Fatima Jinnah Women University

Opening Portals of Excellence Through Higher Education

CLOUD COMPUTING LAB

BSE (V-B)

Submitted By:

Musfira Farooq

Roll No:

2023-BSE-045

Submitted To:

Sir Muhammad Shoaib

LAB 15(PROJECT)

Terraform + Ansible Roles: Nginx Frontend with 3 Backend HTTPD Servers (HA + Auto-Config)

TABLE OF CONTENTS

Introduction

This lab project demonstrates the deployment of a high-availability web architecture on AWS using Terraform for infrastructure provisioning and Ansible roles for configuration management.

The system consists of:

- 1 Frontend EC2 instance running Nginx
- 3 Backend EC2 instances running Apache HTTPD
- 2 active backends + 1 backup backend

Architecture Overview

Component	Description
VPC	Custom VPC with public subnet
Frontend EC2	Nginx reverse proxy and load balancer
Backend EC2s	Apache HTTPD web servers
Security Group	SSH (22) and HTTP (80) access
Automation	Terraform + Ansible integration

Infrastructure Design (Terraform)

Key Terraform Features:

- Custom VPC, Subnet, Internet Gateway
- Public subnet with Internet access
- Security group allowing:
- SSH from developer IP
- HTTP from anywhere

EC2 instances:

- 1 frontend
- 3 backends
- Use of variables, locals, and outputs

Configuration Management (Ansible)

Role	Purpose
backend	Install Apache HTTPD and deploy unique backend page
frontend	Install and configure Nginx load balancer
common (optional)	Base configuration (if used)

Backend Role Behavior

- Installs Apache HTTPD
- Enables and starts HTTPD service

Deploys a dynamic HTML page showing:

- Backend identity
- Private IP address
- Each backend displays distinct content.

Frontend Role Behavior

- Installs Nginx
- Configures Nginx as a reverse proxy

Uses an upstream configuration with:

- 2 primary backend servers
- 1 backup backend server
- Automatically restarts Nginx when configuration changes

Nginx Load Balancer Behavior

Normal Operation

Requests to frontend IP are distributed between:

- Backend 1
- Backend 2

When Backend 1 and Backend 2 are stopped:

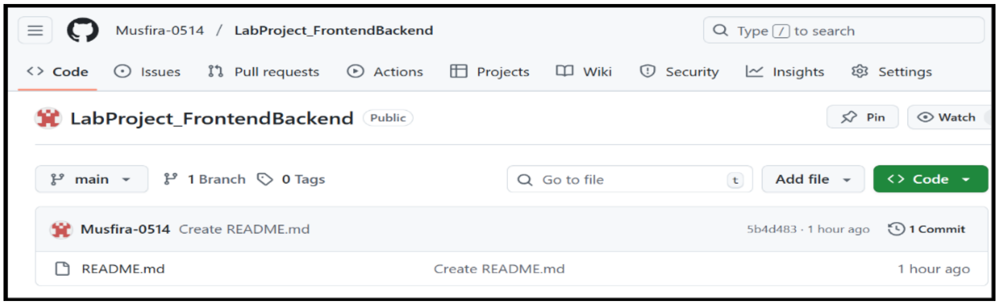
- Requests are served by Backend 3 (backup)
- This confirms high availability and fault tolerance.

Automation with Terraform and Ansible

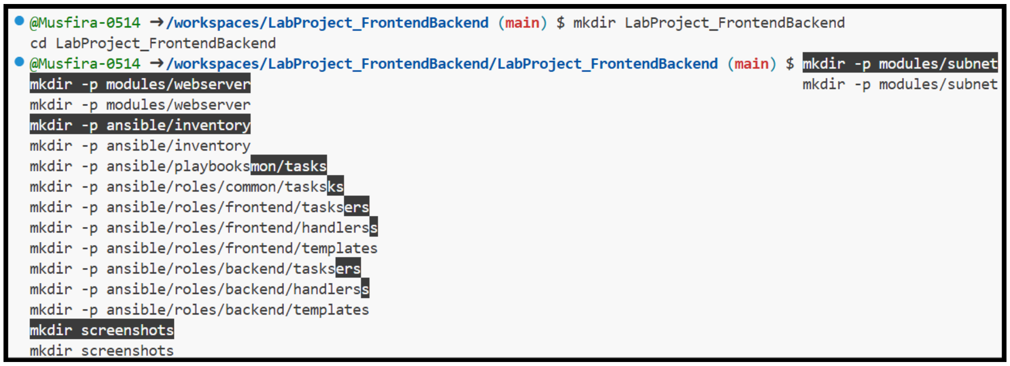
- Terraform provisions all EC2 instances
- Terraform triggers Ansible automatically using null_resource
- No manual ansible-playbook command is required
- Re-running terraform apply is idempotent

Practical outputs

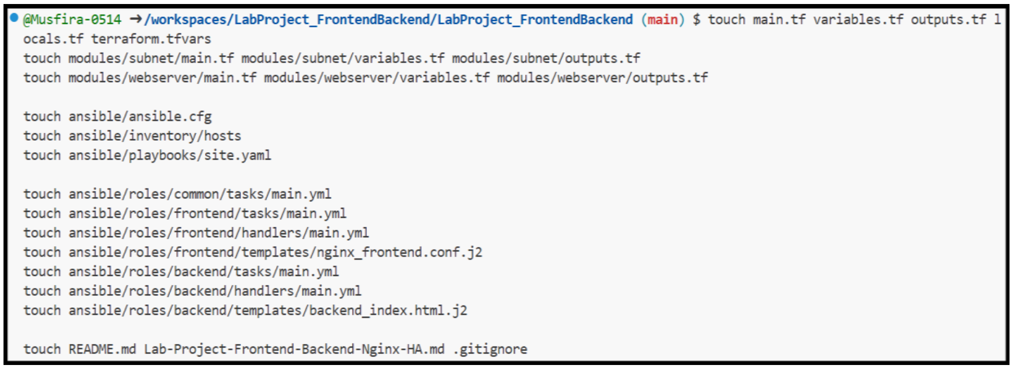
Creating repository:



Codespace loaded and creating files and folders:

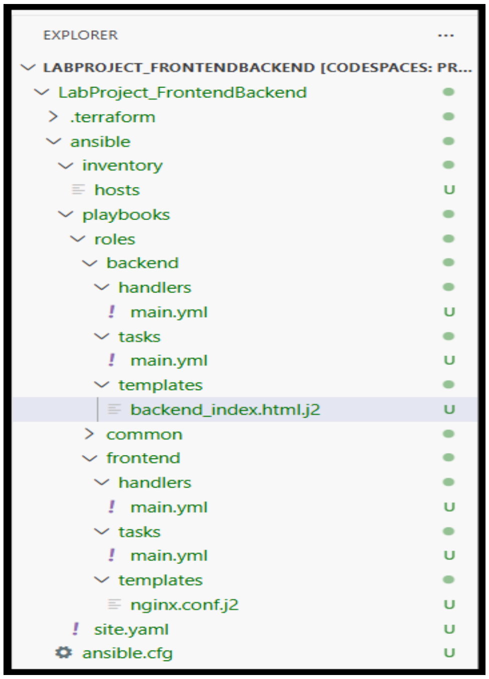


```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend (main) $ mkdir LabProject_FrontendBackend
cd LabProject_FrontendBackend
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ mkdir -p modules/subnet
mkdir -p modules/webserver                                                                 mkdir -p modules/subnet
mkdir -p modules/webserver
mkdir -p ansible/inventory
mkdir -p ansible/inventory
mkdir -p ansible/playbooksmon/tasks
mkdir -p ansible/roles/common/tasksks
mkdir -p ansible/roles/frontend/taskers
mkdir -p ansible/roles/frontend/handlerss
mkdir -p ansible/roles/frontend/templates
mkdir -p ansible/roles/backend/taskers
mkdir -p ansible/roles/backend/handlerss
mkdir -p ansible/roles/backend/templates
mkdir screenshots
mkdir screenshots
```

