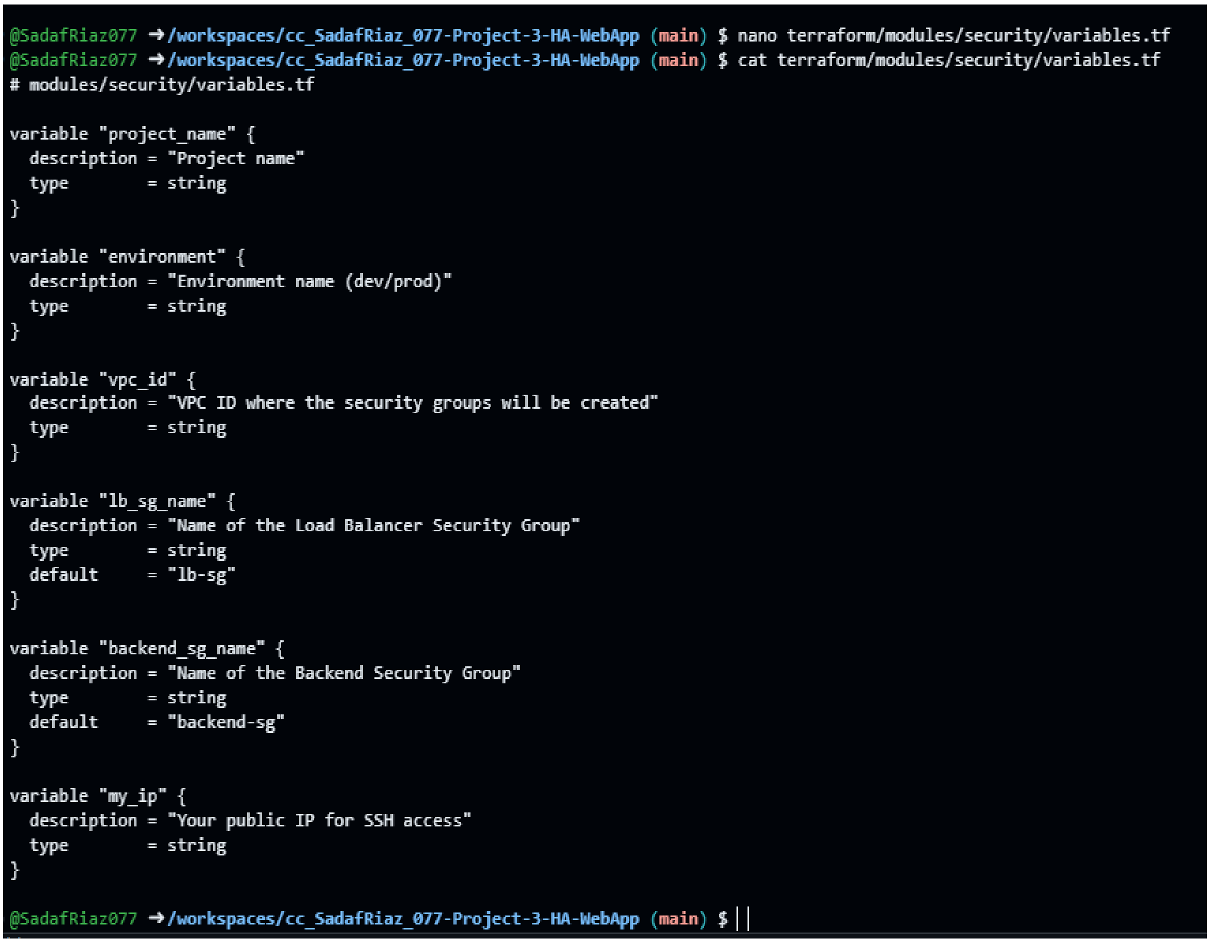


```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/security/main.tf
resource "aws_security_group" "lb" {
  name        = var.lb_sg_name
  vpc_id      = var.vpc_id
  description = "Load Balancer SG"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["${var.my_ip}/32"]
 }

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

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

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Project     = var.project_name
    Environment = var.environment
  }
}

resource "aws_security_group" "backend" {
  name        = var.backend_sg_name
  vpc_id      = var.vpc_id
  description = "Backend SG"

  ingress {
    from_port       = 22
    to_port         = 22
    protocol        = "tcp"
```



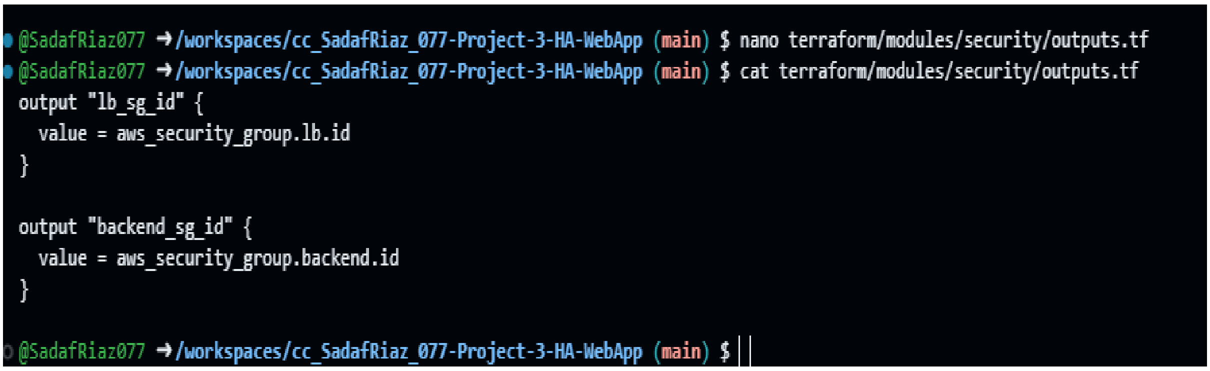
```
  ingress {
    from_port       = 22
    to_port         = 22
    protocol        = "tcp"
    cidr_blocks = ["${var.my_ip}/32"]

  }

  ingress {
    from_port                = 80
    to_port                  = 80
    protocol                 = "tcp"
    security_groups          = [aws_security_group.lb.id]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Project     = var.project_name
    Environment = var.environment
  }
}
```

nano terraform/modules/security/variables.tf



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ nano terraform/modules/security/variables.tf
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/security/variables.tf
# modules/security/variables.tf

variable "project_name" {
  description = "Project name"
  type        = string
}

variable "environment" {
  description = "Environment name (dev/prod)"
  type        = string
}

variable "vpc_id" {
  description = "VPC ID where the security groups will be created"
  type        = string
}

variable "lb_sg_name" {
  description = "Name of the Load Balancer Security Group"
  type        = string
  default     = "lb-sg"
}

variable "backend_sg_name" {
  description = "Name of the Backend Security Group"
  type        = string
  default     = "backend-sg"
}

variable "my_ip" {
  description = "Your public IP for SSH access"
  type        = string
}

@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ |
```

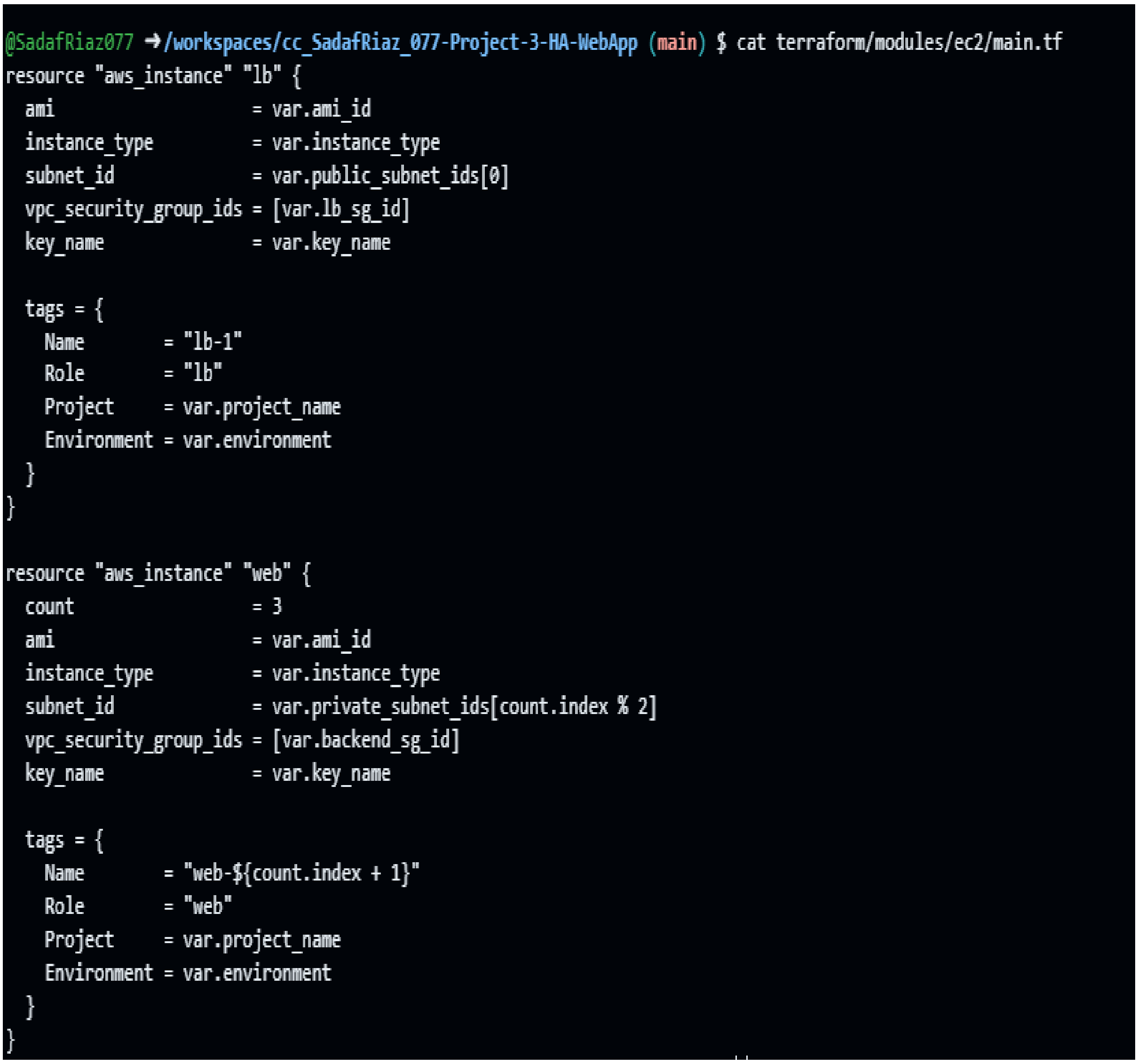
nano terraform/modules/security/outputs.tf



```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ nano terraform/modules/security/outputs.tf
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/security/outputs.tf
output "lb_sg_id" {
  value = aws_security_group.lb.id
}

output "backend_sg_id" {
  value = aws_security_group.backend.id
}

@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $
```

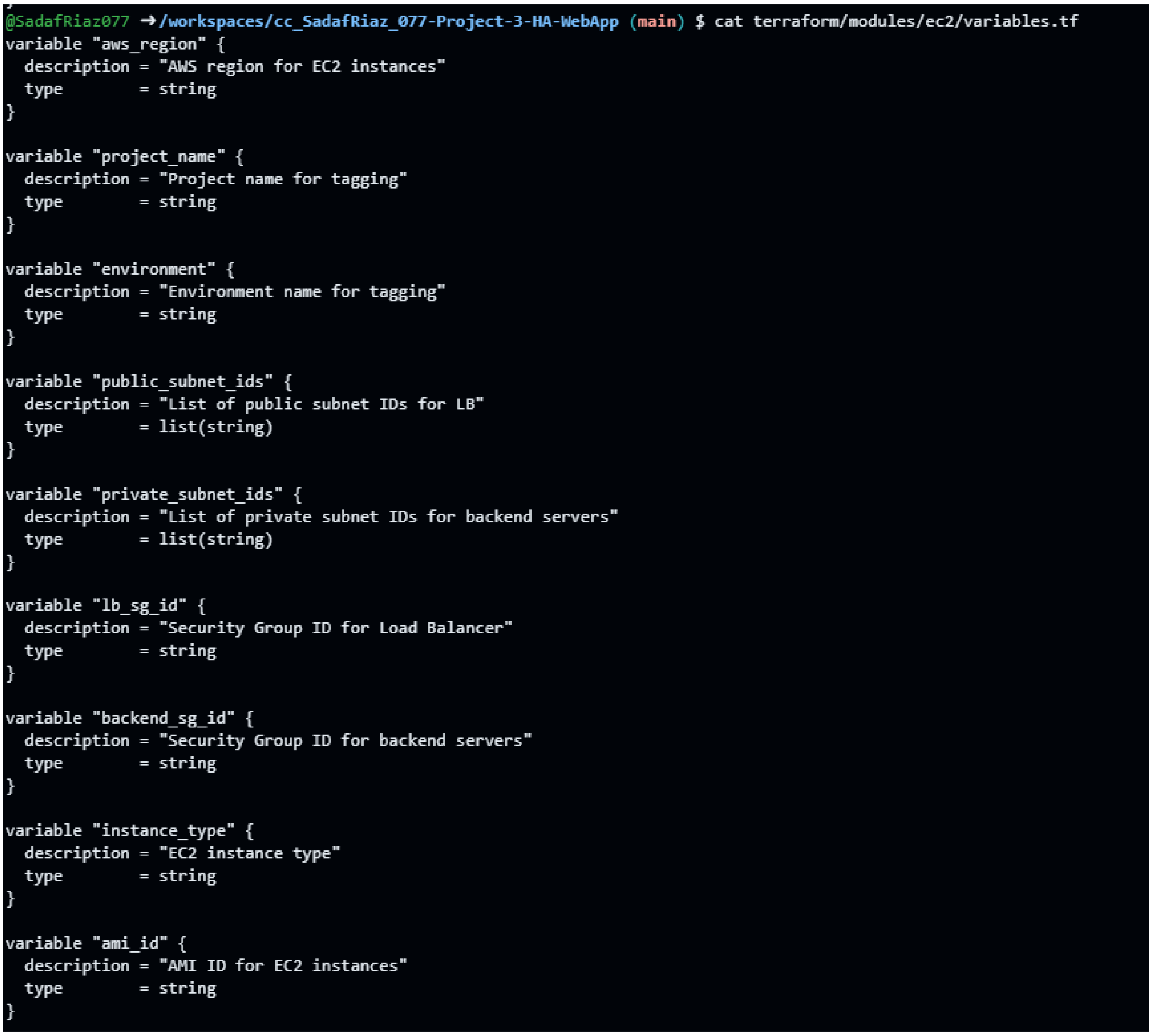
2.3 — EC2 MODULE

cat terraform/modules/ec2/main.tf

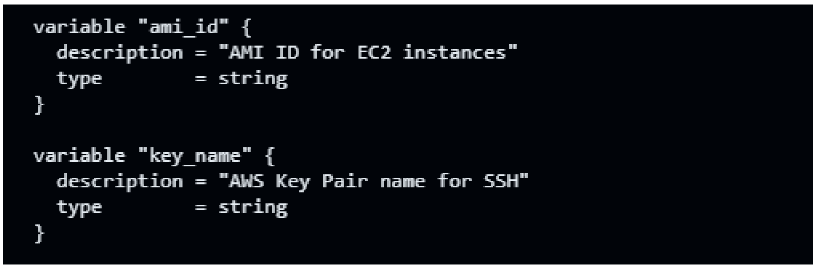


```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/ec2/main.tf
resource "aws_instance" "lb" {
  ami                    = var.ami_id
  instance_type          = var.instance_type
  subnet_id              = var.public_subnet_ids[0]
  vpc_security_group_ids = [var.lb_sg_id]
  key_name               = var.key_name

  tags = {
    Name        = "lb-1"
    Role        = "lb"
    Project     = var.project_name
    Environment = var.environment
  }
}

resource "aws_instance" "web" {
  count                  = 3
  ami                    = var.ami_id
  instance_type          = var.instance_type
  subnet_id              = var.private_subnet_ids[count.index % 2]
  vpc_security_group_ids = [var.backend_sg_id]
  key_name               = var.key_name

  tags = {
    Name        = "web-${count.index + 1}"
    Role        = "web"
    Project     = var.project_name
    Environment = var.environment
  }
}
```

cat terraform/modules/ec2/variables.tf

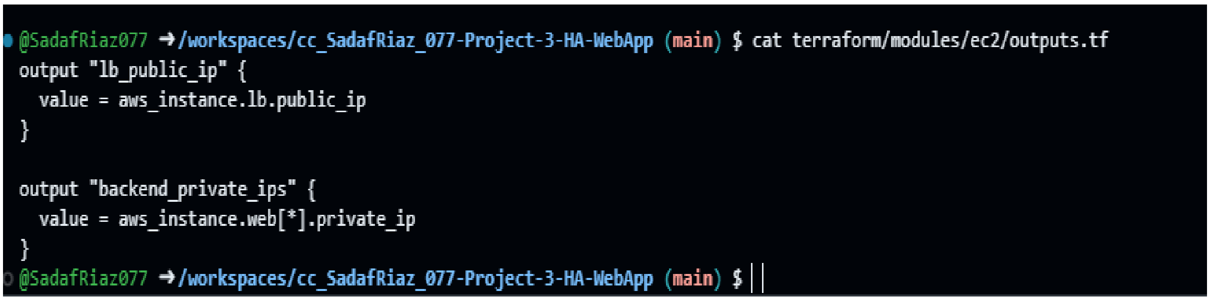


```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/ec2/variables.tf
variable "aws_region" {
  description = "AWS region for EC2 instances"
  type        = string
}

variable "project_name" {
  description = "Project name for tagging"
  type        = string
}

variable "environment" {
  description = "Environment name for tagging"
  type        = string
}

variable "public_subnet_ids" {
  description = "List of public subnet IDs for LB"
  type        = list(string)
}

variable "private_subnet_ids" {
  description = "List of private subnet IDs for backend servers"
  type        = list(string)
}

variable "lb_sg_id" {
  description = "Security Group ID for Load Balancer"
  type        = string
}

variable "backend_sg_id" {
  description = "Security Group ID for backend servers"
  type        = string
}

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

variable "ami_id" {
  description = "AMI ID for EC2 instances"
  type        = string
}
```

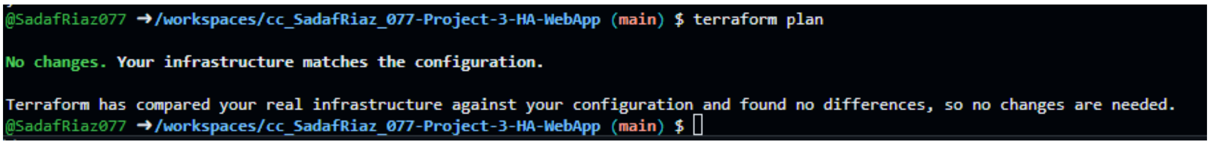


```
variable "ami_id" {
  description = "AMI ID for EC2 instances"
  type        = string
}

variable "key_name" {
  description = "AWS Key Pair name for SSH"
  type        = string
}
```

cat terraform/modules/ec2/outputs.tf



```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat terraform/modules/ec2/outputs.tf
output "lb_public_ip" {
  value = aws_instance.lb.public_ip
}

output "backend_private_ips" {
  value = aws_instance.web[*].private_ip
}
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ |
```

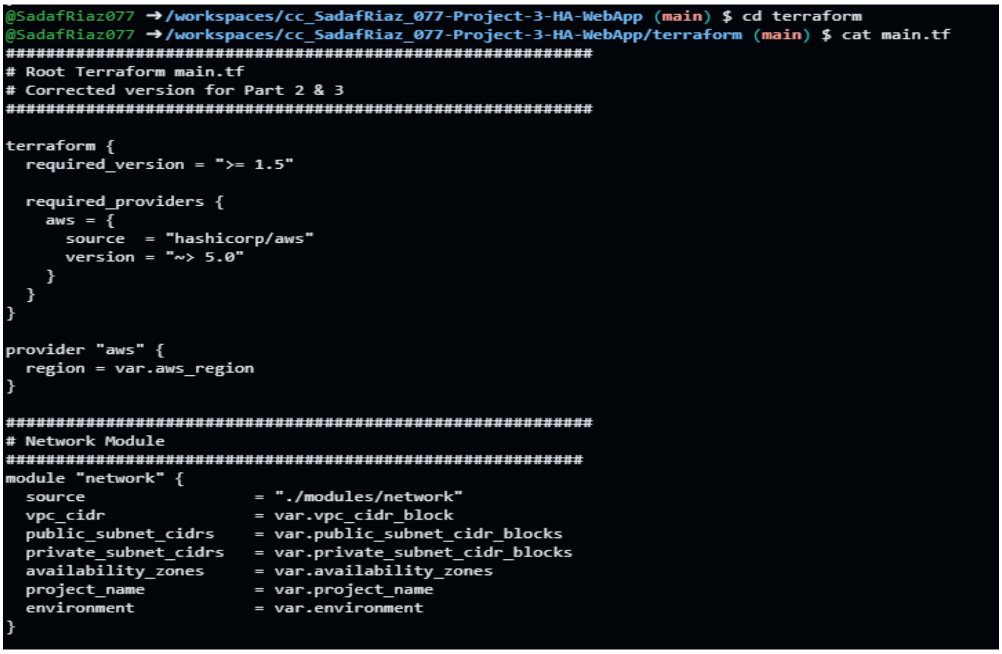


```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ terraform plan

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ 
```

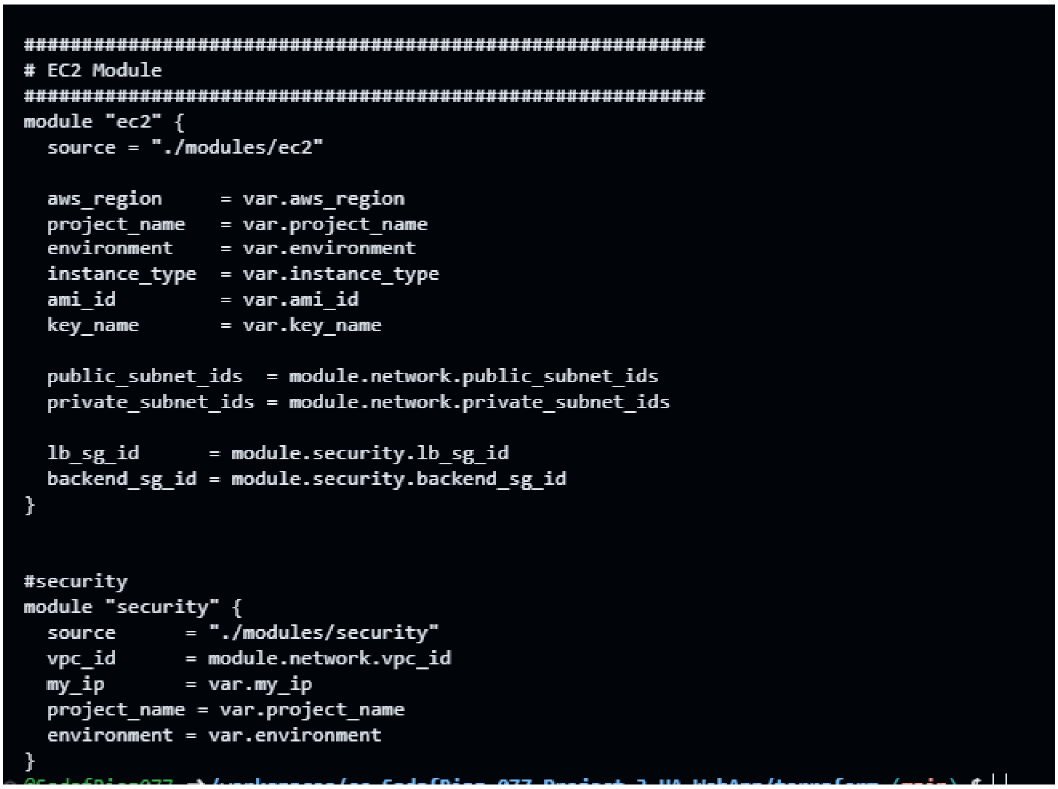
3. Root Terraform & Variables

3.1 Root Terraform Configuration

Root main.tf

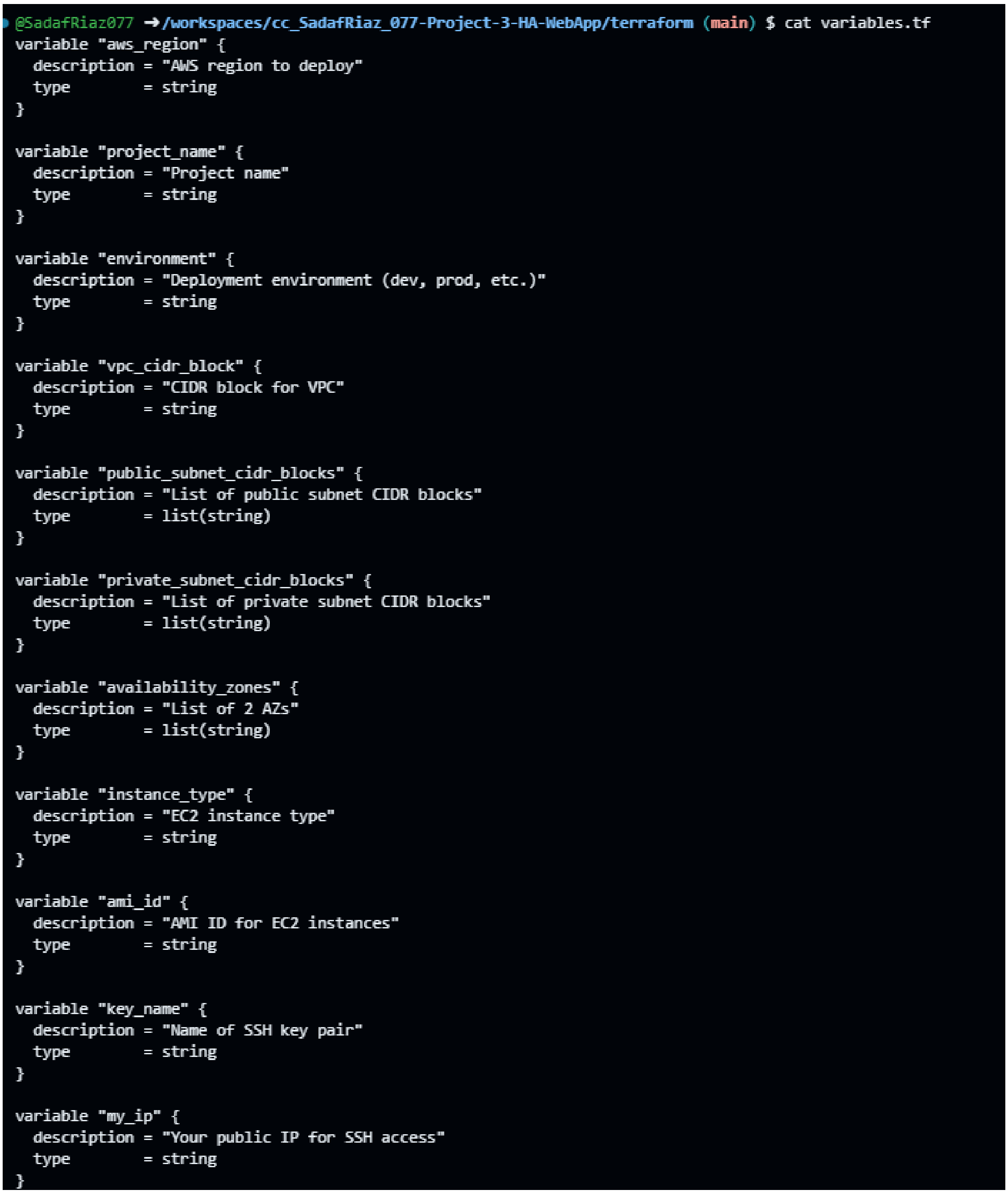


```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cd terraform
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ cat main.tf
############################################################
# Root Terraform main.tf
# Corrected version for Part 2 & 3
############################################################

terraform {
  required_version = ">= 1.5"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = var.aws_region
}

############################################################
# Network Module
###########################################################
module "network" {
  source                  = "./modules/network"
  vpc_cidr                = var.vpc_cidr_block
  public_subnet_cidrs     = var.public_subnet_cidr_blocks
  private_subnet_cidrs    = var.private_subnet_cidr_blocks
  availability_zones      = var.availability_zones
  project_name            = var.project_name
  environment             = var.environment
}
```



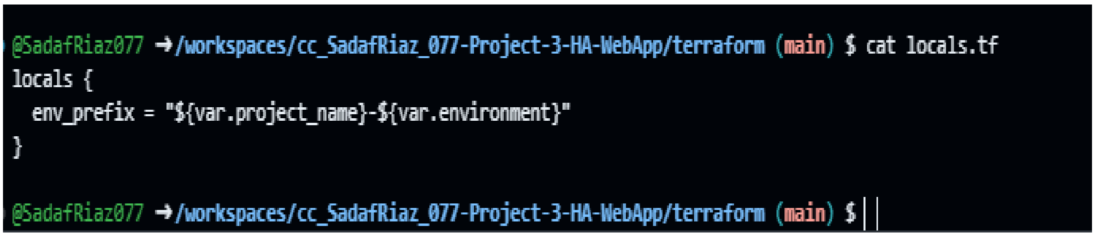
```
############################################################
# EC2 Module
############################################################
module "ec2" {
  source = "./modules/ec2"

  aws_region     = var.aws_region
  project_name   = var.project_name
  environment    = var.environment
  instance_type  = var.instance_type
  ami_id         = var.ami_id
  key_name       = var.key_name

  public_subnet_ids   = module.network.public_subnet_ids
  private_subnet_ids = module.network.private_subnet_ids

  lb_sg_id      = module.security.lb_sg_id
  backend_sg_id = module.security.backend_sg_id
}


#security
module "security" {
  source      = "./modules/security"
  vpc_id      = module.network.vpc_id
  my_ip       = var.my_ip
  project_name = var.project_name
  environment = var.environment
}
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ 
```

Variables.tf



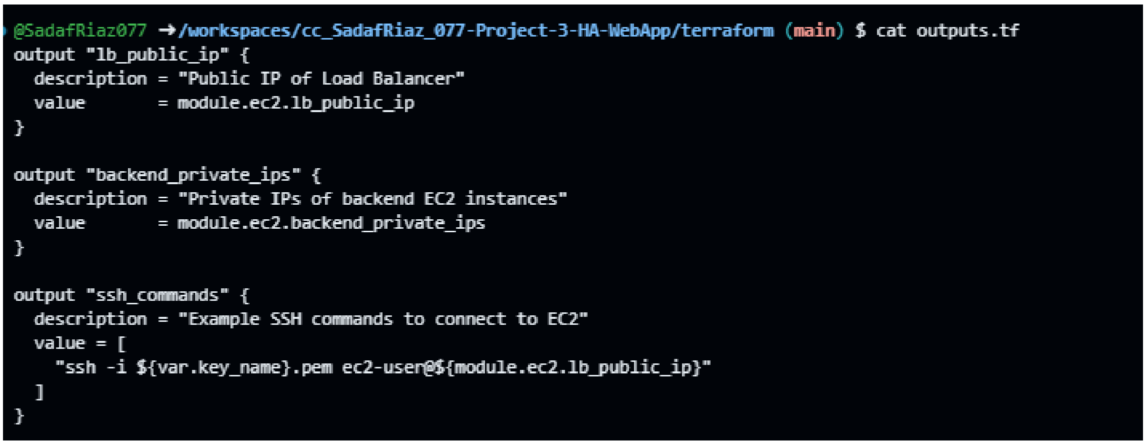
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ cat variables.tf
variable "aws_region" {
  description = "AWS region to deploy"
  type        = string
}

variable "project_name" {
  description = "Project name"
  type        = string
}

variable "environment" {
  description = "Deployment environment (dev, prod, etc.)"
  type        = string
}

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

variable "public_subnet_cidr_blocks" {
  description = "List of public subnet CIDR blocks"
  type        = list(string)
}

variable "private_subnet_cidr_blocks" {
  description = "List of private subnet CIDR blocks"
  type        = list(string)
}

variable "availability_zones" {
  description = "List of 2 AZs"
  type        = list(string)
}

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

variable "ami_id" {
  description = "AMI ID for EC2 instances"
  type        = string
}

variable "key_name" {
  description = "Name of SSH key pair"
  type        = string
}

variable "my_ip" {
  description = "Your public IP for SSH access"
  type        = string
}
```

Locals.tf



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ cat locals.tf
locals {
  env_prefix = "${var.project_name}-${var.environment}"
}

@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ |
```

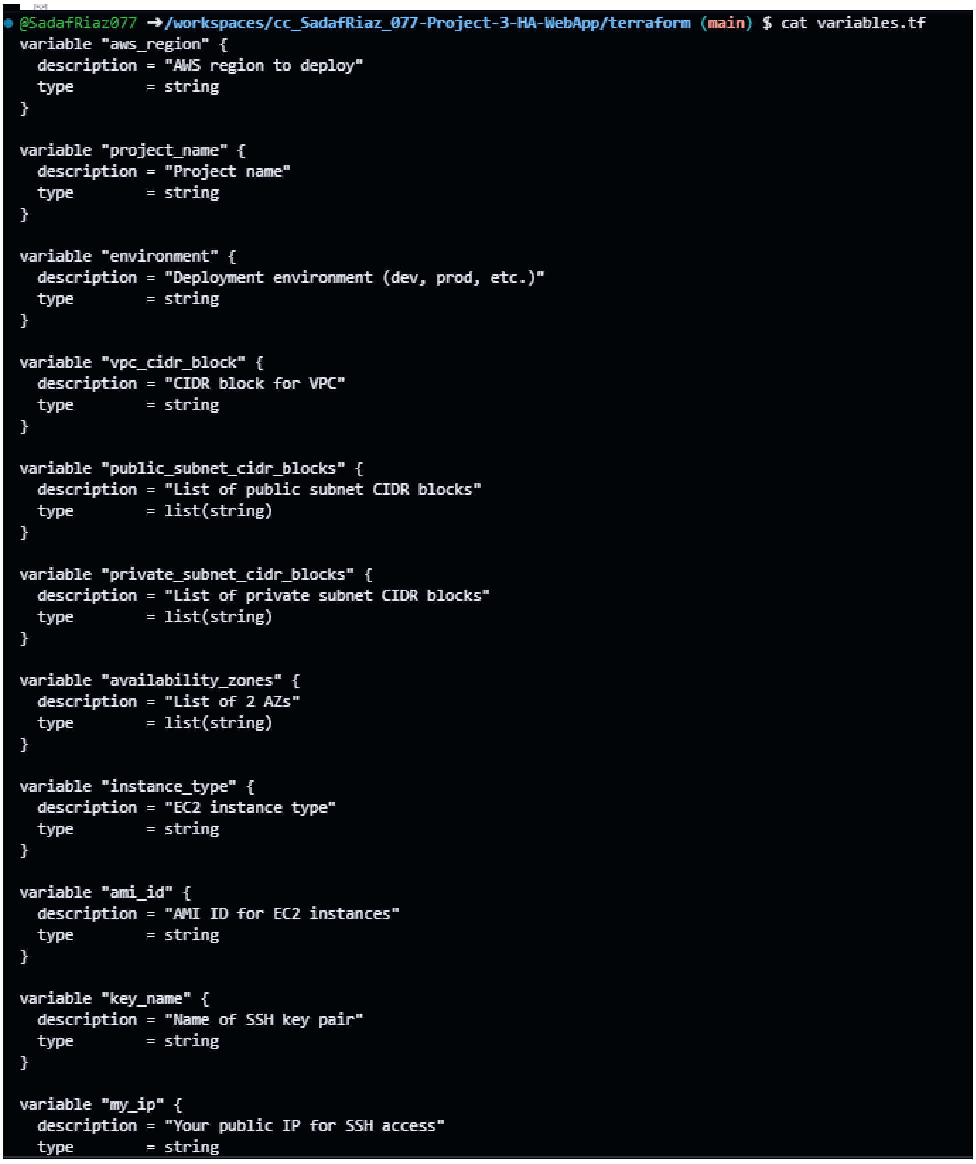
Outputs.tf



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ cat outputs.tf
output "lb_public_ip" {
  description = "Public IP of Load Balancer"
  value       = module.ec2.lb_public_ip
}

output "backend_private_ips" {
  description = "Private IPs of backend EC2 instances"
  value       = module.ec2.backend_private_ips
}

output "ssh_commands" {
  description = "Example SSH commands to connect to EC2"
  value = [
    "ssh -i ${var.key_name}.pem ec2-user@${module.ec2.lb_public_ip}"
  ]
}
```