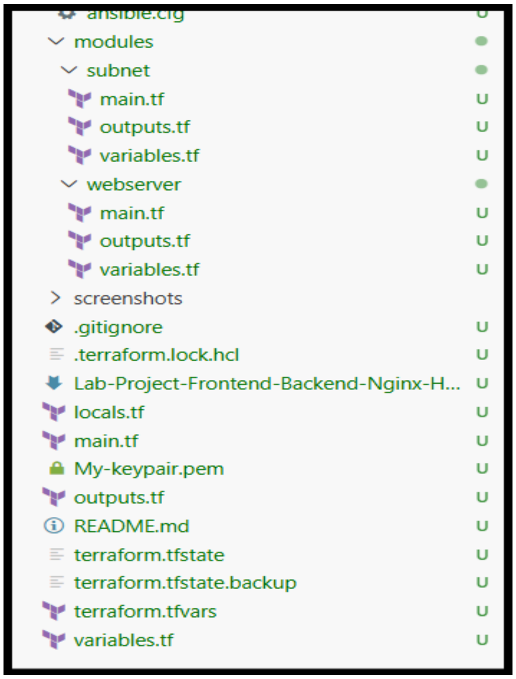


```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ touch main.tf variables.tf outputs.tf l
ocals.tf terraform.tfvars
touch modules/subnet/main.tf modules/subnet/variables.tf modules/subnet/outputs.tf
touch modules/webserver/main.tf modules/webserver/variables.tf modules/webserver/outputs.tf

touch ansible/ansible.cfg
touch ansible/inventory/hosts
touch ansible/playbooks/site.yaml

touch ansible/roles/common/tasks/main.yml
touch ansible/roles/frontend/tasks/main.yml
touch ansible/roles/frontend/handlers/main.yml
touch ansible/roles/frontend/templates/nginx_frontend.conf.j2
touch ansible/roles/backend/tasks/main.yml
touch ansible/roles/backend/handlers/main.yml
touch ansible/roles/backend/templates/backend_index.html.j2

touch README.md Lab-Project-Frontend-Backend-Nginx-HA.md .gitignore
```

Project Structure

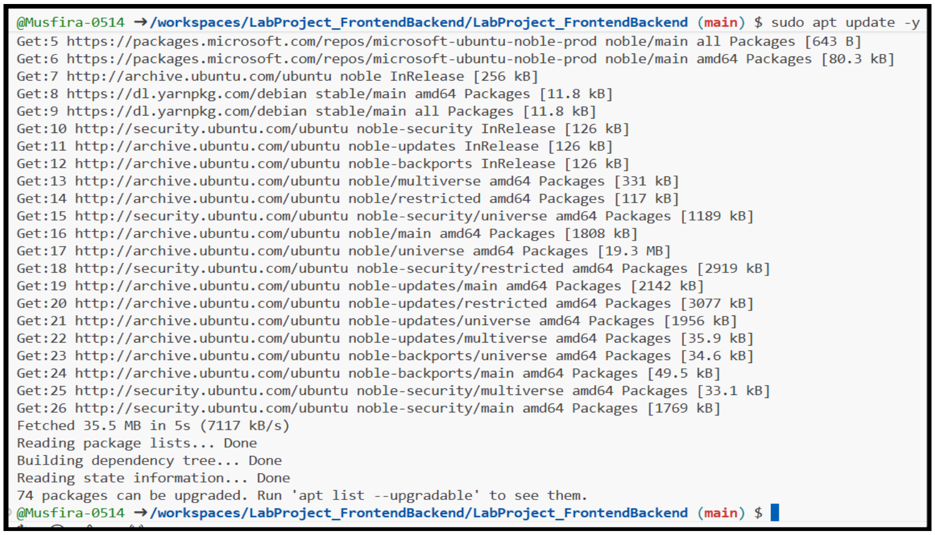


```
EXPLORER                                          ...
∨ LABPROJECT_FRONTENDBACKEND [CODESPACES: PR...
  ∨ LabProject_FrontendBackend                    ●
    > .terraform                                  ●
    ∨ ansible                                     ●
      ∨ inventory                                 ●
          hosts                                   U
      ∨ playbooks                                 ●
        ∨ roles                                   ●
          ∨ backend                               ●
            ∨ handlers                            ●
              ! main.yml                          U
            ∨ tasks                               ●
              ! main.yml                          U
            ∨ templates                           ●
                backend_index.html.j2             U
          > common                                ●
          ∨ frontend                              ●
            ∨ handlers                            ●
              ! main.yml                          U
            ∨ tasks                               ●
              ! main.yml                          U
            ∨ templates                           ●
                nginx.conf.j2                     U
        ! site.yaml                               U
      ansible.cfg                                 U
```

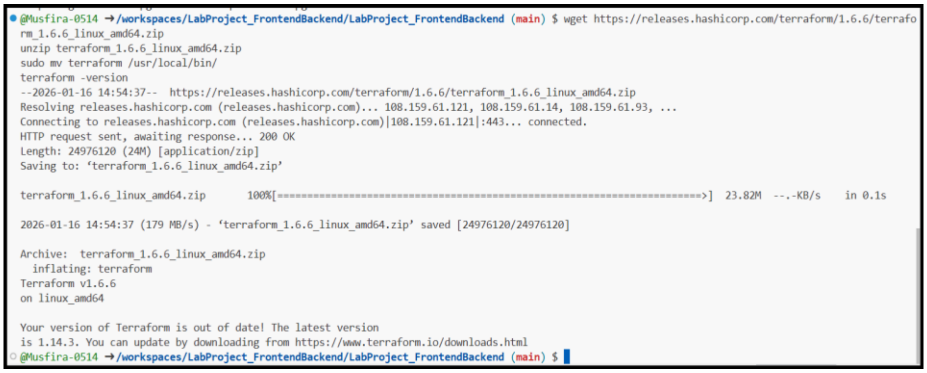


```
      ansible.cfg                                 U
    ∨ modules                                     ●
      ∨ subnet                                    ●
          main.tf                                 U
          outputs.tf                              U
          variables.tf                            U
      ∨ webserver                                 ●
          main.tf                                 U
          outputs.tf                              U
          variables.tf                            U
    > screenshots
      .gitignore                                  U
      .terraform.lock.hcl                         U
      Lab-Project-Frontend-Backend-Nginx-H...     U
      locals.tf                                   U
      main.tf                                     U
      My-keypair.pem                              U
      outputs.tf                                  U
      README.md                                   U
      terraform.tfstate                           U
      terraform.tfstate.backup                    U
      terraform.tfvars                            U
      variables.tf                                U
```

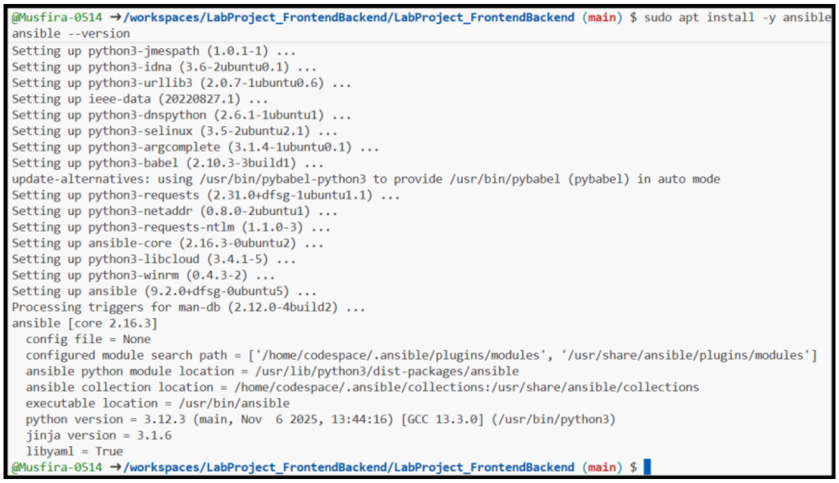
Installing required terraform,aws and ansible:



```
@Musfira-0514 ➜/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ sudo apt update -y
Get:5 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main all Packages [643 B]
Get:6 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main amd64 Packages [80.3 kB]
Get:7 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:8 https://dl.yarnpkg.com/debian stable/main amd64 Packages [11.8 kB]
Get:9 https://dl.yarnpkg.com/debian stable/main all Packages [11.8 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1189 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2919 kB]
Get:19 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [2142 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [3077 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1956 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [35.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [34.6 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [49.5 kB]
Get:25 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [33.1 kB]
Get:26 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1769 kB]
Fetched 35.5 MB in 5s (7117 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
74 packages can be upgraded. Run 'apt list --upgradable' to see them.
@Musfira-0514 ➜/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

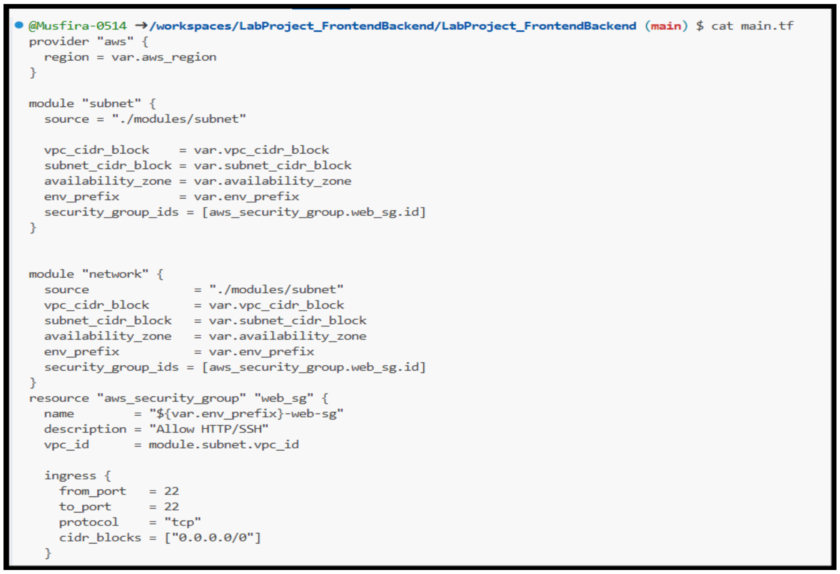