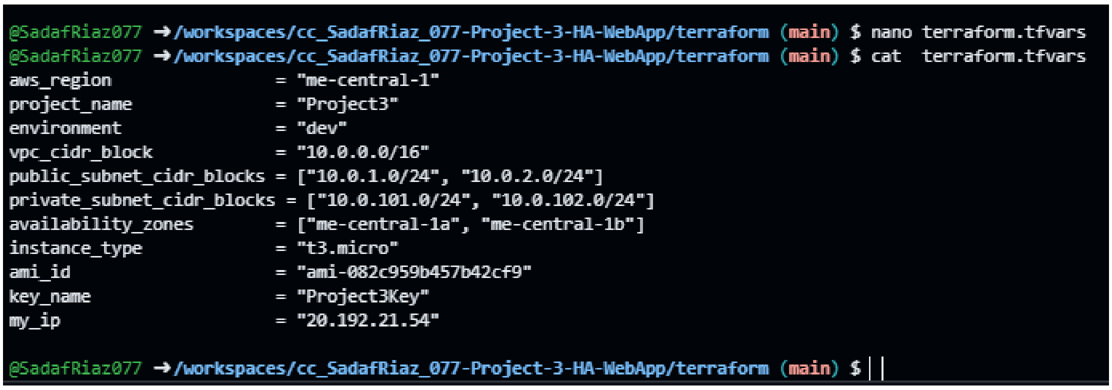
3.2 Variable Configuration

Variables.tf



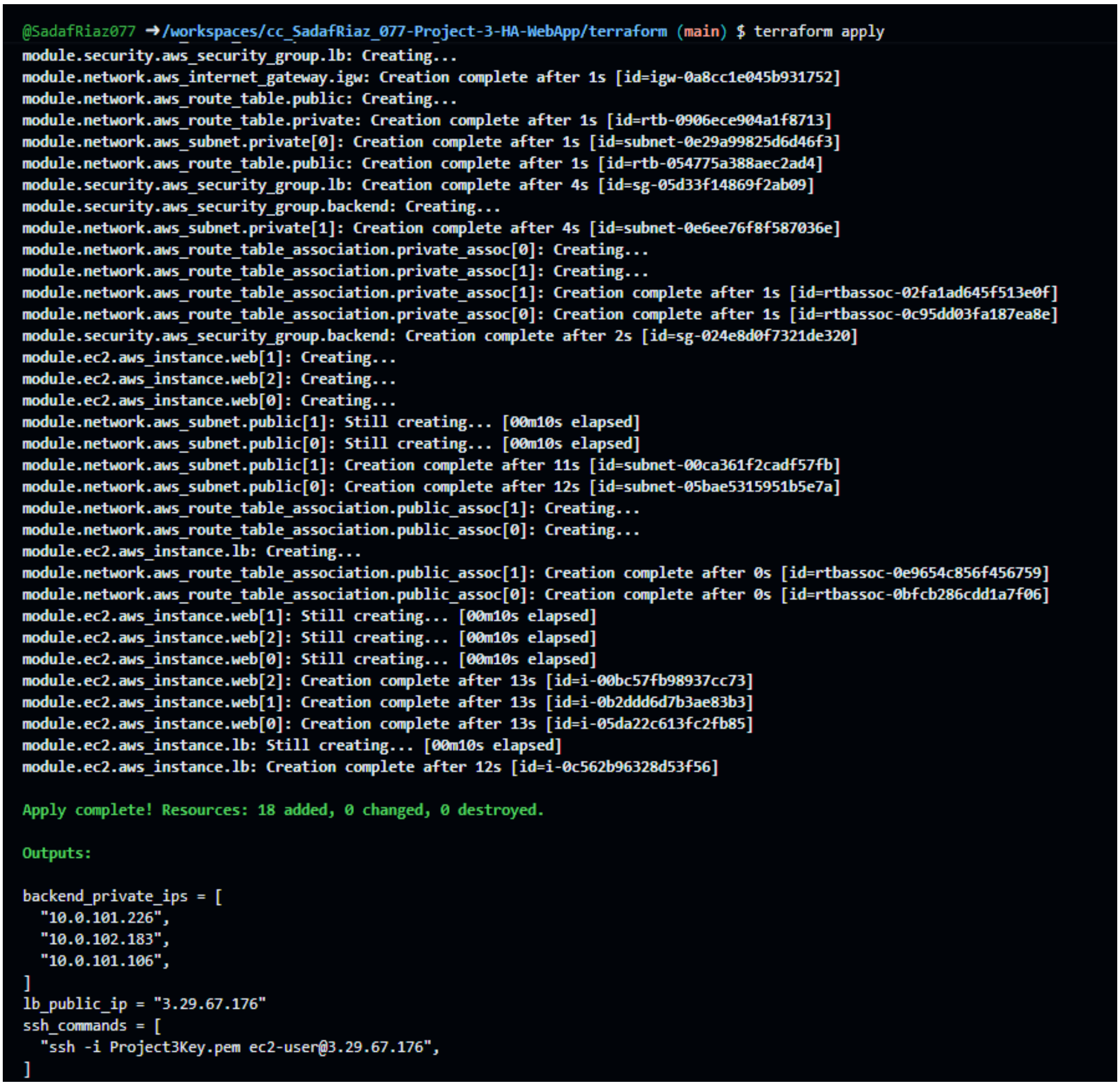
```
● @SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ cat variables.tf
variable "aws_region" {
  description = "AWS region to deploy"
  type        = string
}

variable "project_name" {
  description = "Project name"
  type        = string
}

variable "environment" {
  description = "Deployment environment (dev, prod, etc.)"
  type        = string
}

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

variable "public_subnet_cidr_blocks" {
  description = "List of public subnet CIDR blocks"
  type        = list(string)
}

variable "private_subnet_cidr_blocks" {
  description = "List of private subnet CIDR blocks"
  type        = list(string)
}

variable "availability_zones" {
  description = "List of 2 AZs"
  type        = list(string)
}

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

variable "ami_id" {
  description = "AMI ID for EC2 instances"
  type        = string
}

variable "key_name" {
  description = "Name of SSH key pair"
  type        = string
}

variable "my_ip" {
  description = "Your public IP for SSH access"
  type        = string
```

cat terraform.tfvars



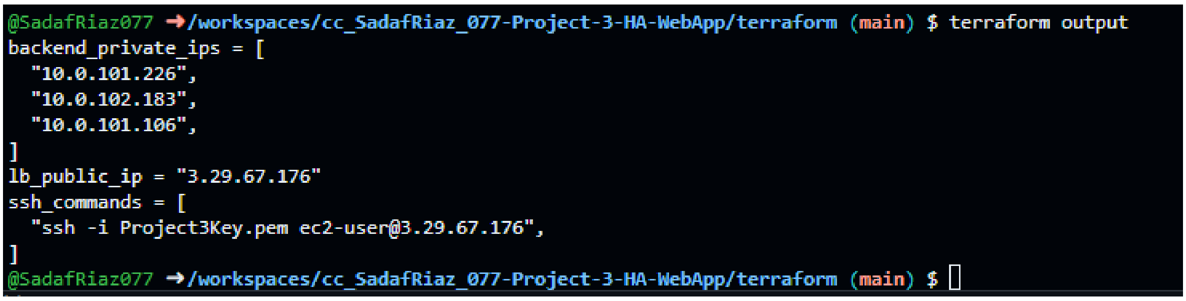
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ nano terraform.tfvars
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ cat  terraform.tfvars
aws_region                = "me-central-1"
project_name              = "Project3"
environment               = "dev"
vpc_cidr_block            = "10.0.0.0/16"
public_subnet_cidr_blocks = ["10.0.1.0/24", "10.0.2.0/24"]
private_subnet_cidr_blocks = ["10.0.101.0/24", "10.0.102.0/24"]
availability_zones        = ["me-central-1a", "me-central-1b"]
instance_type             = "t3.micro"
ami_id                    = "ami-082c959b457b42cf9"
key_name                  = "Project3Key"
my_ip                     = "20.192.21.54"

@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ |
```

Terraform apply:



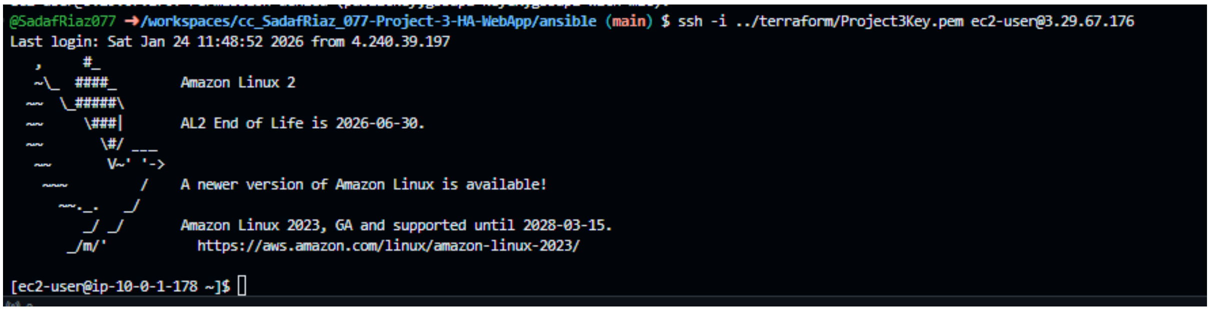
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ terraform apply
module.security.aws_security_group.lb: Creating...
module.network.aws_internet_gateway.igw: Creation complete after 1s [id=igw-0a8cc1e045b931752]
module.network.aws_route_table.public: Creating...
module.network.aws_route_table.private: Creation complete after 1s [id=rtb-0906ece904a1f8713]
module.network.aws_subnet.private[0]: Creation complete after 1s [id=subnet-0e29a99825d6d46f3]
module.network.aws_route_table.public: Creation complete after 1s [id=rtb-054775a388aec2ad4]
module.security.aws_security_group.lb: Creation complete after 4s [id=sg-05d33f14869f2ab09]
module.security.aws_security_group.backend: Creating...
module.network.aws_subnet.private[1]: Creation complete after 4s [id=subnet-0e6ee76f8f587036e]
module.network.aws_route_table_association.private_assoc[0]: Creating...
module.network.aws_route_table_association.private_assoc[1]: Creating...
module.network.aws_route_table_association.private_assoc[1]: Creation complete after 1s [id=rtbassoc-02fa1ad645f513e0f]
module.network.aws_route_table_association.private_assoc[0]: Creation complete after 1s [id=rtbassoc-0c95dd03fa187ea8e]
module.security.aws_security_group.backend: Creation complete after 2s [id=sg-024e8d0f7321de320]
module.ec2.aws_instance.web[1]: Creating...
module.ec2.aws_instance.web[2]: Creating...
module.ec2.aws_instance.web[0]: Creating...
module.network.aws_subnet.public[1]: Still creating... [00m10s elapsed]
module.network.aws_subnet.public[0]: Still creating... [00m10s elapsed]
module.network.aws_subnet.public[1]: Creation complete after 11s [id=subnet-00ca361f2cadf57fb]
module.network.aws_subnet.public[0]: Creation complete after 12s [id=subnet-05bae5315951b5e7a]
module.network.aws_route_table_association.public_assoc[1]: Creating...
module.network.aws_route_table_association.public_assoc[0]: Creating...
module.ec2.aws_instance.lb: Creating...
module.network.aws_route_table_association.public_assoc[1]: Creation complete after 0s [id=rtbassoc-0e9654c856f456759]
module.network.aws_route_table_association.public_assoc[0]: Creation complete after 0s [id=rtbassoc-0bfcb286cdd1a7f06]
module.ec2.aws_instance.web[1]: Still creating... [00m10s elapsed]
module.ec2.aws_instance.web[2]: Still creating... [00m10s elapsed]
module.ec2.aws_instance.web[0]: Still creating... [00m10s elapsed]
module.ec2.aws_instance.web[2]: Creation complete after 13s [id=i-00bc57fb98937cc73]
module.ec2.aws_instance.web[1]: Creation complete after 13s [id=i-0b2ddd6d7b3ae83b3]
module.ec2.aws_instance.web[0]: Creation complete after 13s [id=i-05da22c613fc2fb85]
module.ec2.aws_instance.lb: Still creating... [00m10s elapsed]
module.ec2.aws_instance.lb: Creation complete after 12s [id=i-0c562b96328d53f56]

Apply complete! Resources: 18 added, 0 changed, 0 destroyed.

Outputs:

backend_private_ips = [
  "10.0.101.226",
  "10.0.102.183",
  "10.0.101.106",
]
lb_public_ip = "3.29.67.176"
ssh_commands = [
  "ssh -i Project3Key.pem ec2-user@3.29.67.176",
]
```

Terraform Output:



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ terraform output
backend_private_ips = [
  "10.0.101.226",
  "10.0.102.183",
  "10.0.101.106",
]
lb_public_ip = "3.29.67.176"
ssh_commands = [
  "ssh -i Project3Key.pem ec2-user@3.29.67.176",
]
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/terraform (main) $ 
```

Ec2

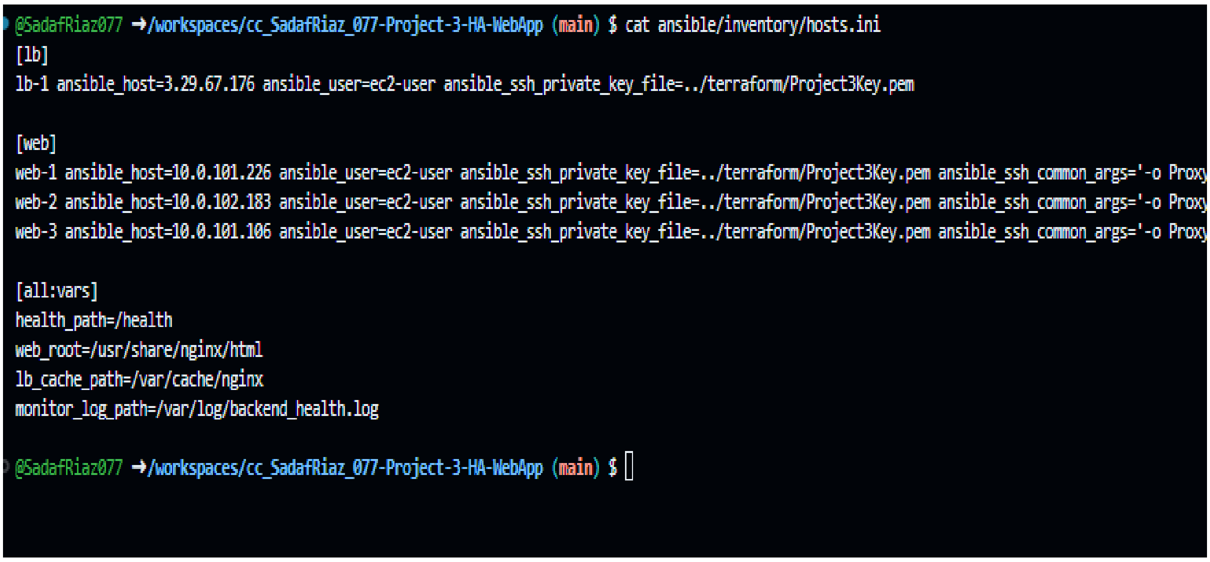


```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ssh -i ../terraform/Project3Key.pem ec2-user@3.29.67.176
Last login: Sat Jan 24 11:48:52 2026 from 4.240.39.197
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-1-178 ~]$ 
```

4. Ansible – Nginx LB, Backends, SSL, Caching, Health, Monitoring

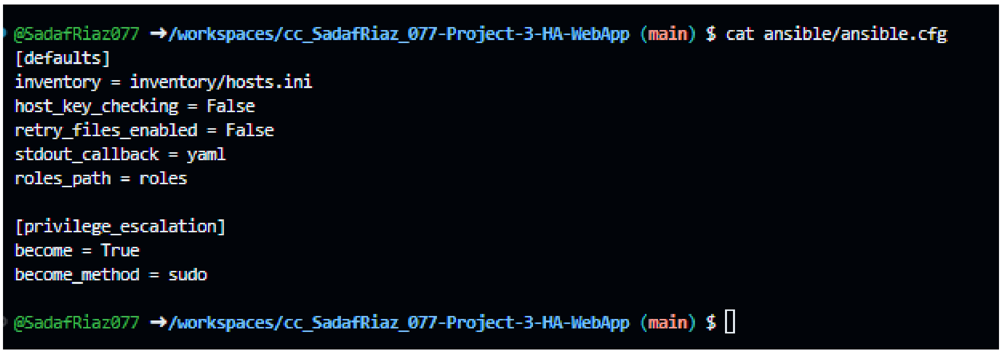
4.1 – Ansible Inventory & Configuration

ansible/inventory/hosts.ini



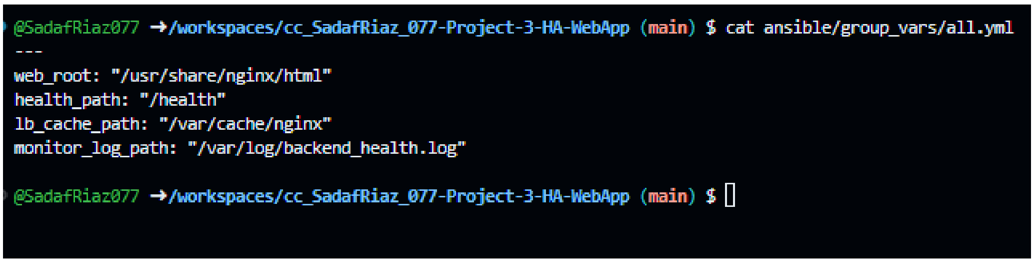
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat ansible/inventory/hosts.ini
[lb]
lb-1 ansible_host=3.29.67.176 ansible_user=ec2-user ansible_ssh_private_key_file=../terraform/Project3Key.pem

[web]
web-1 ansible_host=10.0.101.226 ansible_user=ec2-user ansible_ssh_private_key_file=../terraform/Project3Key.pem ansible_ssh_common_args='-o Proxy
web-2 ansible_host=10.0.102.183 ansible_user=ec2-user ansible_ssh_private_key_file=../terraform/Project3Key.pem ansible_ssh_common_args='-o Proxy
web-3 ansible_host=10.0.101.106 ansible_user=ec2-user ansible_ssh_private_key_file=../terraform/Project3Key.pem ansible_ssh_common_args='-o Proxy

[all:vars]
health_path=/health
web_root=/usr/share/nginx/html
lb_cache_path=/var/cache/nginx
monitor_log_path=/var/log/backend_health.log

@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ 
```

ansible/ansible.cfg



```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat ansible/ansible.cfg
[defaults]
inventory = inventory/hosts.ini
host_key_checking = False
retry_files_enabled = False
stdout_callback = yaml
roles_path = roles

[privilege_escalation]
become = True
become_method = sudo

@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ 
```

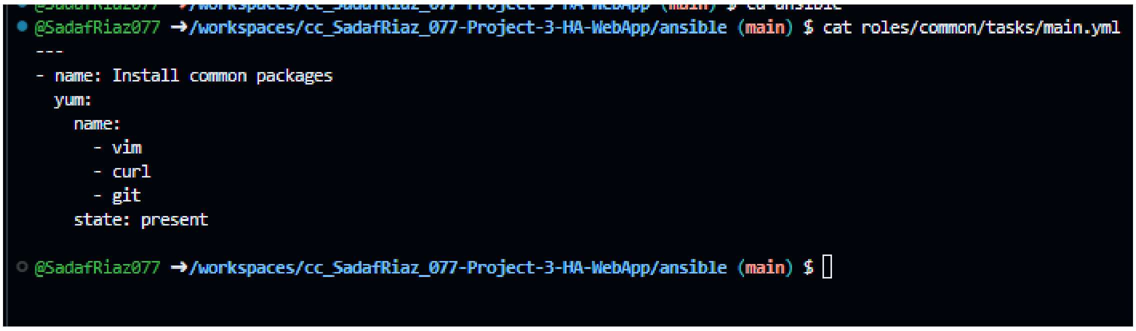
ansible/group_vars/all.yml



```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat ansible/group_vars/all.yml
---
web_root: "/usr/share/nginx/html"
health_path: "/health"
lb_cache_path: "/var/cache/nginx"
monitor_log_path: "/var/log/backend_health.log"

@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ 
```

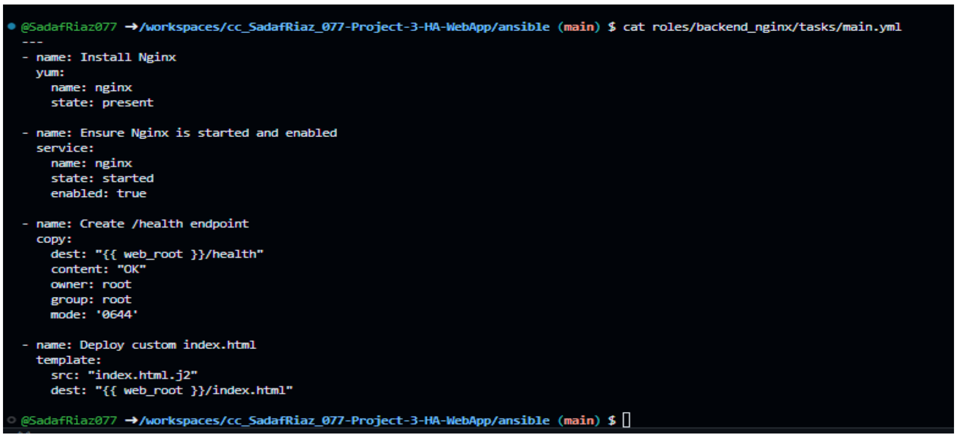
4.2 – Backend Nginx Role + App Deployment

roles/common/tasks/main.yml



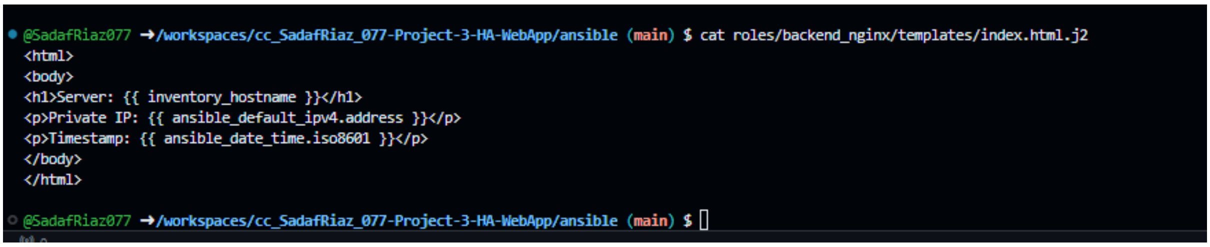
```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cd ansible
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ cat roles/common/tasks/main.yml
---
- name: Install common packages
  yum:
    name:
      - vim
      - curl
      - git
    state: present

@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ 
```

roles/backend_nginx/tasks/main.yml



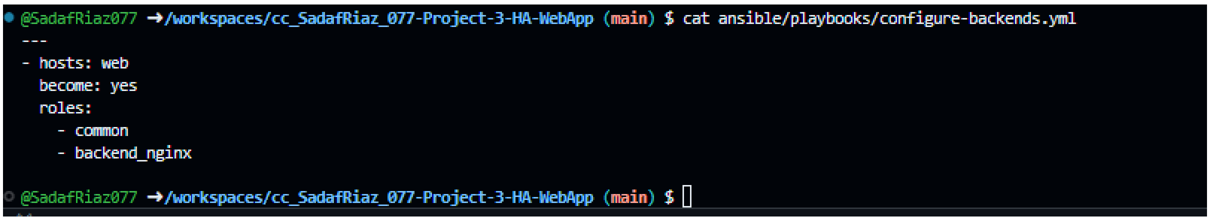
```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ cat roles/backend_nginx/tasks/main.yml
---
- name: Install Nginx
  yum:
    name: nginx
    state: present

- name: Ensure Nginx is started and enabled
  service:
    name: nginx
    state: started
    enabled: true

- name: Create /health endpoint
  copy:
    dest: "{{ web_root }}/health"
    content: "OK"
    owner: root
    group: root
    mode: '0644'

- name: Deploy custom index.html
  template:
    src: "index.html.j2"
    dest: "{{ web_root }}/index.html"

@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ 
```

roles/backend_nginx/templates/index.html.j2



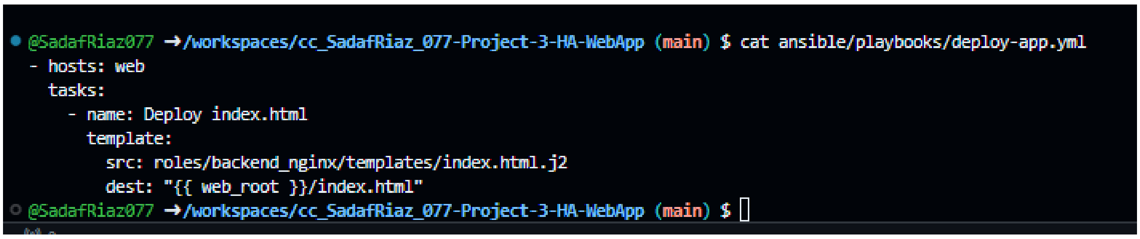
```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ cat roles/backend_nginx/templates/index.html.j2
<html>
<body>
<h1>Server: {{ inventory_hostname }}</h1>
<p>Private IP: {{ ansible_default_ipv4.address }}</p>
<p>Timestamp: {{ ansible_date_time.iso8601 }}</p>
</body>
</html>

@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $
```

ansible/playbooks/configure-backends.yml



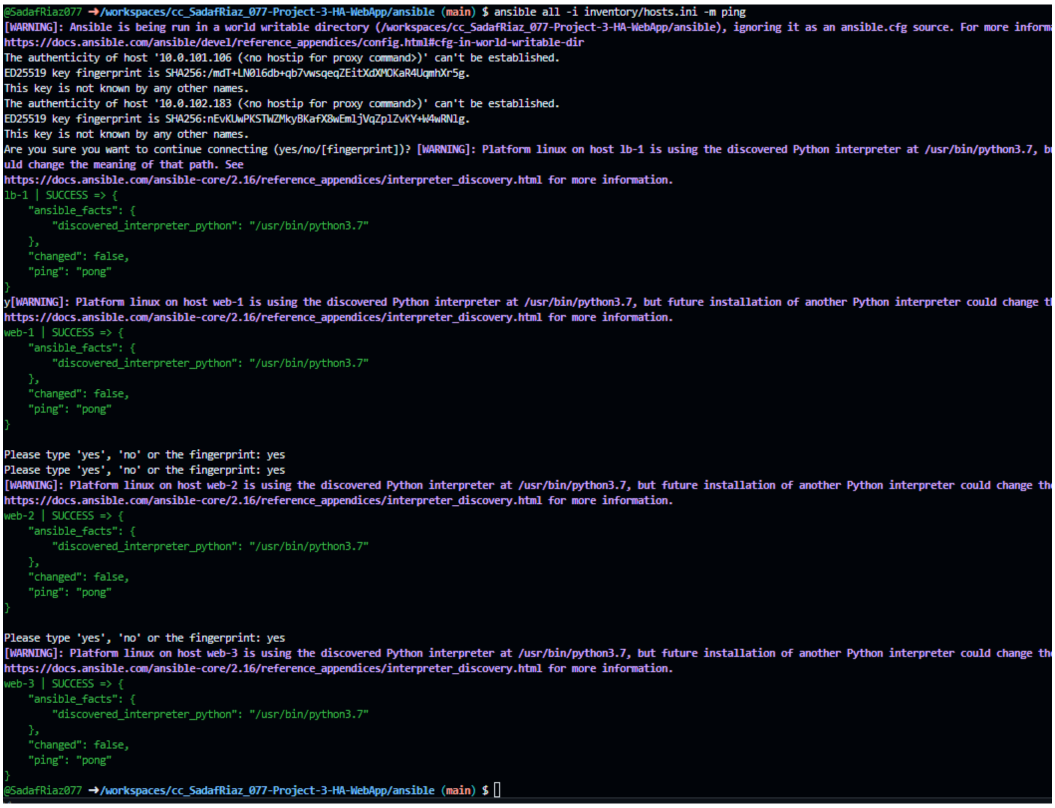
```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat ansible/playbooks/configure-backends.yml
---
- hosts: web
  become: yes
  roles:
    - common
    - backend_nginx

@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $
```

ansible/playbooks/deploy-app.yml



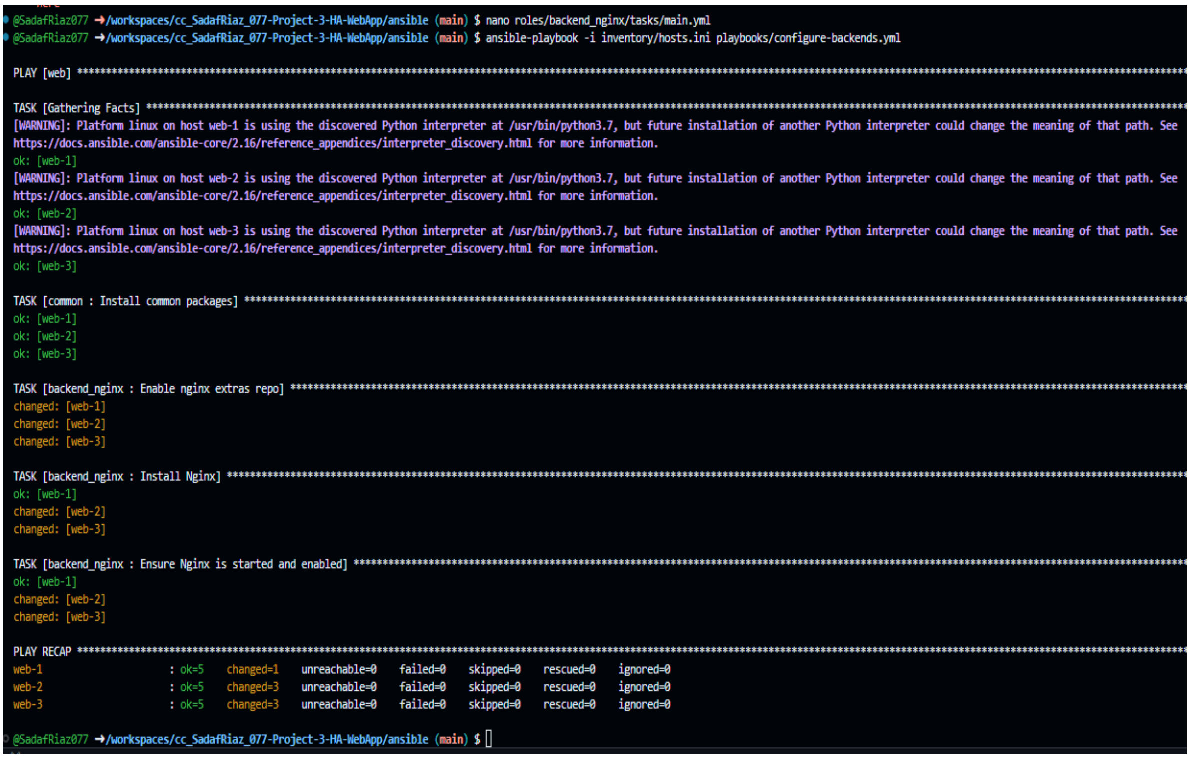
```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $ cat ansible/playbooks/deploy-app.yml
- hosts: web
  tasks:
    - name: Deploy index.html
      template:
        src: roles/backend_nginx/templates/index.html.j2
        dest: "{{ web_root }}/index.html"
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp (main) $
```

Ping



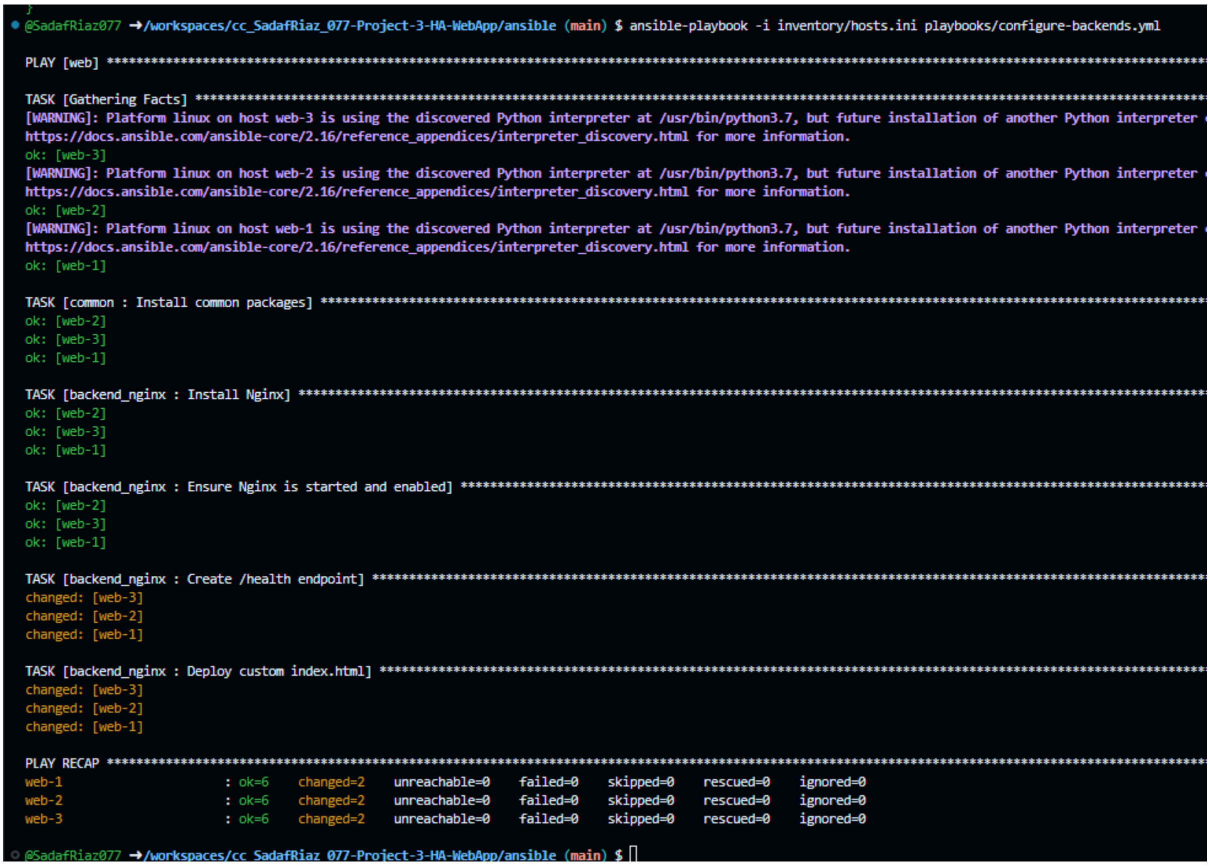
```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ansible all -i inventory/hosts.ini -m ping
[WARNING]: Ansible is being run in a world writable directory (/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible), ignoring it as an ansible.cfg source. For more inform
https://docs.ansible.com/ansible/devel/reference_appendices/config.html#cfg-in-world-writable-dir
The authenticity of host '10.0.101.106 (<no hostip for proxy command>)' can't be established.
ED25519 key fingerprint is SHA256:/mdT+LN016db+qb7vwsqeqZEitXdXMOKaR4UqmhXr5g.
This key is not known by any other names.
The authenticity of host '10.0.102.183 (<no hostip for proxy command>)' can't be established.
ED25519 key fingerprint is SHA256:nEvKUwPKSTWZMkyBKafX8wEmljVqZplZvKY+W4wRNlg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? [WARNING]: Platform linux on host lb-1 is using the discovered Python interpreter at /usr/bin/python3.7, b
uld change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
lb-1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
y[WARNING]: Platform linux on host web-1 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter could change t
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
web-1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}

Please type 'yes', 'no' or the fingerprint: yes
Please type 'yes', 'no' or the fingerprint: yes
[WARNING]: Platform linux on host web-2 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter could change th
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
web-2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}

Please type 'yes', 'no' or the fingerprint: yes
[WARNING]: Platform linux on host web-3 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter could change th
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
web-3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $
```