```
● @Musfira-0514 ➜/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ wget https://releases.hashicorp.com/terraform/1.6.6/terrafo
  rm_1.6.6_linux_amd64.zip
  unzip terraform_1.6.6_linux_amd64.zip
  sudo mv terraform /usr/local/bin/
  terraform -version
  --2026-01-16 14:54:37--  https://releases.hashicorp.com/terraform/1.6.6/terraform_1.6.6_linux_amd64.zip
  Resolving releases.hashicorp.com (releases.hashicorp.com)... 108.159.61.121, 108.159.61.14, 108.159.61.93, ...
  Connecting to releases.hashicorp.com (releases.hashicorp.com)|108.159.61.121|:443... connected.
  HTTP request sent, awaiting response... 200 OK
  Length: 24976120 (24M) [application/zip]
  Saving to: ‘terraform_1.6.6_linux_amd64.zip’

  terraform_1.6.6_linux_amd64.zip      100%[======================================================================>]  23.82M  --.-KB/s    in 0.1s

  2026-01-16 14:54:37 (179 MB/s) - ‘terraform_1.6.6_linux_amd64.zip’ saved [24976120/24976120]

  Archive:  terraform_1.6.6_linux_amd64.zip
    inflating: terraform
  Terraform v1.6.6
  on linux_amd64

  Your version of Terraform is out of date! The latest version
  is 1.14.3. You can update by downloading from https://www.terraform.io/downloads.html
○ @Musfira-0514 ➜/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

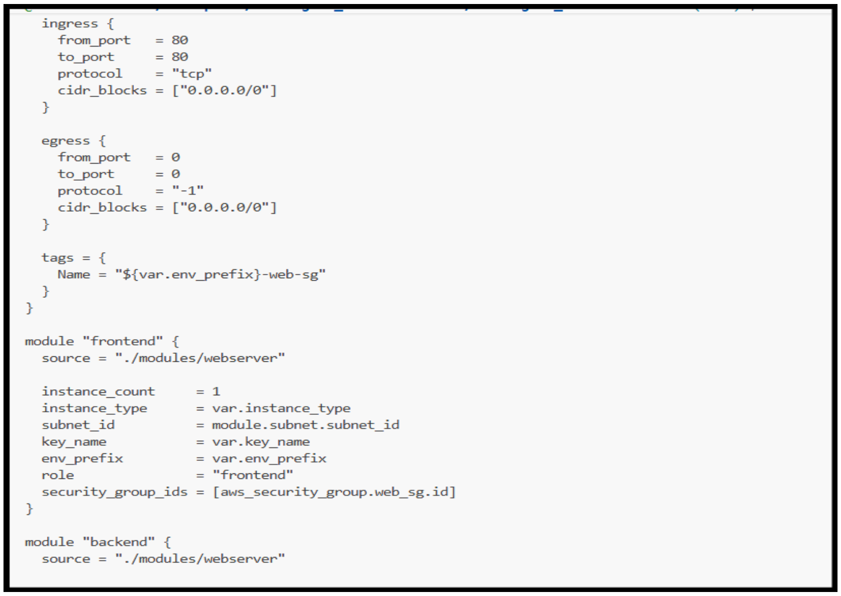


```
@Musfira-0514 ➜/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ sudo apt install -y ansible
ansible --version
Setting up python3-jmespath (1.0.1-1) ...
Setting up python3-idna (3.6-2ubuntu0.1) ...
Setting up python3-urllib3 (2.0.7-1ubuntu0.6) ...
Setting up ieee-data (20220827.1) ...
Setting up python3-dnspython (2.6.1-1ubuntu1) ...
Setting up python3-selinux (3.5-2ubuntu2.1) ...
Setting up python3-argcomplete (3.1.4-1ubuntu0.1) ...
Setting up python3-babel (2.10.3-3build1) ...
update-alternatives: using /usr/bin/pybabel-python3 to provide /usr/bin/pybabel (pybabel) in auto mode
Setting up python3-requests (2.31.0+dfsg-1ubuntu1.1) ...
Setting up python3-netaddr (0.8.0-2ubuntu1) ...
Setting up python3-requests-ntlm (1.1.0-3) ...
Setting up ansible-core (2.16.3-0ubuntu2) ...
Setting up python3-libcloud (3.4.1-5) ...
Setting up python3-winrm (0.4.3-2) ...
Setting up ansible (9.2.0+dfsg-0ubuntu5) ...
Processing triggers for man-db (2.12.0-4build2) ...
ansible [core 2.16.3]
  config file = None
  configured module search path = ['/home/codespace/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/codespace/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Nov  6 2025, 13:44:16) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.6
  libyaml = True
@Musfira-0514 ➜/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

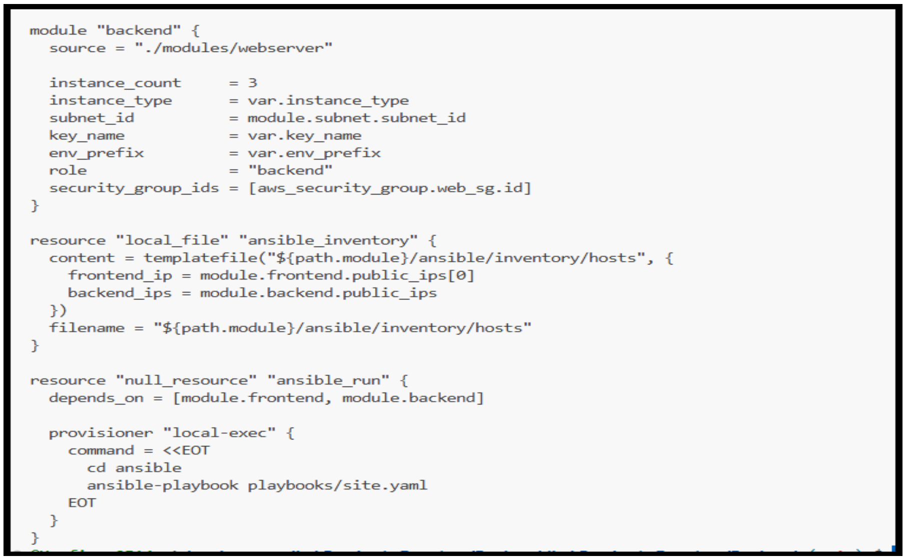
Root main.tf:



```
● @Musfira-0514 ➜/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat main.tf
provider "aws" {
  region = var.aws_region
}

module "subnet" {
  source = "./modules/subnet"

  vpc_cidr_block    = var.vpc_cidr_block
  subnet_cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  env_prefix        = var.env_prefix
  security_group_ids = [aws_security_group.web_sg.id]
}


module "network" {
  source              = "./modules/subnet"
  vpc_cidr_block      = var.vpc_cidr_block
  subnet_cidr_block   = var.subnet_cidr_block
  availability_zone   = var.availability_zone
  env_prefix          = var.env_prefix
  security_group_ids = [aws_security_group.web_sg.id]
}
resource "aws_security_group" "web_sg" {
  name        = "${var.env_prefix}-web-sg"
  description = "Allow HTTP/SSH"
  vpc_id      = module.subnet.vpc_id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

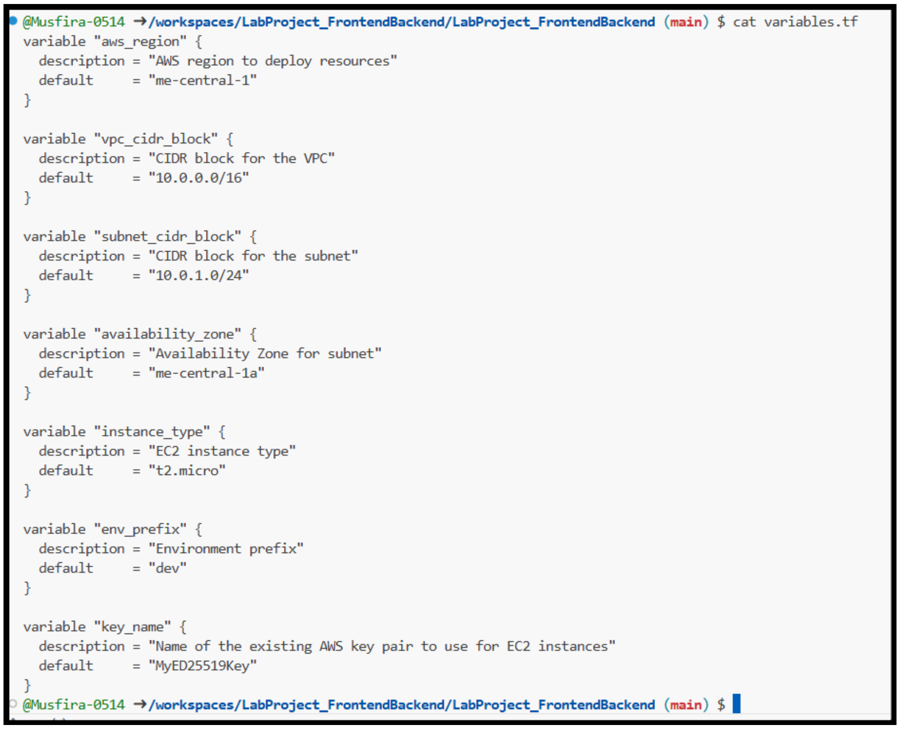


```
  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

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

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

module "frontend" {
  source = "./modules/webserver"

  instance_count     = 1
  instance_type      = var.instance_type
  subnet_id          = module.subnet.subnet_id
  key_name           = var.key_name
  env_prefix         = var.env_prefix
  role               = "frontend"
  security_group_ids = [aws_security_group.web_sg.id]
}

module "backend" {
  source = "./modules/webserver"
```