Backend Configuration Playbook Execution



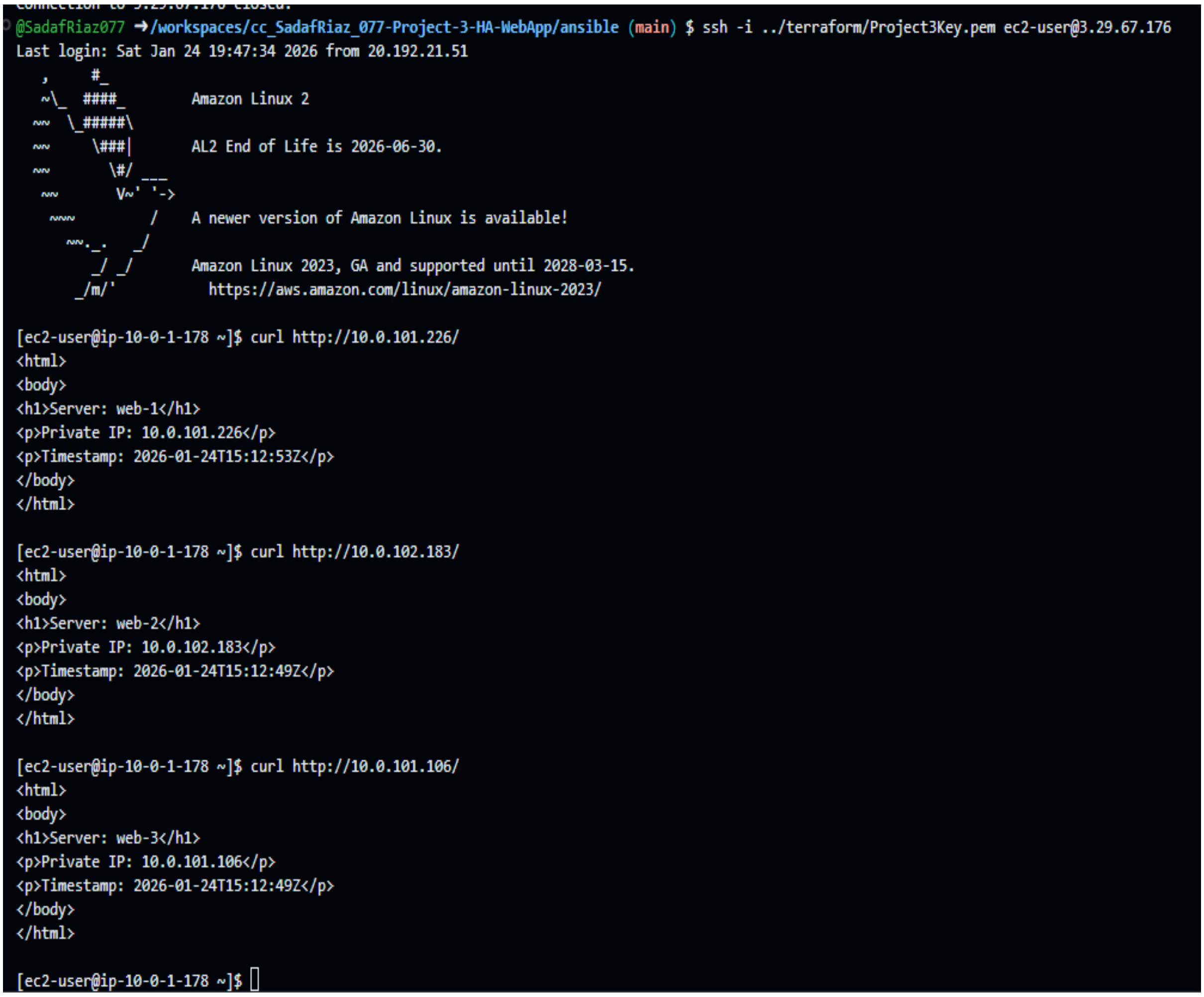
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ nano roles/backend_nginx/tasks/main.yml
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ansible-playbook -i inventory/hosts.ini playbooks/configure-backends.yml

PLAY [web] ************************************************************************************************************************************

TASK [Gathering Facts] ************************************************************************************************************************
[WARNING]: Platform linux on host web-1 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [web-1]
[WARNING]: Platform linux on host web-2 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [web-2]
[WARNING]: Platform linux on host web-3 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [web-3]

TASK [common : Install common packages] *******************************************************************************************************
ok: [web-1]
ok: [web-2]
ok: [web-3]

TASK [backend_nginx : Enable nginx extras repo] ***********************************************************************************************
changed: [web-1]
changed: [web-2]
changed: [web-3]

TASK [backend_nginx : Install Nginx] **********************************************************************************************************
ok: [web-1]
changed: [web-2]
changed: [web-3]

TASK [backend_nginx : Ensure Nginx is started and enabled] ************************************************************************************
ok: [web-1]
changed: [web-2]
changed: [web-3]

PLAY RECAP ************************************************************************************************************************************
web-1                      : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web-2                      : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web-3                      : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ 
```

Application Deployment Playbook Execution



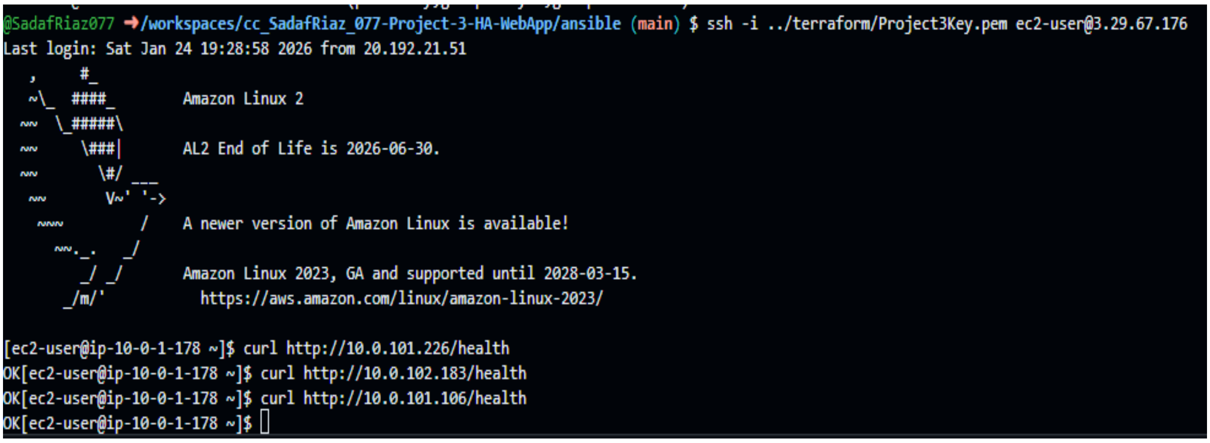
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ansible-playbook -i inventory/hosts.ini playbooks/configure-backends.yml

PLAY [web] ********************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
[WARNING]: Platform linux on host web-3 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [web-3]
[WARNING]: Platform linux on host web-2 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [web-2]
[WARNING]: Platform linux on host web-1 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [web-1]

TASK [common : Install common packages] ***************************************************************************************
ok: [web-2]
ok: [web-3]
ok: [web-1]

TASK [backend_nginx : Install Nginx] ******************************************************************************************
ok: [web-2]
ok: [web-3]
ok: [web-1]

TASK [backend_nginx : Ensure Nginx is started and enabled] ********************************************************************
ok: [web-2]
ok: [web-3]
ok: [web-1]

TASK [backend_nginx : Create /health endpoint] ********************************************************************************
changed: [web-3]
changed: [web-2]
changed: [web-1]

TASK [backend_nginx : Deploy custom index.html] *******************************************************************************
changed: [web-3]
changed: [web-2]
changed: [web-1]

PLAY RECAP ********************************************************************************************************************
web-1                      : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web-2                      : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
web-3                      : ok=6    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ 
```

Backend and Application Verification Using curl



```
Connection to 3.29.67.176 closed.
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ssh -i ../terraform/Project3Key.pem ec2-user@3.29.67.176
Last login: Sat Jan 24 19:47:34 2026 from 20.192.21.51
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-1-178 ~]$ curl http://10.0.101.226/
<html>
<body>
<h1>Server: web-1</h1>
<p>Private IP: 10.0.101.226</p>
<p>Timestamp: 2026-01-24T15:12:53Z</p>
</body>
</html>

[ec2-user@ip-10-0-1-178 ~]$ curl http://10.0.102.183/
<html>
<body>
<h1>Server: web-2</h1>
<p>Private IP: 10.0.102.183</p>
<p>Timestamp: 2026-01-24T15:12:49Z</p>
</body>
</html>

[ec2-user@ip-10-0-1-178 ~]$ curl http://10.0.101.106/
<html>
<body>
<h1>Server: web-3</h1>
<p>Private IP: 10.0.101.106</p>
<p>Timestamp: 2026-01-24T15:12:49Z</p>
</body>
</html>

[ec2-user@ip-10-0-1-178 ~]$ 
```

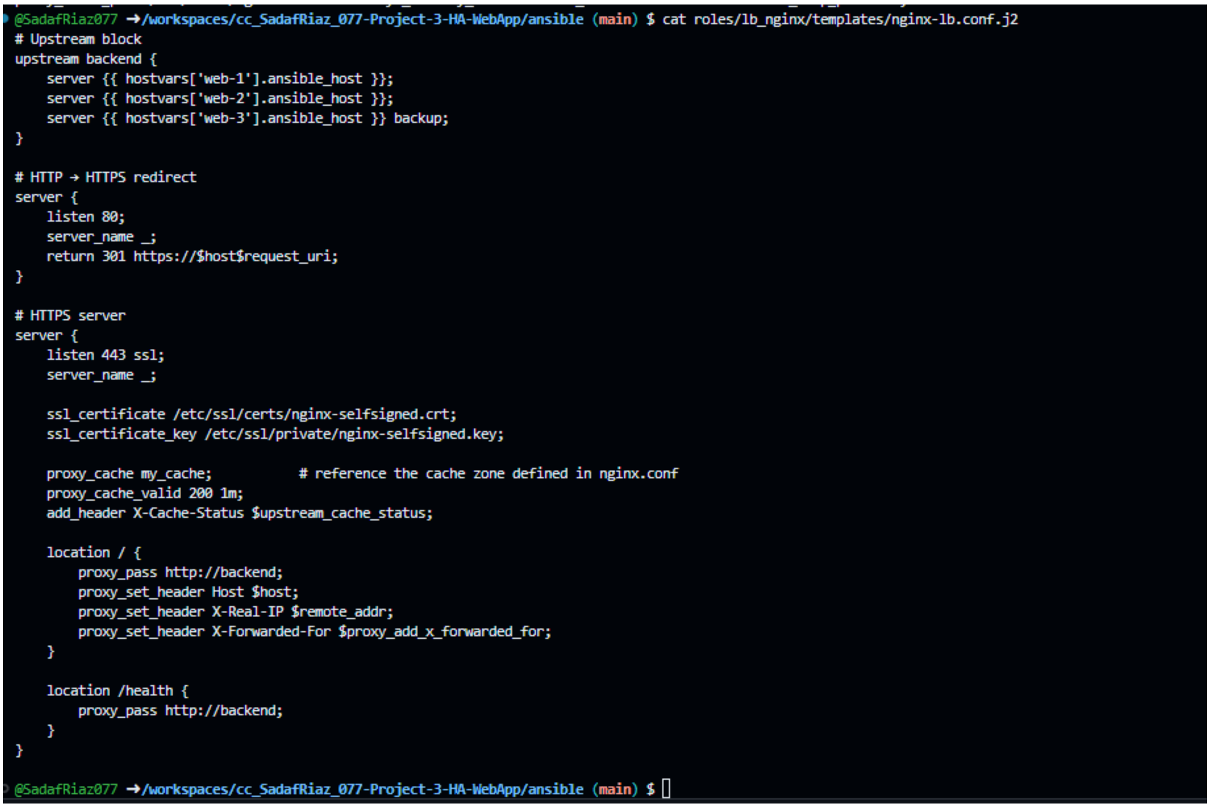
Backend Health Verification Using curl



```
@SadafRiaz077 ➜ /workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ssh -i ../terraform/Project3Key.pem ec2-user@3.29.67.176
Last login: Sat Jan 24 19:28:58 2026 from 20.192.21.51
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-1-178 ~]$ curl http://10.0.101.226/health
OK[ec2-user@ip-10-0-1-178 ~]$ curl http://10.0.102.183/health
OK[ec2-user@ip-10-0-1-178 ~]$ curl http://10.0.101.106/health
OK[ec2-user@ip-10-0-1-178 ~]$ 
```

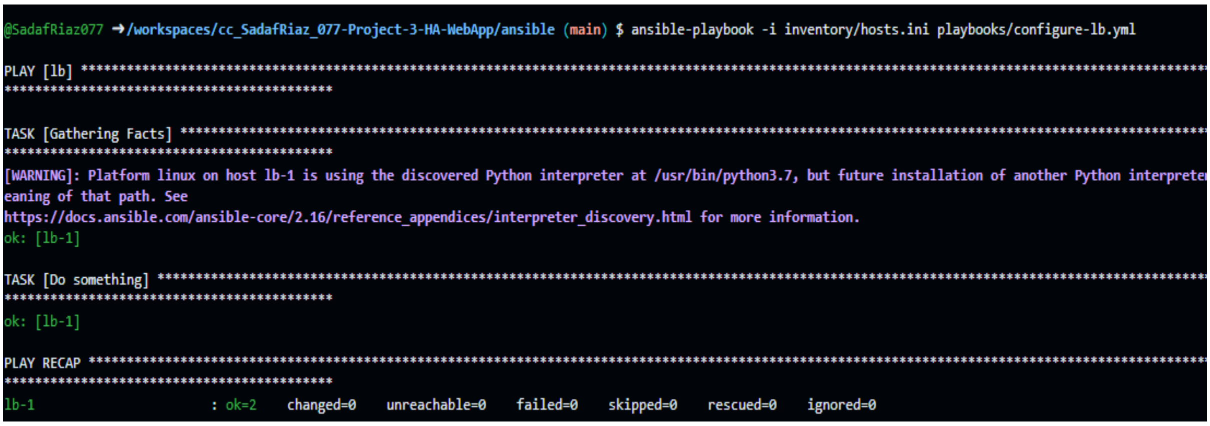
4.3 Load Balancer Nginx Role (SSL + LB + Backup + Cache)

cat roles/lb-nginx-lb.conf.j2



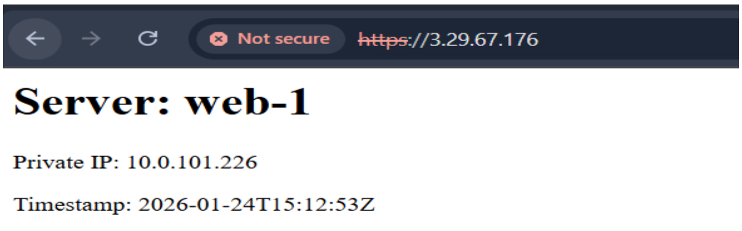
```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ cat roles/lb_nginx/templates/nginx-lb.conf.j2
# Upstream block
upstream backend {
    server {{ hostvars['web-1'].ansible_host }};
    server {{ hostvars['web-2'].ansible_host }};
    server {{ hostvars['web-3'].ansible_host }} backup;
}

# HTTP → HTTPS redirect
server {
    listen 80;
    server_name _;
    return 301 https://$host$request_uri;
}

# HTTPS server
server {
    listen 443 ssl;
    server_name _;

    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;

    proxy_cache my_cache;          # reference the cache zone defined in nginx.conf
    proxy_cache_valid 200 1m;
    add_header X-Cache-Status $upstream_cache_status;

    location / {
        proxy_pass http://backend;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location /health {
        proxy_pass http://backend;
    }
}

@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $
```