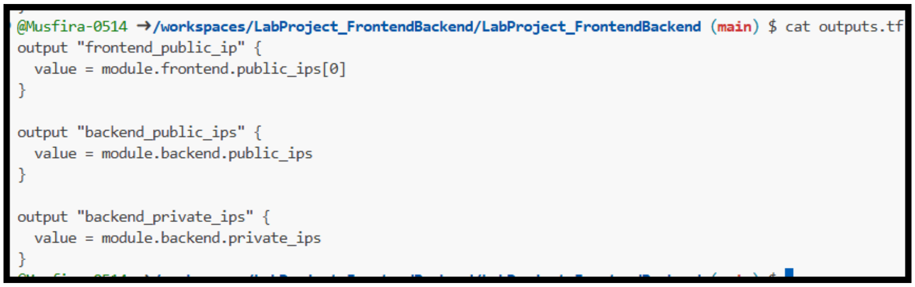
```
module "backend" {
  source = "./modules/webserver"

  instance_count     = 3
  instance_type      = var.instance_type
  subnet_id          = module.subnet.subnet_id
  key_name           = var.key_name
  env_prefix         = var.env_prefix
  role               = "backend"
  security_group_ids = [aws_security_group.web_sg.id]
}

resource "local_file" "ansible_inventory" {
  content = templatefile("${path.module}/ansible/inventory/hosts", {
    frontend_ip = module.frontend.public_ips[0]
    backend_ips = module.backend.public_ips
  })
  filename = "${path.module}/ansible/inventory/hosts"
}

resource "null_resource" "ansible_run" {
  depends_on = [module.frontend, module.backend]

  provisioner "local-exec" {
    command = <<EOT
      cd ansible
      ansible-playbook playbooks/site.yaml
    EOT
  }
}
```

Root variables.tf:



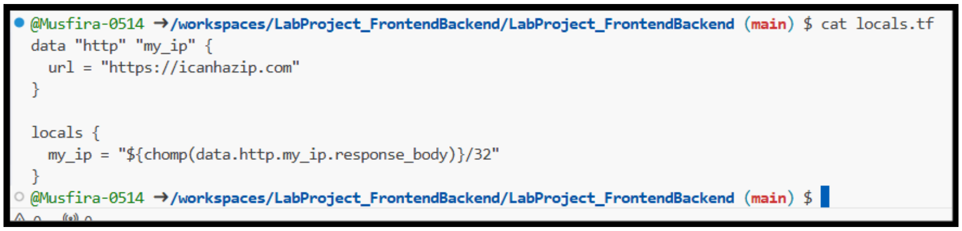
```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat variables.tf
variable "aws_region" {
  description = "AWS region to deploy resources"
  default     = "me-central-1"
}

variable "vpc_cidr_block" {
  description = "CIDR block for the VPC"
  default     = "10.0.0.0/16"
}

variable "subnet_cidr_block" {
  description = "CIDR block for the subnet"
  default     = "10.0.1.0/24"
}

variable "availability_zone" {
  description = "Availability Zone for subnet"
  default     = "me-central-1a"
}

variable "instance_type" {
  description = "EC2 instance type"
  default     = "t2.micro"
}

variable "env_prefix" {
  description = "Environment prefix"
  default     = "dev"
}

variable "key_name" {
  description = "Name of the existing AWS key pair to use for EC2 instances"
  default     = "MyED25519Key"
}
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Root outputs.tf:



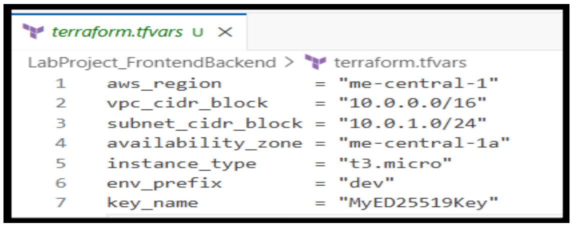
```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat outputs.tf
output "frontend_public_ip" {
  value = module.frontend.public_ips[0]
}

output "backend_public_ips" {
  value = module.backend.public_ips
}

output "backend_private_ips" {
  value = module.backend.private_ips
}
```

Root locals.tf:



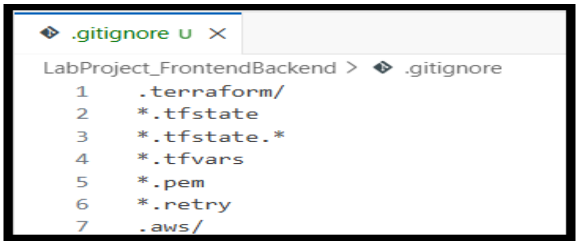
```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat locals.tf
data "http" "my_ip" {
  url = "https://icanhazip.com"
}

locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Terraform.tfvars:



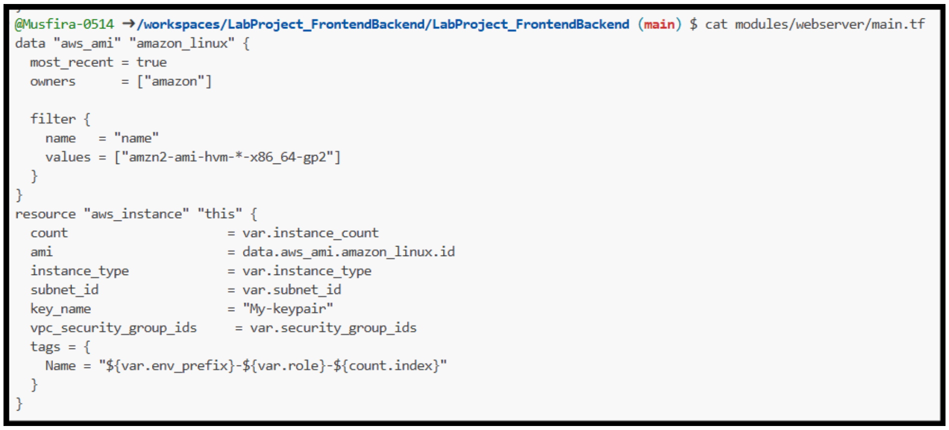
```
aws_region        = "me-central-1"
vpc_cidr_block    = "10.0.0.0/16"
subnet_cidr_block = "10.0.1.0/24"
availability_zone = "me-central-1a"
instance_type     = "t3.micro"
env_prefix        = "dev"
key_name          = "MyED25519Key"
```

.Gitignore



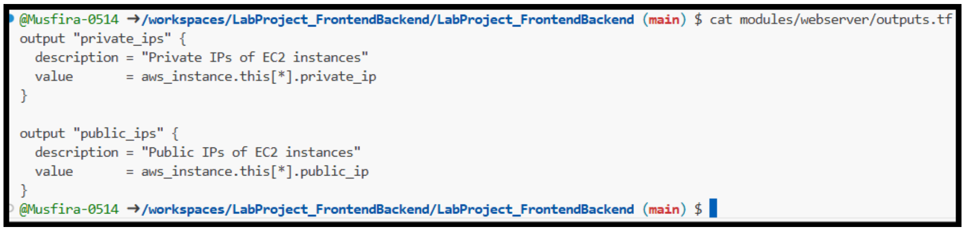
```
.terraform/
*.tfstate
*.tfstate.*
*.tfvars
*.pem
*.retry
.aws/
```

Webserver/main.tf:



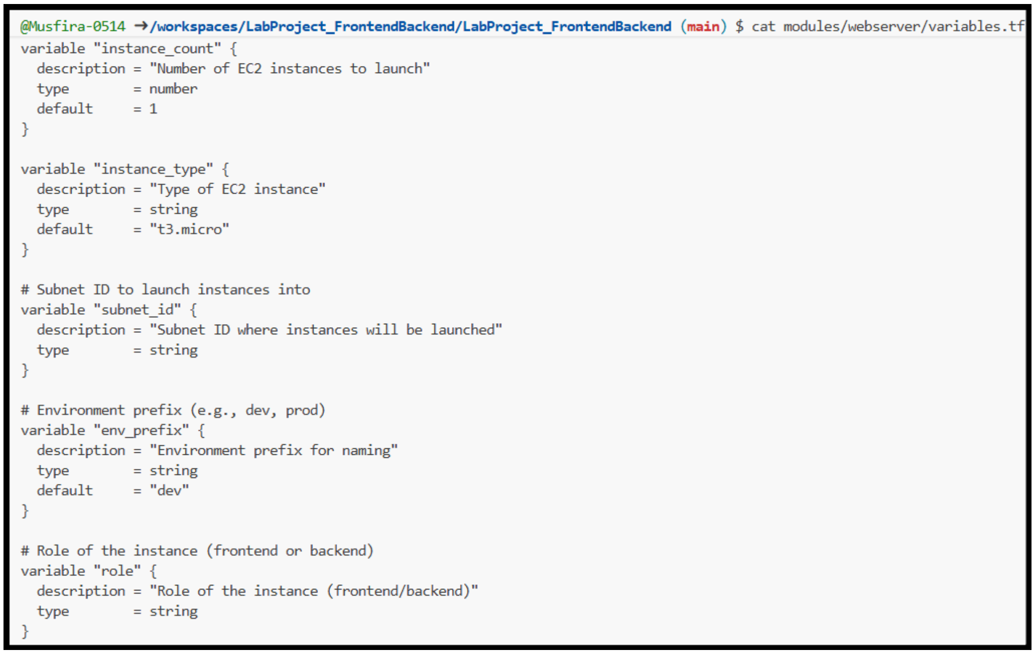
```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat modules/webserver/main.tf
data "aws_ami" "amazon_linux" {
  most_recent = true
  owners      = ["amazon"]

  filter {
    name   = "name"
    values = ["amzn2-ami-hvm-*-x86_64-gp2"]
  }
}
resource "aws_instance" "this" {
  count                     = var.instance_count
  ami                       = data.aws_ami.amazon_linux.id
  instance_type             = var.instance_type
  subnet_id                 = var.subnet_id
  key_name                  = "My-keypair"
  vpc_security_group_ids     = var.security_group_ids
  tags = {
    Name = "${var.env_prefix}-${var.role}-${count.index}"
  }
}
```

outputs.tf:

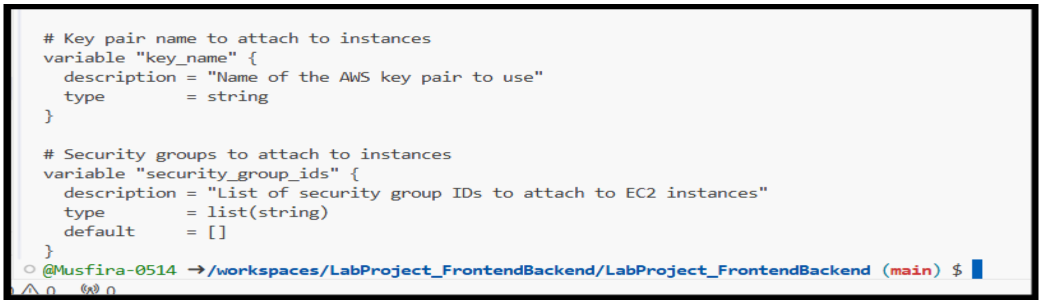


```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat modules/webserver/outputs.tf
output "private_ips" {
  description = "Private IPs of EC2 instances"
  value       = aws_instance.this[*].private_ip
}

output "public_ips" {
  description = "Public IPs of EC2 instances"
  value       = aws_instance.this[*].public_ip
}
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Variables.tf:

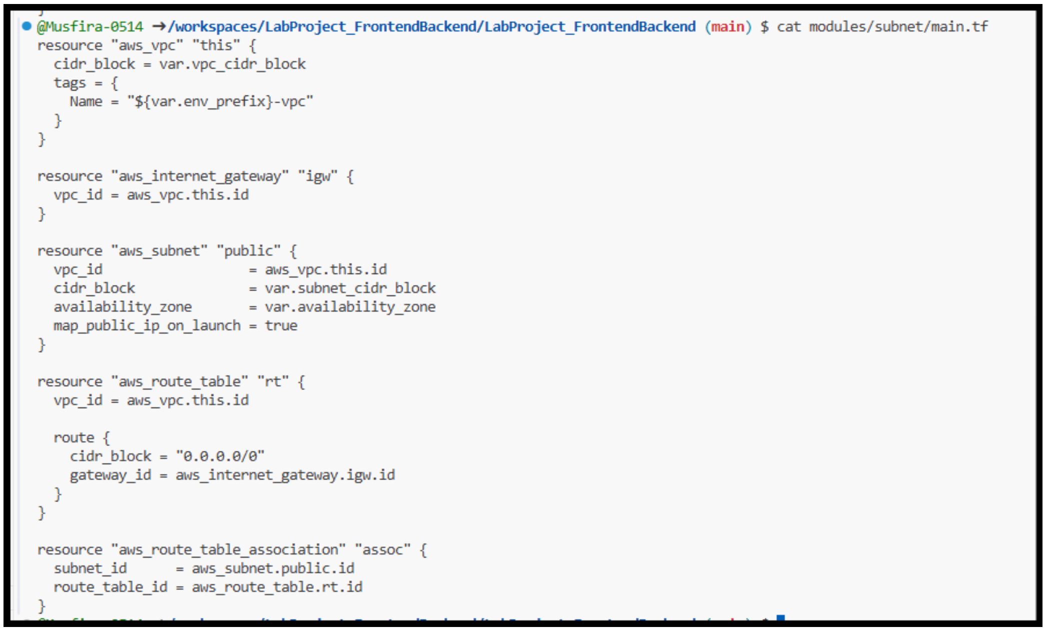


```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat modules/webserver/variables.tf
variable "instance_count" {
  description = "Number of EC2 instances to launch"
  type        = number
  default     = 1
}

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

# Subnet ID to launch instances into
variable "subnet_id" {
  description = "Subnet ID where instances will be launched"
  type        = string
}

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

# Role of the instance (frontend or backend)
variable "role" {
  description = "Role of the instance (frontend/backend)"
  type        = string
}
```



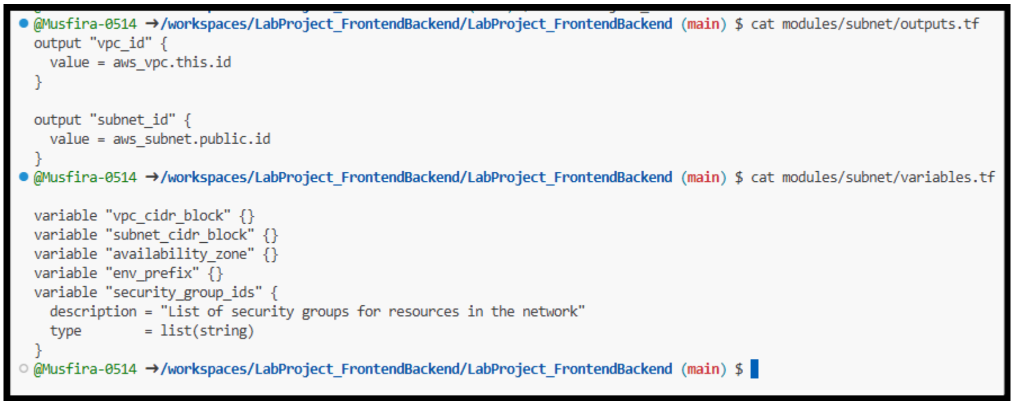
```
# Key pair name to attach to instances
variable "key_name" {
  description = "Name of the AWS key pair to use"
  type        = string
}

# Security groups to attach to instances
variable "security_group_ids" {
  description = "List of security group IDs to attach to EC2 instances"
  type        = list(string)
  default     = []
}
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Subnet/main.tf:



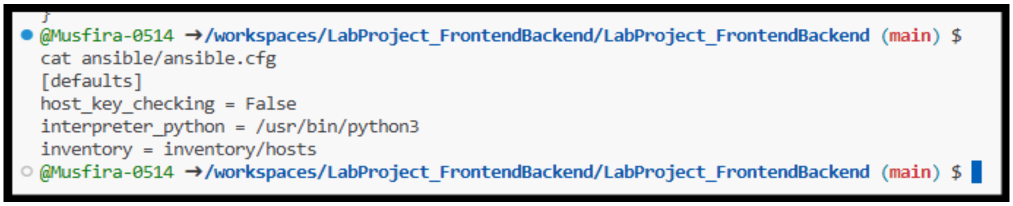
```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat modules/subnet/main.tf
resource "aws_vpc" "this" {
  cidr_block = var.vpc_cidr_block
  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}

resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.this.id
}

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

resource "aws_route_table" "rt" {
  vpc_id = aws_vpc.this.id

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

resource "aws_route_table_association" "assoc" {
  subnet_id      = aws_subnet.public.id
  route_table_id = aws_route_table.rt.id
}
```