Load Balancer Configuration Playbook Execution



```
@SadafRiaz077 →/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ansible-playbook -i inventory/hosts.ini playbooks/configure-lb.yml

PLAY [lb] ***********************************************************************************************************************
*******************************************

TASK [Gathering Facts] **********************************************************************************************************
*******************************************
[WARNING]: Platform linux on host lb-1 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interprete
eaning of that path. See
https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [lb-1]

TASK [Do something] *************************************************************************************************************
*******************************************
ok: [lb-1]

PLAY RECAP **********************************************************************************************************************
*******************************************
lb-1                       : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

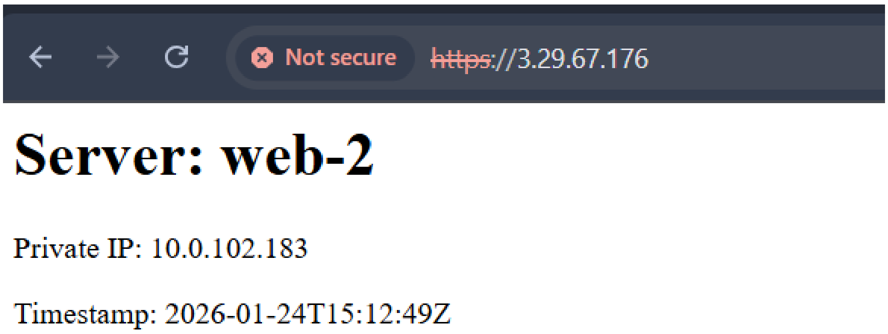
Test on Browser: web 1 & web 2



Server: web-1

Private IP: 10.0.101.226

Timestamp: 2026-01-24T15:12:53Z



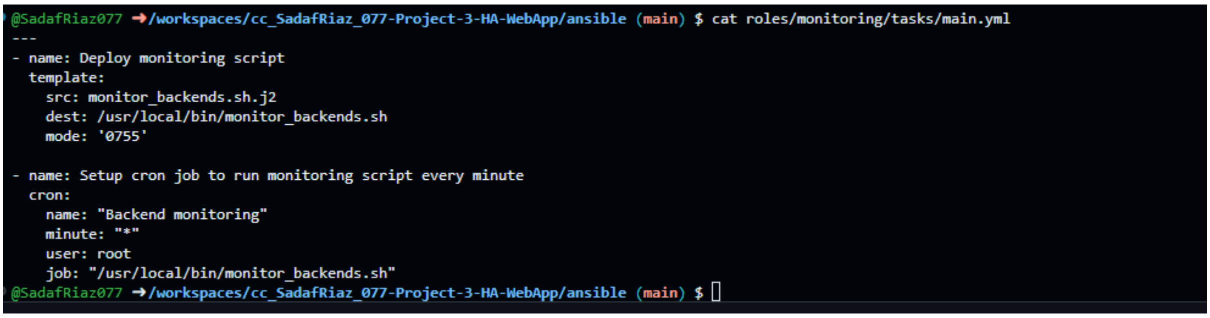
~~https~~://3.29.67.176

Server: web-2

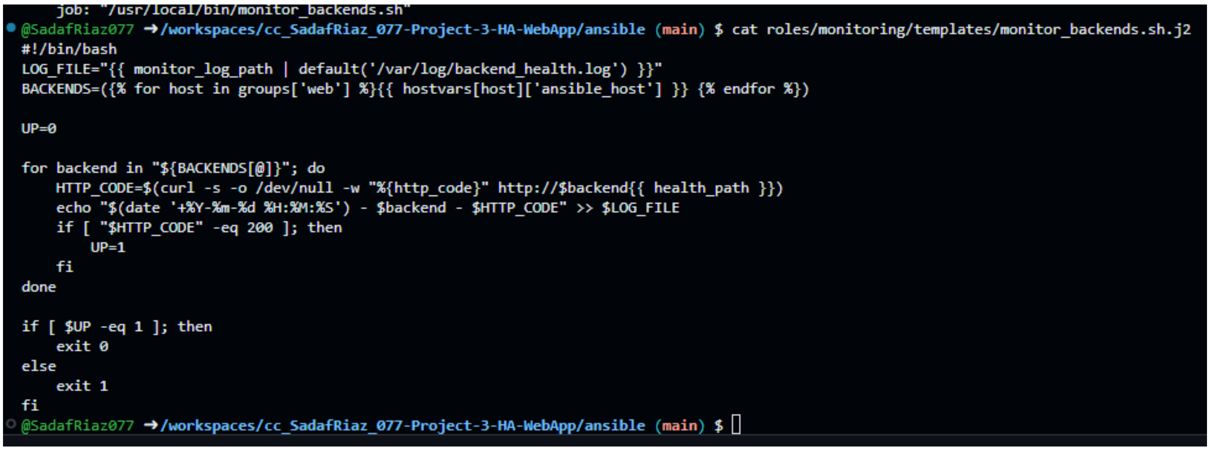
Private IP: 10.0.102.183

Timestamp: 2026-01-24T15:12:49Z

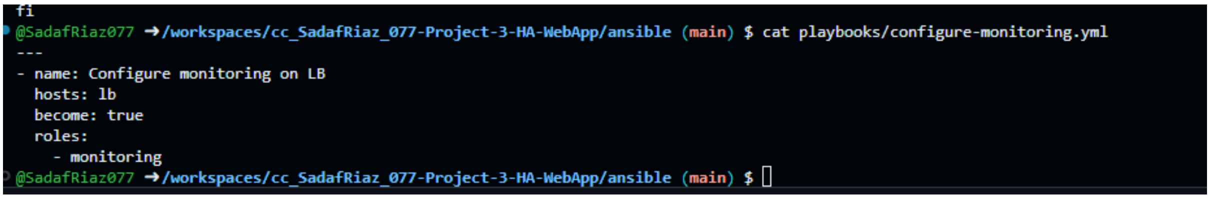
4.4 Monitoring Role (LB Monitoring Script)



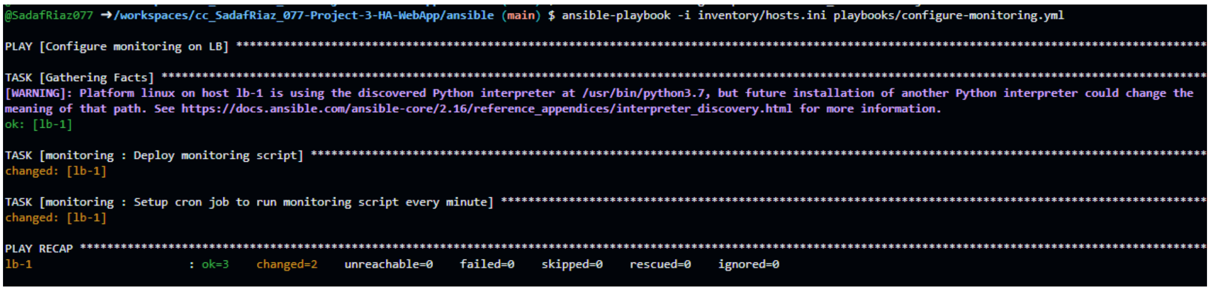
```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ cat roles/monitoring/tasks/main.yml
---
- name: Deploy monitoring script
  template:
    src: monitor_backends.sh.j2
    dest: /usr/local/bin/monitor_backends.sh
    mode: '0755'

- name: Setup cron job to run monitoring script every minute
  cron:
    name: "Backend monitoring"
    minute: "*"
    user: root
    job: "/usr/local/bin/monitor_backends.sh"
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ 
```



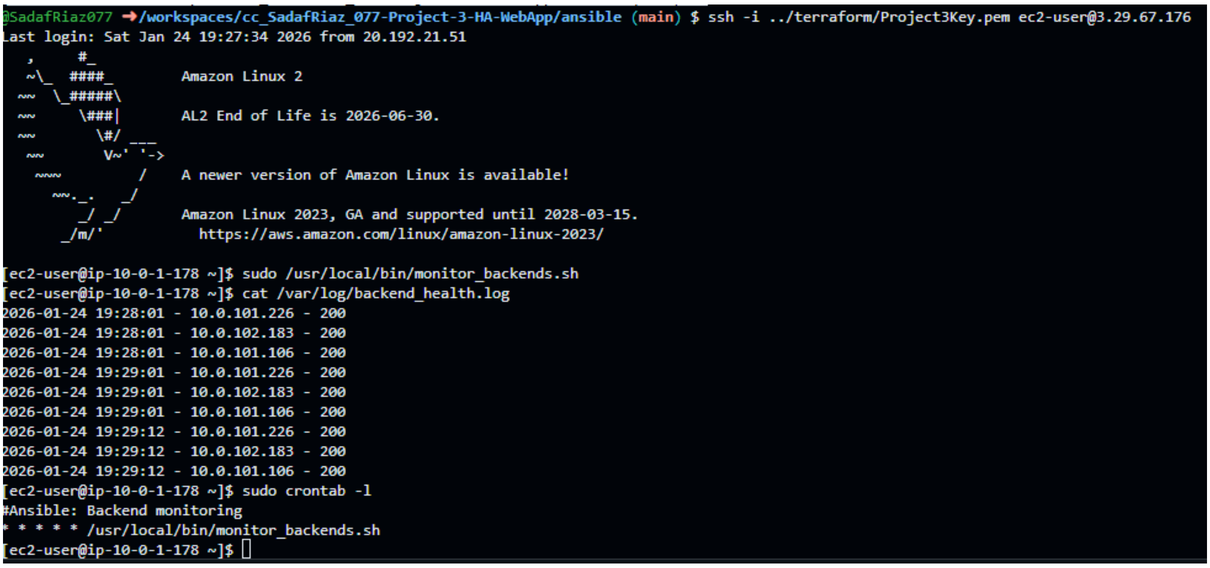
```
    job: "/usr/local/bin/monitor_backends.sh"
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ cat roles/monitoring/templates/monitor_backends.sh.j2
#!/bin/bash
LOG_FILE="{{ monitor_log_path | default('/var/log/backend_health.log') }}"
BACKENDS=({% for host in groups['web'] %}{{ hostvars[host]['ansible_host'] }} {% endfor %})

UP=0

for backend in "${BACKENDS[@]}"; do
    HTTP_CODE=$(curl -s -o /dev/null -w "%{http_code}" http://$backend{{ health_path }})
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $backend - $HTTP_CODE" >> $LOG_FILE
    if [ "$HTTP_CODE" -eq 200 ]; then
        UP=1
    fi
done

if [ $UP -eq 1 ]; then
    exit 0
else
    exit 1
fi
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ 
```



```
fi
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ cat playbooks/configure-monitoring.yml
---
- name: Configure monitoring on LB
  hosts: lb
  become: true
  roles:
    - monitoring
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ 
```



```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ansible-playbook -i inventory/hosts.ini playbooks/configure-monitoring.yml

PLAY [Configure monitoring on LB] **********************************************************************************************

TASK [Gathering Facts] *********************************************************************************************************
[WARNING]: Platform linux on host lb-1 is using the discovered Python interpreter at /usr/bin/python3.7, but future installation of another Python interpreter could change the
meaning of that path. See https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more information.
ok: [lb-1]

TASK [monitoring : Deploy monitoring script] ***********************************************************************************
changed: [lb-1]

TASK [monitoring : Setup cron job to run monitoring script every minute] *******************************************************
changed: [lb-1]

PLAY RECAP *********************************************************************************************************************
lb-1                       : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

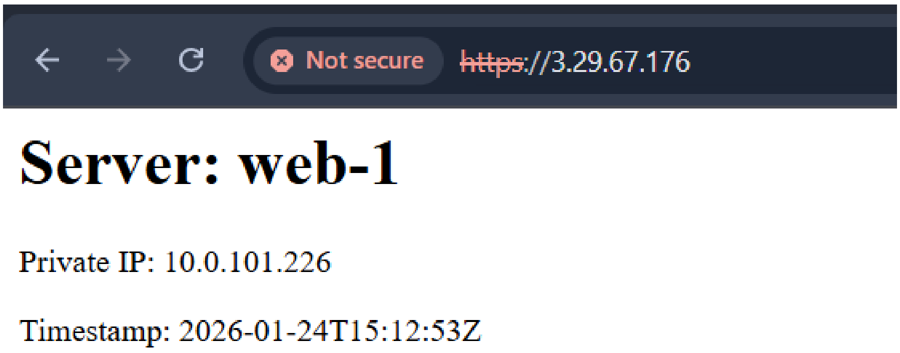


```
@SadafRiaz077 ➜/workspaces/cc_SadafRiaz_077-Project-3-HA-WebApp/ansible (main) $ ssh -i ../terraform/Project3Key.pem ec2-user@3.29.67.176
Last login: Sat Jan 24 19:27:34 2026 from 20.192.21.51
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-1-178 ~]$ sudo /usr/local/bin/monitor_backends.sh
[ec2-user@ip-10-0-1-178 ~]$ cat /var/log/backend_health.log
2026-01-24 19:28:01 - 10.0.101.226 - 200
2026-01-24 19:28:01 - 10.0.102.183 - 200
2026-01-24 19:28:01 - 10.0.101.106 - 200
2026-01-24 19:29:01 - 10.0.101.226 - 200
2026-01-24 19:29:01 - 10.0.102.183 - 200
2026-01-24 19:29:01 - 10.0.101.106 - 200
2026-01-24 19:29:12 - 10.0.101.226 - 200
2026-01-24 19:29:12 - 10.0.102.183 - 200
2026-01-24 19:29:12 - 10.0.101.106 - 200
[ec2-user@ip-10-0-1-178 ~]$ sudo crontab -l
#Ansible: Backend monitoring
* * * * * /usr/local/bin/monitor_backends.sh
[ec2-user@ip-10-0-1-178 ~]$ 
```

5.High Availability, Caching, SSL & Monitoring Testing

5.1 Failover Test (Backup Server)

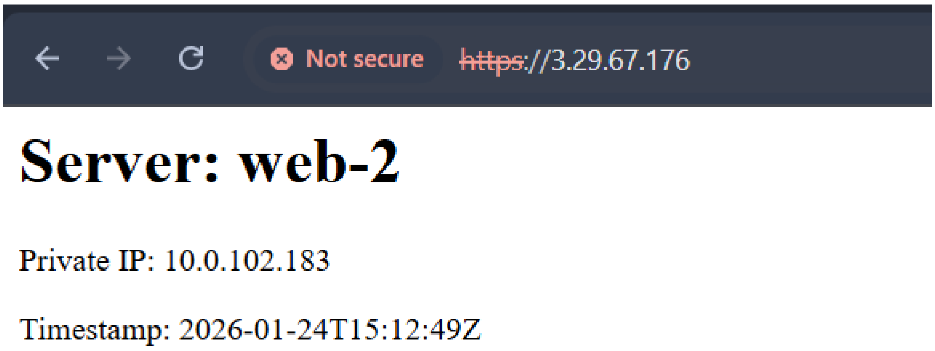
Traffic Load Balancing Between Web Server 1 and Web Server 2



Server: web-1

Private IP: 10.0.101.226

Timestamp: 2026-01-24T15:12:53Z

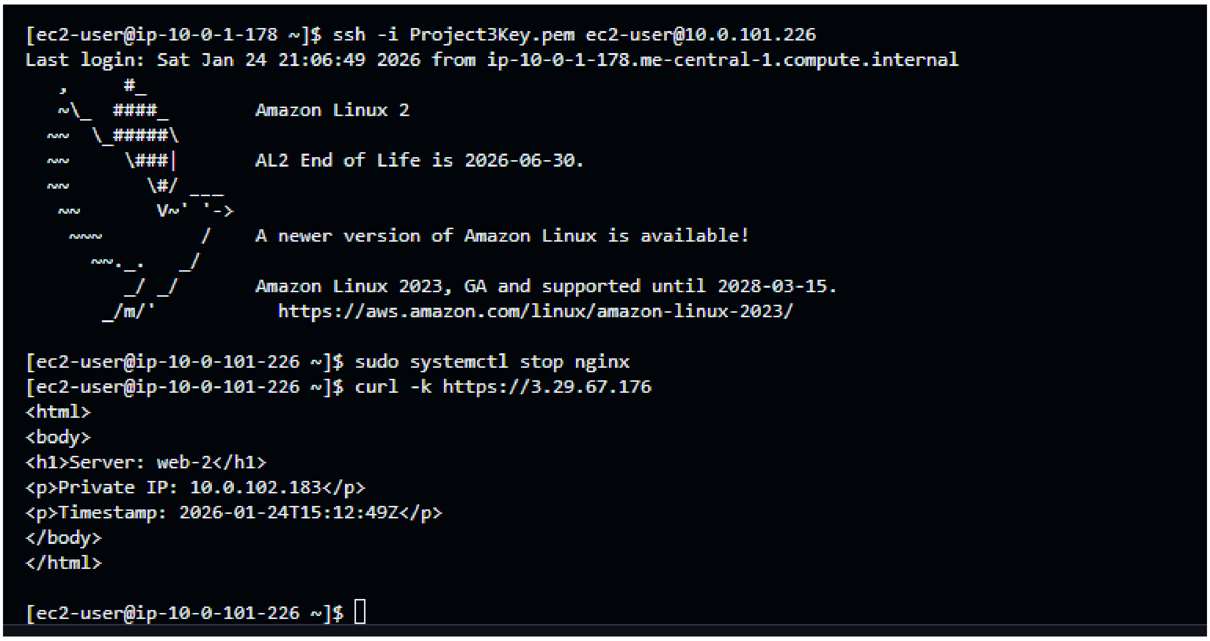


Server: web-2

Private IP: 10.0.102.183

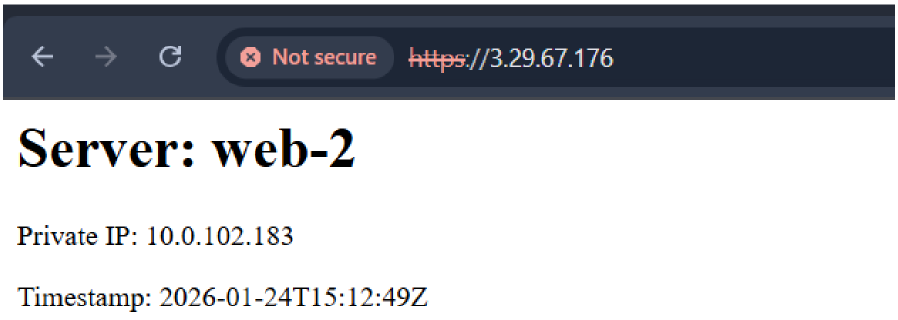
Timestamp: 2026-01-24T15:12:49Z

Nginx Service Stopped on Web Server 1



```
[ec2-user@ip-10-0-1-178 ~]$ ssh -i Project3Key.pem ec2-user@10.0.101.226
Last login: Sat Jan 24 21:06:49 2026 from ip-10-0-1-178.me-central-1.compute.internal
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-101-226 ~]$ sudo systemctl stop nginx
[ec2-user@ip-10-0-101-226 ~]$ curl -k https://3.29.67.176
<html>
<body>
<h1>Server: web-2</h1>
<p>Private IP: 10.0.102.183</p>
<p>Timestamp: 2026-01-24T15:12:49Z</p>
</body>
</html>

[ec2-user@ip-10-0-101-226 ~]$ 
```