Outputs.tf & Variables.tf:



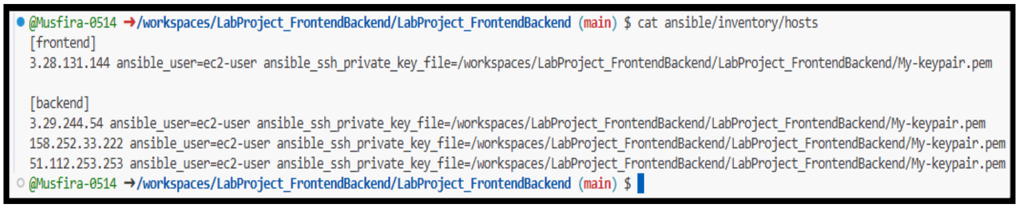
```
● @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat modules/subnet/outputs.tf
  output "vpc_id" {
    value = aws_vpc.this.id
  }

  output "subnet_id" {
    value = aws_subnet.public.id
  }
● @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat modules/subnet/variables.tf

  variable "vpc_cidr_block" {}
  variable "subnet_cidr_block" {}
  variable "availability_zone" {}
  variable "env_prefix" {}
  variable "security_group_ids" {
    description = "List of security groups for resources in the network"
    type        = list(string)
  }
○ @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Ansible.config:



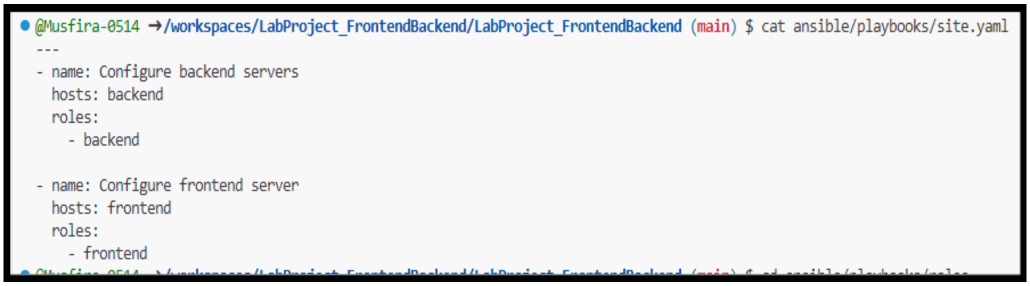
```
● @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
  cat ansible/ansible.cfg
  [defaults]
  host_key_checking = False
  interpreter_python = /usr/bin/python3
  inventory = inventory/hosts
○ @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Inventory/hosts:



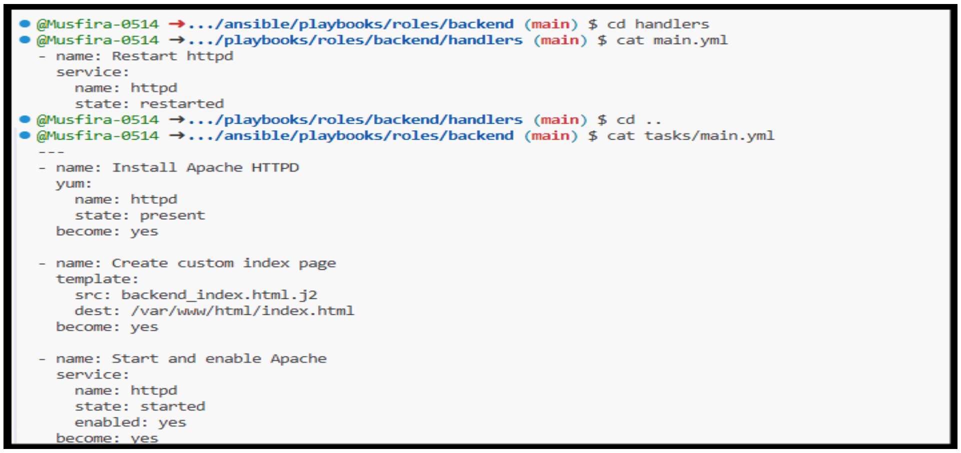
```
● @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat ansible/inventory/hosts
  [frontend]
  3.28.131.144 ansible_user=ec2-user ansible_ssh_private_key_file=/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend/My-keypair.pem

  [backend]
  3.29.244.54 ansible_user=ec2-user ansible_ssh_private_key_file=/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend/My-keypair.pem
  158.252.33.222 ansible_user=ec2-user ansible_ssh_private_key_file=/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend/My-keypair.pem
  51.112.253.253 ansible_user=ec2-user ansible_ssh_private_key_file=/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend/My-keypair.pem
○ @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Playbook/site.yml:

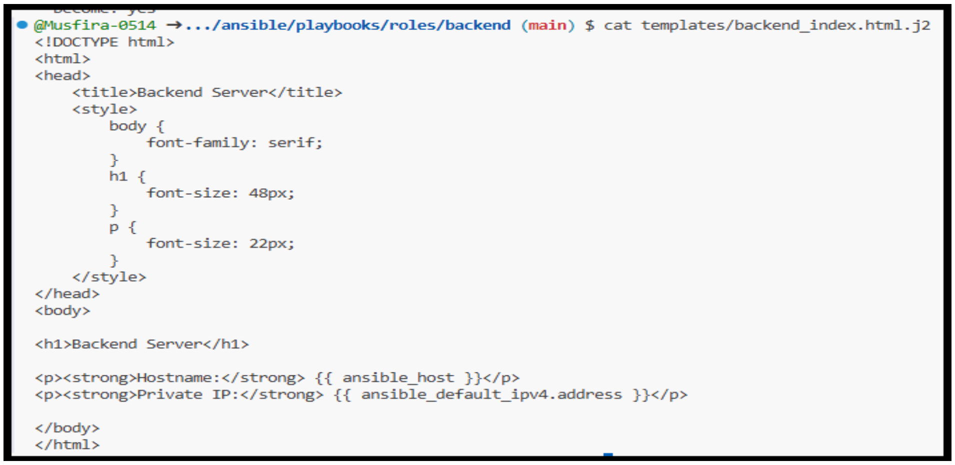


```
● @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cat ansible/playbooks/site.yaml
  ---
  - name: Configure backend servers
    hosts: backend
    roles:
      - backend

  - name: Configure frontend server
    hosts: frontend
    roles:
      - frontend
```

Backend Files:

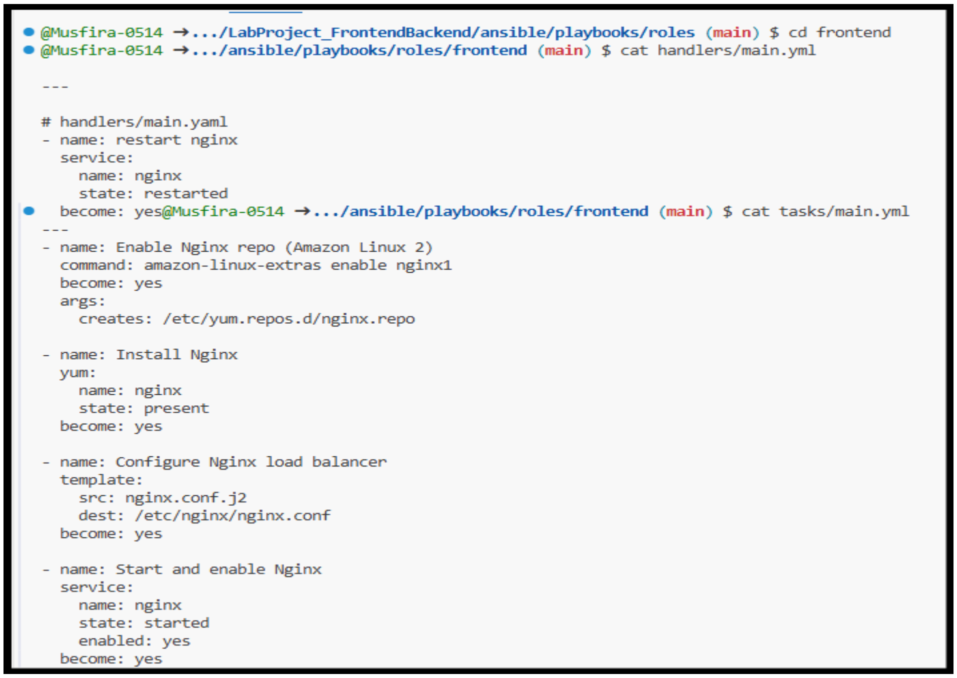


```
● @Musfira-0514 ➜.../ansible/playbooks/roles/backend (main) $ cd handlers
● @Musfira-0514 ➜.../playbooks/roles/backend/handlers (main) $ cat main.yml
  - name: Restart httpd
    service:
      name: httpd
      state: restarted
● @Musfira-0514 ➜.../playbooks/roles/backend/handlers (main) $ cd ..
● @Musfira-0514 ➜.../ansible/playbooks/roles/backend (main) $ cat tasks/main.yml
  ---
  - name: Install Apache HTTPD
    yum:
      name: httpd
      state: present
    become: yes