Show only web 2

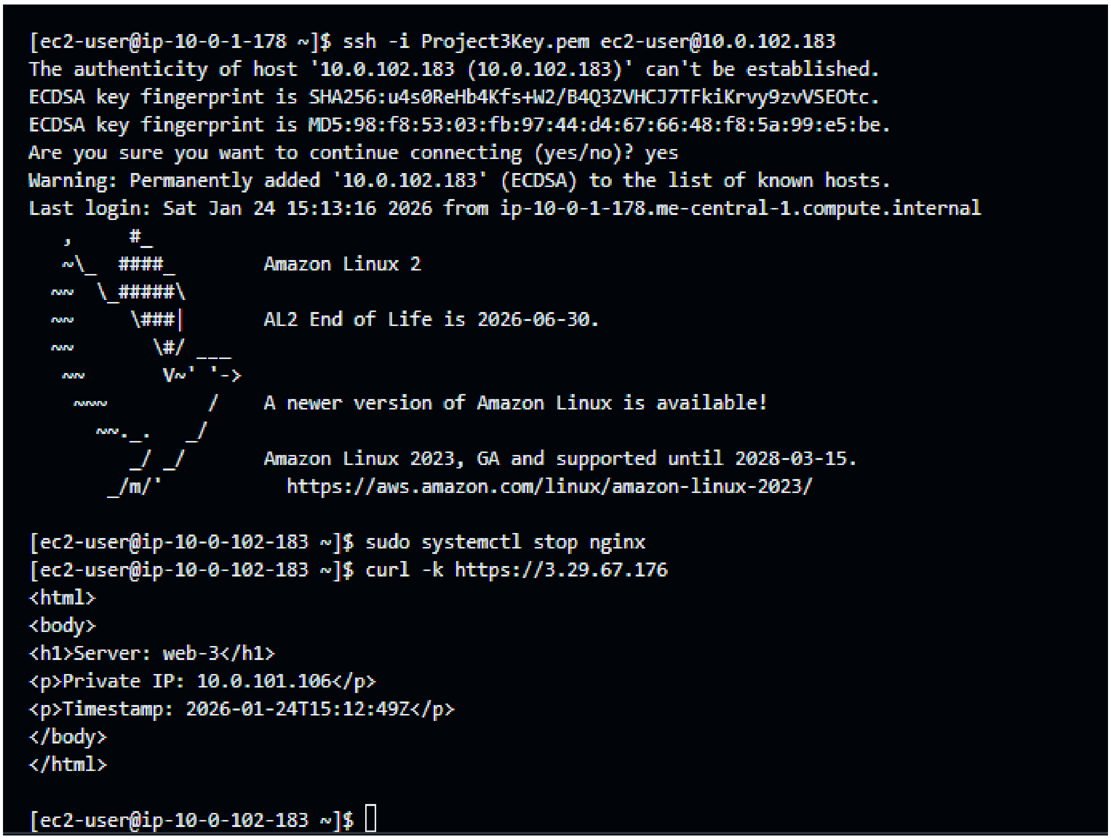


Server: web-2

Private IP: 10.0.102.183

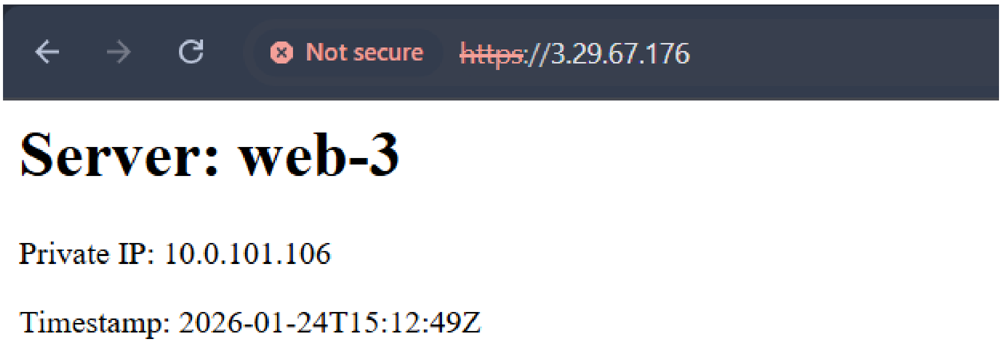
Timestamp: 2026-01-24T15:12:49Z

Nginx Service Stopped on Web Server 2



```
[ec2-user@ip-10-0-1-178 ~]$ ssh -i Project3Key.pem ec2-user@10.0.102.183
The authenticity of host '10.0.102.183 (10.0.102.183)' can't be established.
ECDSA key fingerprint is SHA256:u4s0ReHb4Kfs+W2/B4Q3ZVHCJ7TFkiKrvy9zvVSEOtc.
ECDSA key fingerprint is MD5:98:f8:53:03:fb:97:44:d4:67:66:48:f8:5a:99:e5:be.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.102.183' (ECDSA) to the list of known hosts.
Last login: Sat Jan 24 15:13:16 2026 from ip-10-0-1-178.me-central-1.compute.internal
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-102-183 ~]$ sudo systemctl stop nginx
[ec2-user@ip-10-0-102-183 ~]$ curl -k https://3.29.67.176
<html>
<body>
<h1>Server: web-3</h1>
<p>Private IP: 10.0.101.106</p>
<p>Timestamp: 2026-01-24T15:12:49Z</p>
</body>
</html>

[ec2-user@ip-10-0-102-183 ~]$ 
```

Show web 3

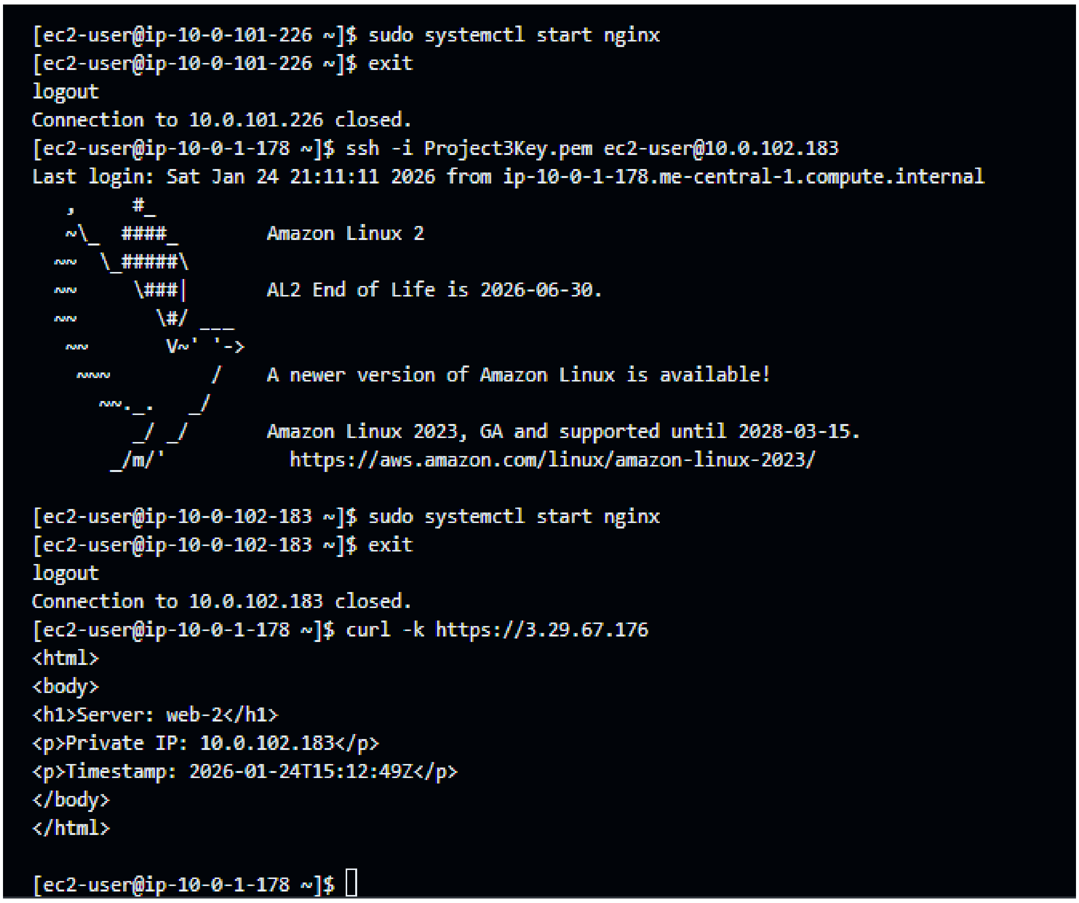


Server: web-3

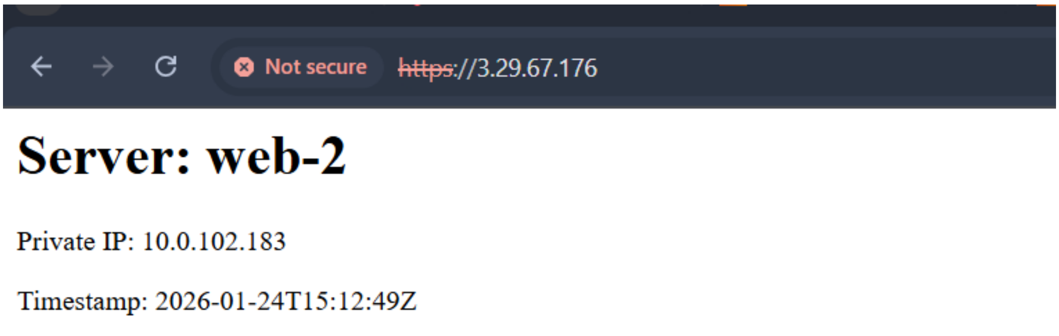
Private IP: 10.0.101.106

Timestamp: 2026-01-24T15:12:49Z

Again start nginx service



```
[ec2-user@ip-10-0-101-226 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-101-226 ~]$ exit
logout
Connection to 10.0.101.226 closed.
[ec2-user@ip-10-0-1-178 ~]$ ssh -i Project3Key.pem ec2-user@10.0.102.183
Last login: Sat Jan 24 21:11:11 2026 from ip-10-0-1-178.me-central-1.compute.internal
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-102-183 ~]$ sudo systemctl start nginx
[ec2-user@ip-10-0-102-183 ~]$ exit
logout
Connection to 10.0.102.183 closed.
[ec2-user@ip-10-0-1-178 ~]$ curl -k https://3.29.67.176
<html>
<body>
<h1>Server: web-2</h1>
<p>Private IP: 10.0.102.183</p>
<p>Timestamp: 2026-01-24T15:12:49Z</p>
</body>
</html>

[ec2-user@ip-10-0-1-178 ~]$
```



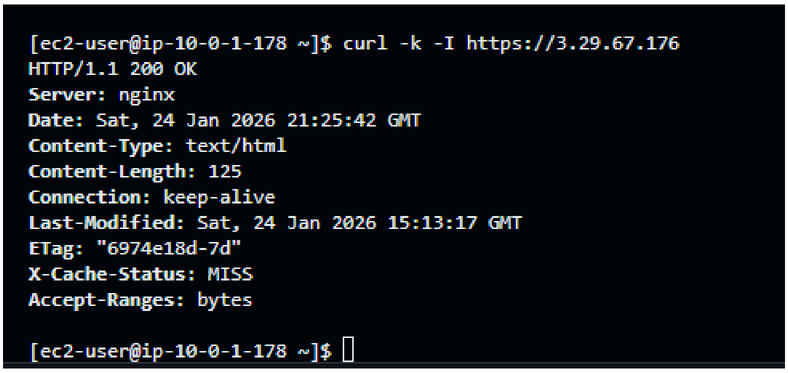
Server: web-2

Private IP: 10.0.102.183

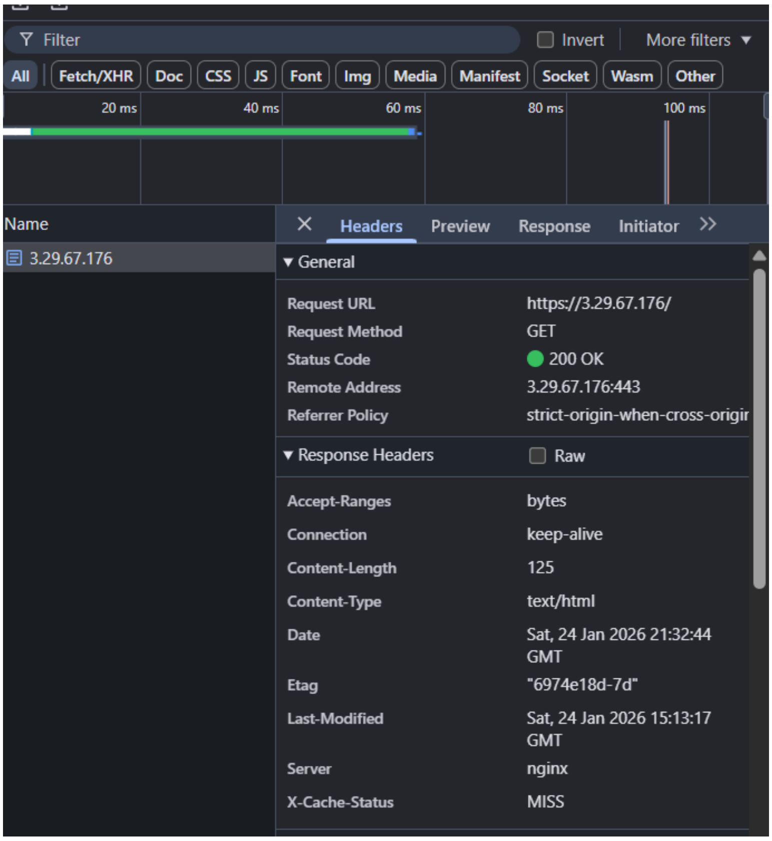
Timestamp: 2026-01-24T15:12:49Z

5.2 Caching Test

Initial Cache Response Showing X-Cache-Status: MISS



```
[ec2-user@ip-10-0-1-178 ~]$ curl -k -I https://3.29.67.176
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 24 Jan 2026 21:25:42 GMT
Content-Type: text/html
Content-Length: 125
Connection: keep-alive
Last-Modified: Sat, 24 Jan 2026 15:13:17 GMT
ETag: "6974e18d-7d"
X-Cache-Status: MISS
Accept-Ranges: bytes

[ec2-user@ip-10-0-1-178 ~]$ 
```



3.29.67.176

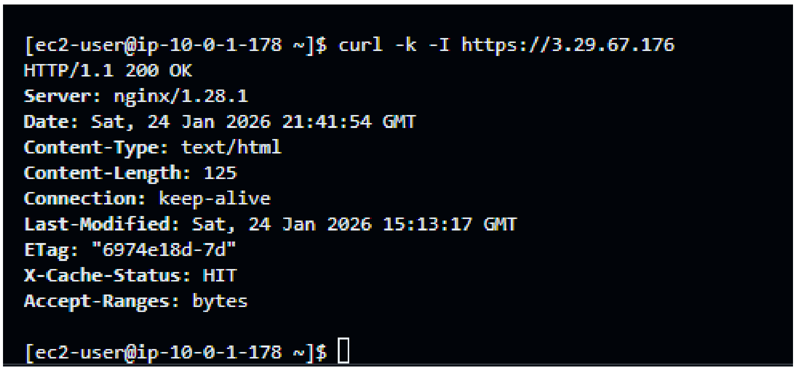
▾ General

Request URL	https://3.29.67.176/
Request Method	GET
Status Code	● 200 OK
Remote Address	3.29.67.176:443
Referrer Policy	strict-origin-when-cross-origir

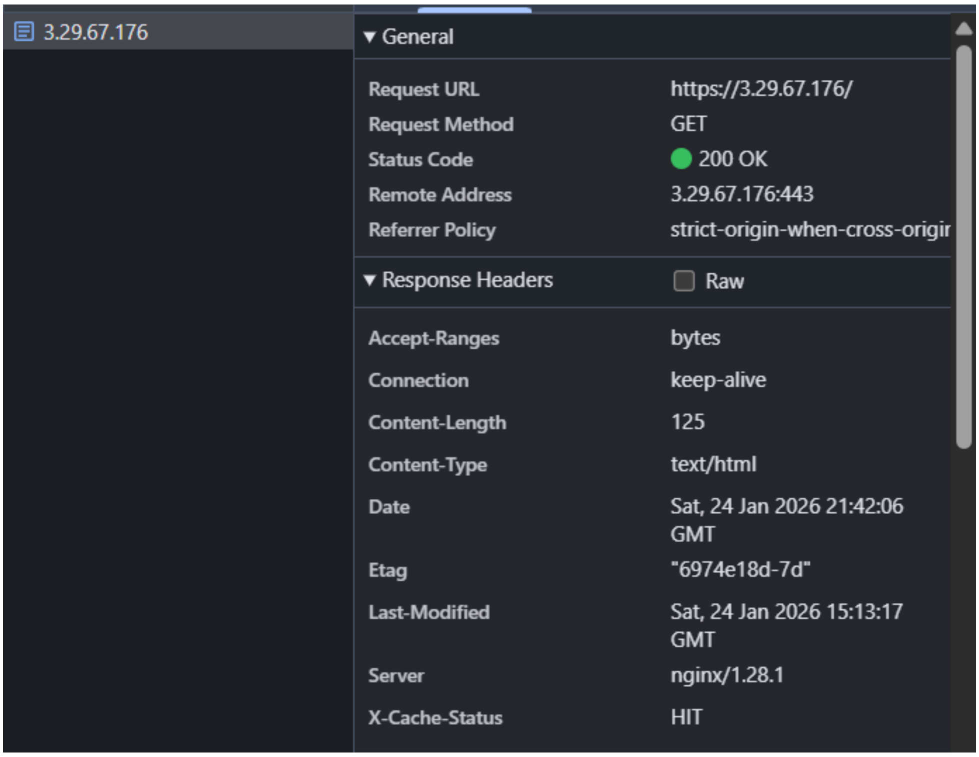
▾ Response Headers □ Raw

Accept-Ranges	bytes
Connection	keep-alive
Content-Length	125
Content-Type	text/html
Date	Sat, 24 Jan 2026 21:32:44 GMT
Etag	"6974e18d-7d"
Last-Modified	Sat, 24 Jan 2026 15:13:17 GMT
Server	nginx
X-Cache-Status	MISS

Subsequent Cache Response Showing X-Cache-Status: HIT



```
[ec2-user@ip-10-0-1-178 ~]$ curl -k -I https://3.29.67.176
HTTP/1.1 200 OK
Server: nginx/1.28.1
Date: Sat, 24 Jan 2026 21:41:54 GMT
Content-Type: text/html
Content-Length: 125
Connection: keep-alive
Last-Modified: Sat, 24 Jan 2026 15:13:17 GMT
ETag: "6974e18d-7d"
X-Cache-Status: HIT
Accept-Ranges: bytes

[ec2-user@ip-10-0-1-178 ~]$ 
```



3.29.67.176

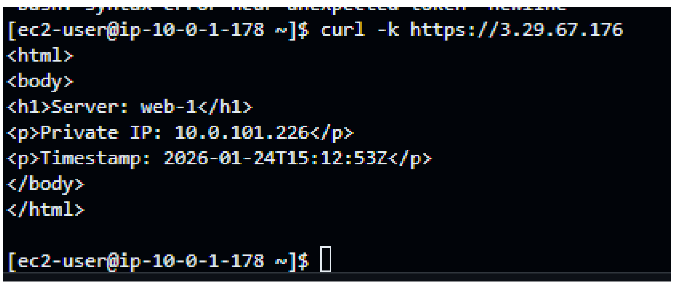
▼ General

Request URL	https://3.29.67.176/
Request Method	GET
Status Code	● 200 OK
Remote Address	3.29.67.176:443
Referrer Policy	strict-origin-when-cross-origir

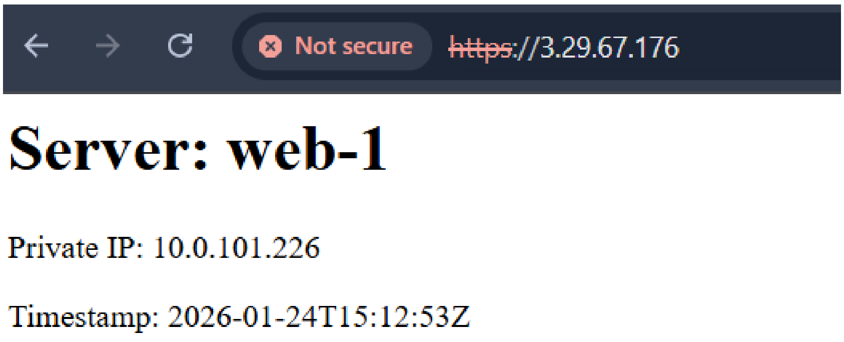
▼ Response Headers ☐ Raw

Accept-Ranges	bytes
Connection	keep-alive
Content-Length	125
Content-Type	text/html
Date	Sat, 24 Jan 2026 21:42:06 GMT
Etag	"6974e18d-7d"
Last-Modified	Sat, 24 Jan 2026 15:13:17 GMT
Server	nginx/1.28.1
X-Cache-Status	HIT

5.3 SSL Test



```
[ec2-user@ip-10-0-1-178 ~]$ curl -k https://3.29.67.176
<html>
<body>
<h1>Server: web-1</h1>
<p>Private IP: 10.0.101.226</p>
<p>Timestamp: 2026-01-24T15:12:53Z</p>
</body>
</html>

[ec2-user@ip-10-0-1-178 ~]$ 
```

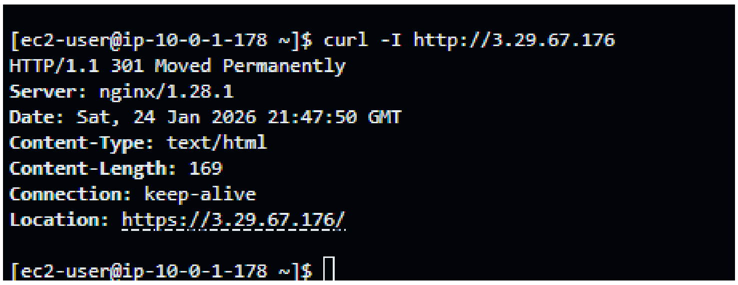


Server: web-1

Private IP: 10.0.101.226

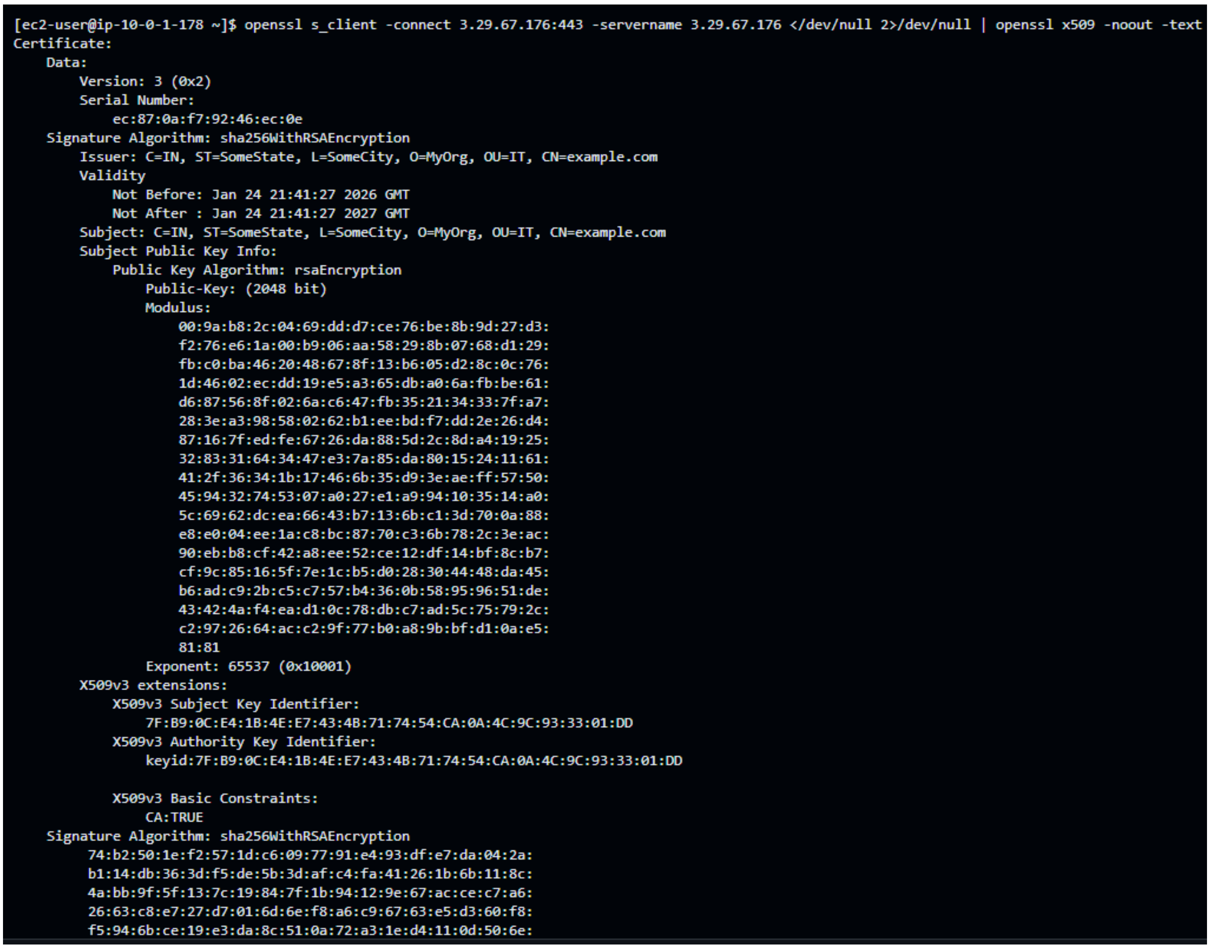
Timestamp: 2026-01-24T15:12:53Z

HTTP to HTTPS Redirection Verification



```
[ec2-user@ip-10-0-1-178 ~]$ curl -I http://3.29.67.176
HTTP/1.1 301 Moved Permanently
Server: nginx/1.28.1
Date: Sat, 24 Jan 2026 21:47:50 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://3.29.67.176/

[ec2-user@ip-10-0-1-178 ~]$ 
```

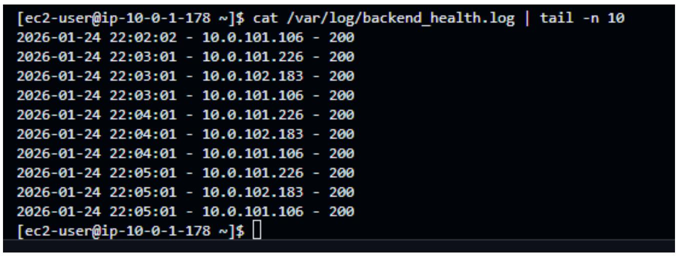
SSL Certificate Details (Self-Signed Certificate)



```
[ec2-user@ip-10-0-1-178 ~]$ openssl s_client -connect 3.29.67.176:443 -servername 3.29.67.176 </dev/null 2>/dev/null | openssl x509 -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            ec:87:0a:f7:92:46:ec:0e
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=IN, ST=SomeState, L=SomeCity, O=MyOrg, OU=IT, CN=example.com
        Validity
            Not Before: Jan 24 21:41:27 2026 GMT
            Not After : Jan 24 21:41:27 2027 GMT
        Subject: C=IN, ST=SomeState, L=SomeCity, O=MyOrg, OU=IT, CN=example.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:9a:b8:2c:04:69:dd:d7:ce:76:be:8b:9d:27:d3:
                    f2:76:e6:1a:00:b9:06:aa:58:29:8b:07:68:d1:29:
                    fb:c0:ba:46:20:48:67:8f:13:b6:05:d2:8c:0c:76:
                    1d:46:02:ec:dd:19:e5:a3:65:db:a0:6a:fb:be:61:
                    d6:87:56:8f:02:6a:c6:47:fb:35:21:34:33:7f:a7:
                    28:3e:a3:98:58:02:62:b1:ee:bd:f7:dd:2e:26:d4:
                    87:16:7f:ed:fe:67:26:da:88:5d:2c:8d:a4:19:25:
                    32:83:31:64:34:47:e3:7a:85:da:80:15:24:11:61:
                    41:2f:36:34:1b:17:46:6b:35:d9:3e:ae:ff:57:50:
                    45:94:32:74:53:07:a0:27:e1:a9:94:10:35:14:a0:
                    5c:69:62:dc:ea:66:43:b7:13:6b:c1:3d:70:0a:88:
                    e8:e0:04:ee:1a:c8:bc:87:70:c3:6b:78:2c:3e:ac:
                    90:eb:b8:cf:42:a8:ee:52:ce:12:df:14:bf:8c:b7:
                    cf:9c:85:16:5f:7e:1c:b5:d0:28:30:44:48:da:45:
                    b6:ad:c9:2b:c5:c7:57:b4:36:0b:58:95:96:51:de:
                    43:42:4a:f4:ea:d1:0c:78:db:c7:ad:5c:75:79:2c:
                    c2:97:26:64:ac:c2:9f:77:b0:a8:9b:bf:d1:0a:e5:
                    81:81
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                7F:B9:0C:E4:1B:4E:E7:43:4B:71:74:54:CA:0A:4C:9C:93:33:01:DD
            X509v3 Authority Key Identifier:
                keyid:7F:B9:0C:E4:1B:4E:E7:43:4B:71:74:54:CA:0A:4C:9C:93:33:01:DD

            X509v3 Basic Constraints:
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
         74:b2:50:1e:f2:57:1d:c6:09:77:91:e4:93:df:e7:da:04:2a:
         b1:14:db:36:3d:f5:de:5b:3d:af:c4:fa:41:26:1b:6b:11:8c:
         4a:bb:9f:5f:13:7c:19:84:7f:1b:94:12:9e:67:ac:ce:c7:a6:
         26:63:c8:e7:27:d7:01:6d:6e:f8:a6:c9:67:63:e5:d3:60:f8:
         f5:94:6b:ce:19:e3:da:8c:51:0a:72:a3:1e:d4:11:0d:50:6e:
```

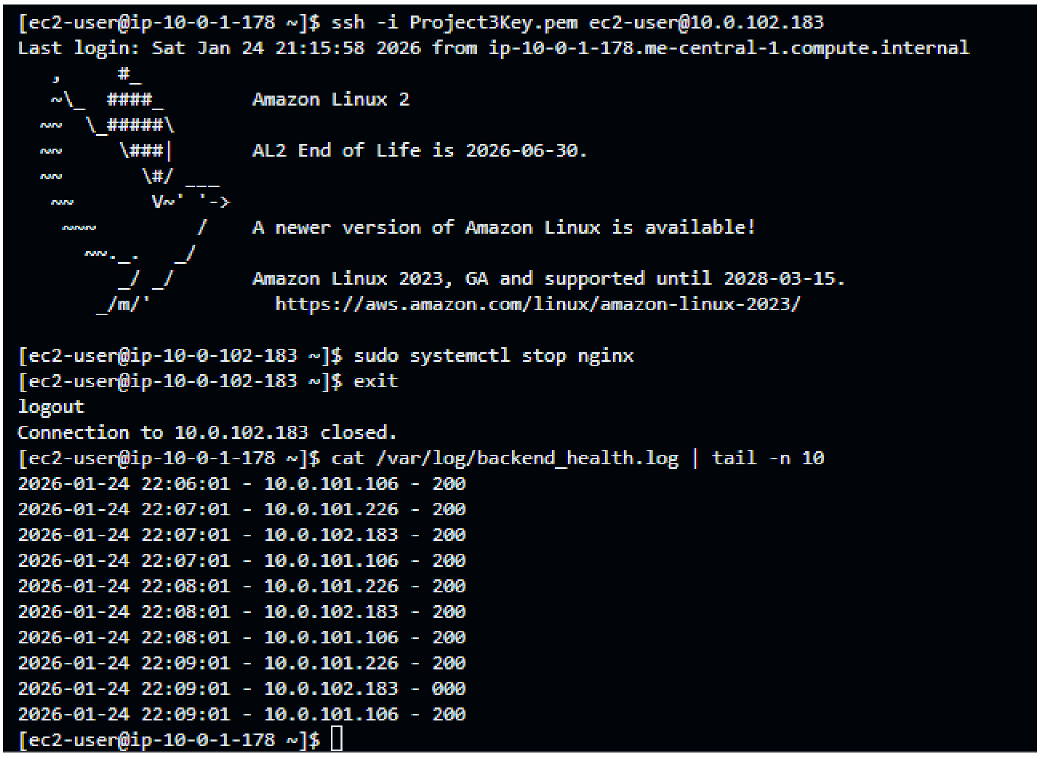
5.4 Monitoring Validation

Backend Health Monitoring Log File



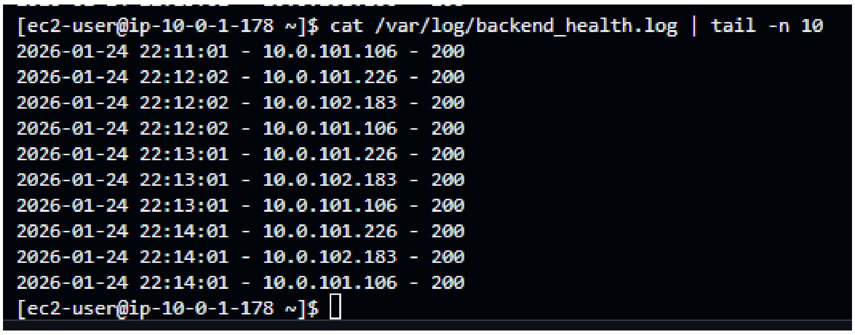
```
[ec2-user@ip-10-0-1-178 ~]$ cat /var/log/backend_health.log | tail -n 10
2026-01-24 22:02:02 - 10.0.101.106 - 200
2026-01-24 22:03:01 - 10.0.101.226 - 200
2026-01-24 22:03:01 - 10.0.102.183 - 200
2026-01-24 22:03:01 - 10.0.101.106 - 200
2026-01-24 22:04:01 - 10.0.101.226 - 200
2026-01-24 22:04:01 - 10.0.102.183 - 200
2026-01-24 22:04:01 - 10.0.101.106 - 200
2026-01-24 22:05:01 - 10.0.101.226 - 200
2026-01-24 22:05:01 - 10.0.102.183 - 200
2026-01-24 22:05:01 - 10.0.101.106 - 200
[ec2-user@ip-10-0-1-178 ~]$ 
```

Monitoring Log Showing Backend Failure



```
[ec2-user@ip-10-0-1-178 ~]$ ssh -i Project3Key.pem ec2-user@10.0.102.183
Last login: Sat Jan 24 21:15:58 2026 from ip-10-0-1-178.me-central-1.compute.internal
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-102-183 ~]$ sudo systemctl stop nginx
[ec2-user@ip-10-0-102-183 ~]$ exit
logout
Connection to 10.0.102.183 closed.
[ec2-user@ip-10-0-1-178 ~]$ cat /var/log/backend_health.log | tail -n 10
2026-01-24 22:06:01 - 10.0.101.106 - 200
2026-01-24 22:07:01 - 10.0.101.226 - 200
2026-01-24 22:07:01 - 10.0.102.183 - 200
2026-01-24 22:07:01 - 10.0.101.106 - 200
2026-01-24 22:08:01 - 10.0.101.226 - 200
2026-01-24 22:08:01 - 10.0.102.183 - 200
2026-01-24 22:08:01 - 10.0.101.106 - 200
2026-01-24 22:09:01 - 10.0.101.226 - 200
2026-01-24 22:09:01 - 10.0.102.183 - 000
2026-01-24 22:09:01 - 10.0.101.106 - 200
[ec2-user@ip-10-0-1-178 ~]$
```

Monitoring Log Showing Backend Recovery



```
[ec2-user@ip-10-0-1-178 ~]$ cat /var/log/backend_health.log | tail -n 10
2026-01-24 22:11:01 - 10.0.101.106 - 200
2026-01-24 22:12:02 - 10.0.101.226 - 200
2026-01-24 22:12:02 - 10.0.102.183 - 200
2026-01-24 22:12:02 - 10.0.101.106 - 200
2026-01-24 22:13:01 - 10.0.101.226 - 200
2026-01-24 22:13:01 - 10.0.102.183 - 200
2026-01-24 22:13:01 - 10.0.101.106 - 200
2026-01-24 22:14:01 - 10.0.101.226 - 200
2026-01-24 22:14:01 - 10.0.102.183 - 200
2026-01-24 22:14:01 - 10.0.101.106 - 200
[ec2-user@ip-10-0-1-178 ~]$
```

Challenges & Solutions

Challenge 1: Backend Servers Not Reachable from Load Balancer

During initial testing, the load balancer returned **502 Bad Gateway** errors when forwarding client requests to the backend servers. This indicated a communication issue between the load balancer and the backend tier.

Solution:
The issue was resolved by verifying the backend private IP addresses configured in the Nginx upstream block and ensuring that the backend security group allowed HTTP traffic **only from the load balancer security group**. Once the security group rules and upstream configuration were corrected, the load balancer successfully forwarded traffic to the backend servers.

Challenge 2: SSL Termination Configuration Issues

SSL termination initially failed due to incorrect certificate file paths and configuration errors in the Nginx configuration on the load balancer.

Solution:
Self-signed SSL certificates were regenerated on the load balancer using OpenSSL. Certificate and key paths were corrected, file permissions were adjusted, and the Nginx configuration was validated using the nginx -t command before restarting the service. After these fixes, HTTPS access and HTTP-to-HTTPS redirection worked correctly.

Challenge 3: Nginx Caching Always Showing MISS

During caching verification, the response header consistently showed X-Cache-Status: MISS, indicating that cached content was not being served.

Solution:
The cache directory permissions were corrected, and the proxy_cache_path and proxy_cache directives were reviewed in the Nginx configuration. After restarting the Nginx service on the load balancer, repeated requests returned X-Cache-Status: HIT, confirming that caching was functioning properly.

Challenge 4: Monitoring Script Not Producing Logs

The backend health monitoring script initially failed to generate log entries, making it difficult to verify backend availability.

Solution:
The script was made executable, the cron service was verified to be running, and the correct backend private IP addresses were passed to the script through the Ansible template. Once these issues were resolved, the monitoring script successfully logged backend health status at regular intervals.

Conclusion & Future Improvements

Conclusion

This project successfully demonstrates the implementation of a **highly available web application architecture** using AWS EC2 instances, Terraform, Ansible, and Nginx. The infrastructure supports load balancing, failover to a backup server, SSL termination, caching, and automated monitoring. All required tests—including failover, caching, SSL redirection, and monitoring validation—were performed successfully, confirming the reliability and robustness of the system.

The project strengthened practical understanding of **Infrastructure as Code (IaC)**, **configuration management**, and **high availability principles** in real-world cloud environments.

Future Improvements

- Implement **Auto Scaling Groups** to dynamically adjust backend capacity.
- Integrate **CloudWatch or external monitoring tools** for alert-based monitoring.
- Use **managed SSL certificates** from AWS Certificate Manager.
- Introduce **centralized logging** for better observability.
- Add a **CI/CD pipeline** for automated application updates.