  - name: Create custom index page
    template:
      src: backend_index.html.j2
      dest: /var/www/html/index.html
    become: yes

  - name: Start and enable Apache
    service:
      name: httpd
      state: started
      enabled: yes
    become: yes
```

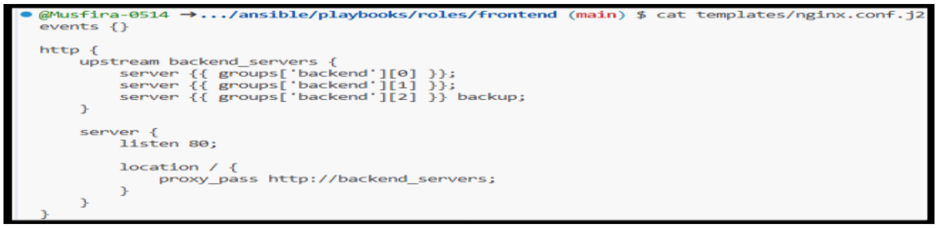


```
    become: yes
● @Musfira-0514 ➜.../ansible/playbooks/roles/backend (main) $ cat templates/backend_index.html.j2
  <!DOCTYPE html>
  <html>
  <head>
      <title>Backend Server</title>
      <style>
          body {
              font-family: serif;
          }
          h1 {
              font-size: 48px;
          }
          p {
              font-size: 22px;
          }
      </style>
  </head>
  <body>

  <h1>Backend Server</h1>

  <p><strong>Hostname:</strong> {{ ansible_host }}</p>
  <p><strong>Private IP:</strong> {{ ansible_default_ipv4.address }}</p>

  </body>
  </html>
```

Frontend Files:

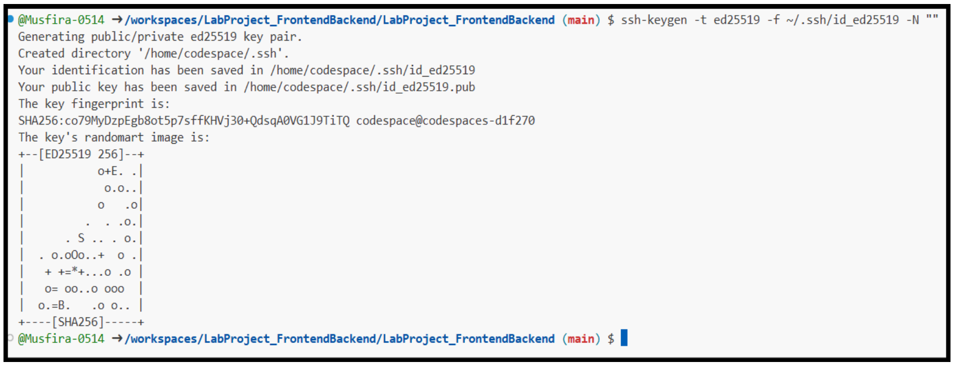


```
● @Musfira-0514 ➜.../LabProject_FrontendBackend/ansible/playbooks/roles (main) $ cd frontend
● @Musfira-0514 ➜.../ansible/playbooks/roles/frontend (main) $ cat handlers/main.yml

  ---

  # handlers/main.yaml
  - name: restart nginx
    service:
      name: nginx
      state: restarted
●   become: yes@Musfira-0514 ➜.../ansible/playbooks/roles/frontend (main) $ cat tasks/main.yml
  ---
  - name: Enable Nginx repo (Amazon Linux 2)
    command: amazon-linux-extras enable nginx1
    become: yes
    args:
      creates: /etc/yum.repos.d/nginx.repo

  - name: Install Nginx
    yum:
      name: nginx
      state: present
    become: yes

  - name: Configure Nginx load balancer
    template:
      src: nginx.conf.j2
      dest: /etc/nginx/nginx.conf
    become: yes

  - name: Start and enable Nginx
    service:
      name: nginx
      state: started
      enabled: yes
    become: yes
```



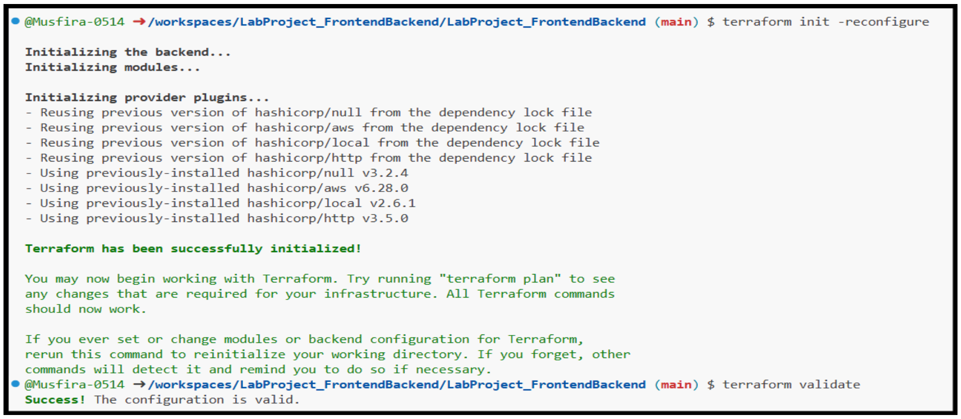
```
@Musfira-0514 →.../ansible/playbooks/roles/frontend (main) $ cat templates/nginx.conf.j2
events {}

http {
    upstream backend_servers {
        server {{ groups['backend'][0] }};
        server {{ groups['backend'][1] }};
        server {{ groups['backend'][2] }} backup;
    }

    server {
        listen 80;

        location / {
            proxy_pass http://backend_servers;
        }
    }
}
```

Key generation:



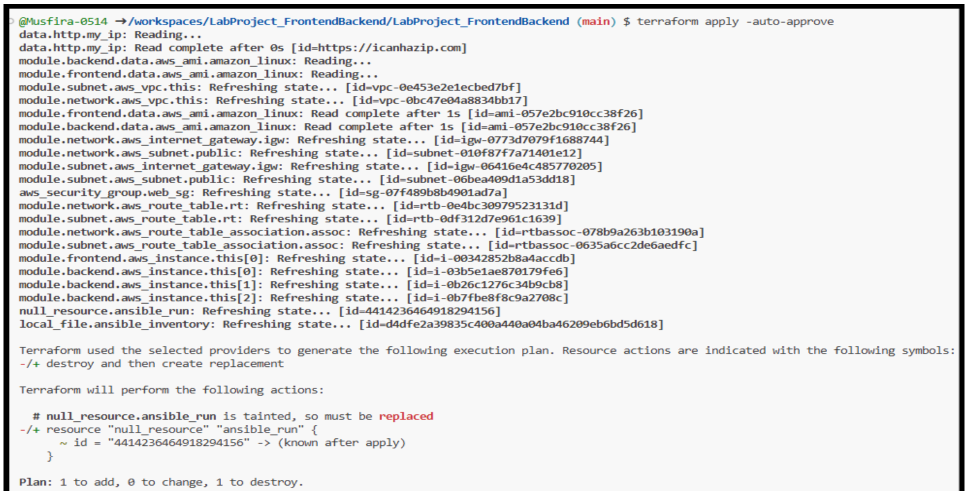
```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
Created directory '/home/codespace/.ssh'.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:co79MyDzpEgb8ot5p7sffKHVj30+QdsqA0VG1J9TiTQ codespace@codespaces-d1f270
The key's randomart image is:
+--[ED25519 256]--+
|           o+E. .|
|            o.o..|
|           o   .o|
|         .  . .o.|
|       . S .. . o.|
|   . o.oOo..+  o .|
|    + +=*+...o .o |
|    o= oo..o ooo  |
|   o.=B.   .o o.. |
+----[SHA256]-----+
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

Terraform init & validate:

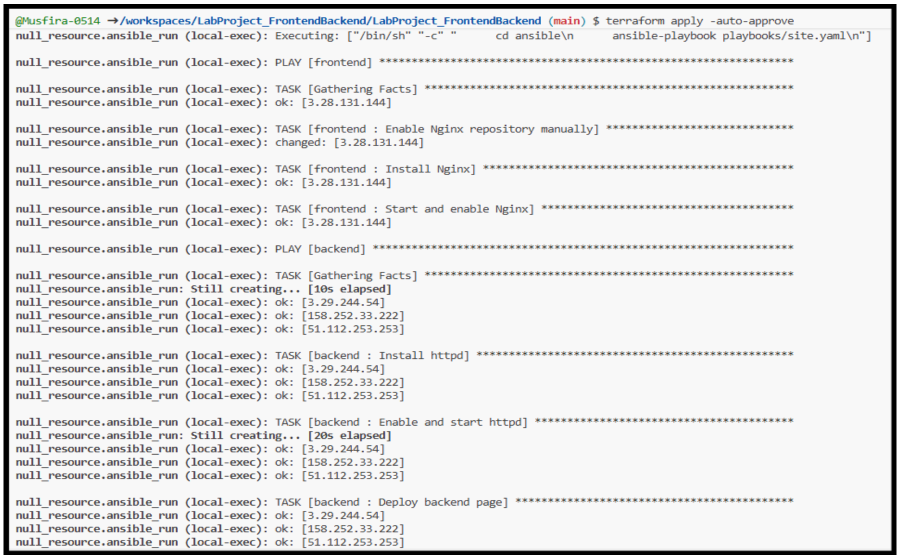


```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ terraform init -reconfigure

Initializing the backend...
Initializing modules...

Initializing provider plugins...
- Reusing previous version of hashicorp/null from the dependency lock file
- Reusing previous version of hashicorp/aws from the dependency lock file
- Reusing previous version of hashicorp/local from the dependency lock file
- Reusing previous version of hashicorp/http from the dependency lock file
- Using previously-installed hashicorp/null v3.2.4
- Using previously-installed hashicorp/aws v6.28.0
- Using previously-installed hashicorp/local v2.6.1
- 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.
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ terraform validate
Success! The configuration is valid.
```

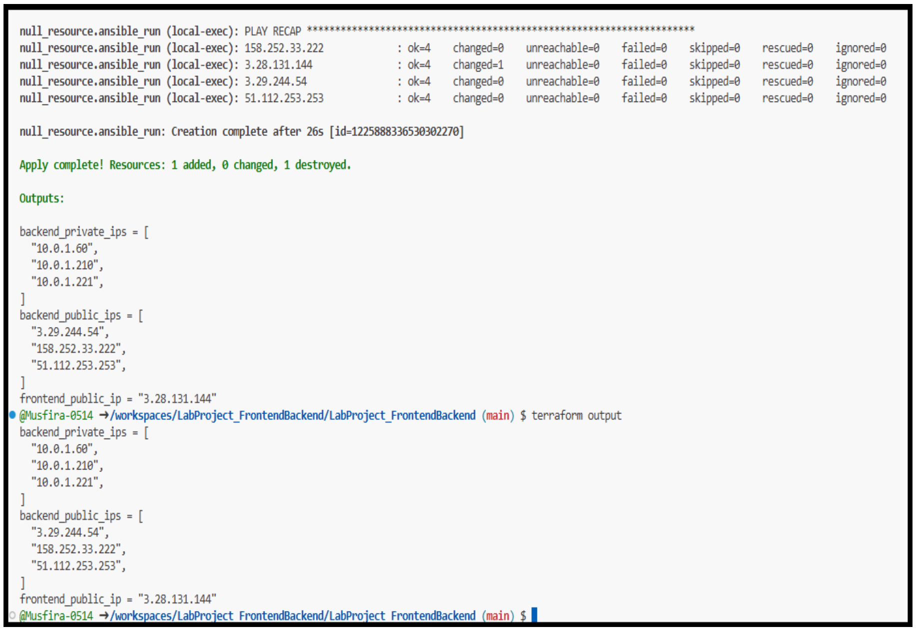
Terraform auto approve:



```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ terraform apply -auto-approve
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
module.backend.data.aws_ami.amazon_linux: Reading...
module.frontend.data.aws_ami.amazon_linux: Reading...
module.subnet.aws_vpc.this: Refreshing state... [id=vpc-0e453e2e1ecbed7bf]
module.network.aws_vpc.this: Refreshing state... [id=vpc-0bc47e04a8834bb17]
module.frontend.data.aws_ami.amazon_linux: Read complete after 1s [id=ami-057e2bc910cc38f26]
module.backend.data.aws_ami.amazon_linux: Read complete after 1s [id=ami-057e2bc910cc38f26]
module.network.aws_internet_gateway.igw: Refreshing state... [id=igw-0773d7079f1688744]
module.network.aws_subnet.public: Refreshing state... [id=subnet-010f87f7a71401e12]
module.subnet.aws_internet_gateway.igw: Refreshing state... [id=igw-06416e4c485770205]
module.subnet.aws_subnet.public: Refreshing state... [id=subnet-06bea409d1a53dd18]
aws_security_group.web_sg: Refreshing state... [id=sg-07f489b8b4901ad7a]
module.network.aws_route_table.rt: Refreshing state... [id=rtb-0e4bc30979523131d]
module.subnet.aws_route_table.rt: Refreshing state... [id=rtb-0df312d7e961c1639]
module.network.aws_route_table_association.assoc: Refreshing state... [id=rtbassoc-078b9a263b103190a]
module.subnet.aws_route_table_association.assoc: Refreshing state... [id=rtbassoc-0635a6cc2de6aedfc]
module.frontend.aws_instance.this[0]: Refreshing state... [id=i-00342852b8a4accdb]
module.backend.aws_instance.this[0]: Refreshing state... [id=i-03b5e1ae870179fe6]
module.backend.aws_instance.this[1]: Refreshing state... [id=i-0b26c1276c34b9cb8]
module.backend.aws_instance.this[2]: Refreshing state... [id=i-0b7fbe8f8c9a2708c]
null_resource.ansible_run: Refreshing state... [id=4414236464918294156]
local_file.ansible_inventory: Refreshing state... [id=d4dfe2a39835c400a440a04ba46209eb6bd5d618]

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

Terraform will perform the following actions:

  # null_resource.ansible_run is tainted, so must be replaced
-/+ resource "null_resource" "ansible_run" {
      ~ id = "4414236464918294156" -> (known after apply)
    }

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

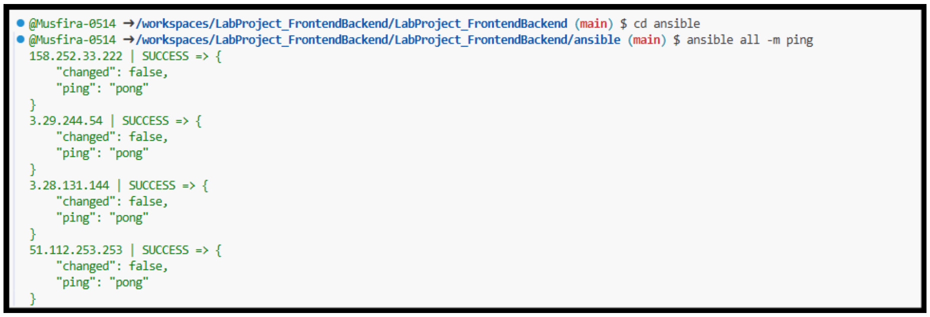


```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ terraform apply -auto-approve
null_resource.ansible_run (local-exec): Executing: ["/bin/sh" "-c" "      cd ansible\n      ansible-playbook playbooks/site.yaml\n"]

null_resource.ansible_run (local-exec): PLAY [frontend] *****************************************************************

null_resource.ansible_run (local-exec): TASK [Gathering Facts] **********************************************************
null_resource.ansible_run (local-exec): ok: [3.28.131.144]

null_resource.ansible_run (local-exec): TASK [frontend : Enable Nginx repository manually] ******************************
null_resource.ansible_run (local-exec): changed: [3.28.131.144]

null_resource.ansible_run (local-exec): TASK [frontend : Install Nginx] *************************************************
null_resource.ansible_run (local-exec): ok: [3.28.131.144]

null_resource.ansible_run (local-exec): TASK [frontend : Start and enable Nginx] ****************************************
null_resource.ansible_run (local-exec): ok: [3.28.131.144]

null_resource.ansible_run (local-exec): PLAY [backend] ******************************************************************

null_resource.ansible_run (local-exec): TASK [Gathering Facts] **********************************************************
null_resource.ansible_run: Still creating... [10s elapsed]
null_resource.ansible_run (local-exec): ok: [3.29.244.54]
null_resource.ansible_run (local-exec): ok: [158.252.33.222]
null_resource.ansible_run (local-exec): ok: [51.112.253.253]

null_resource.ansible_run (local-exec): TASK [backend : Install httpd] **************************************************
null_resource.ansible_run (local-exec): ok: [3.29.244.54]
null_resource.ansible_run (local-exec): ok: [158.252.33.222]
null_resource.ansible_run (local-exec): ok: [51.112.253.253]

null_resource.ansible_run (local-exec): TASK [backend : Enable and start httpd] *****************************************
null_resource.ansible_run: Still creating... [20s elapsed]
null_resource.ansible_run (local-exec): ok: [3.29.244.54]
null_resource.ansible_run (local-exec): ok: [158.252.33.222]
null_resource.ansible_run (local-exec): ok: [51.112.253.253]

null_resource.ansible_run (local-exec): TASK [backend : Deploy backend page] ********************************************
null_resource.ansible_run (local-exec): ok: [3.29.244.54]
null_resource.ansible_run (local-exec): ok: [158.252.33.222]
null_resource.ansible_run (local-exec): ok: [51.112.253.253]
```

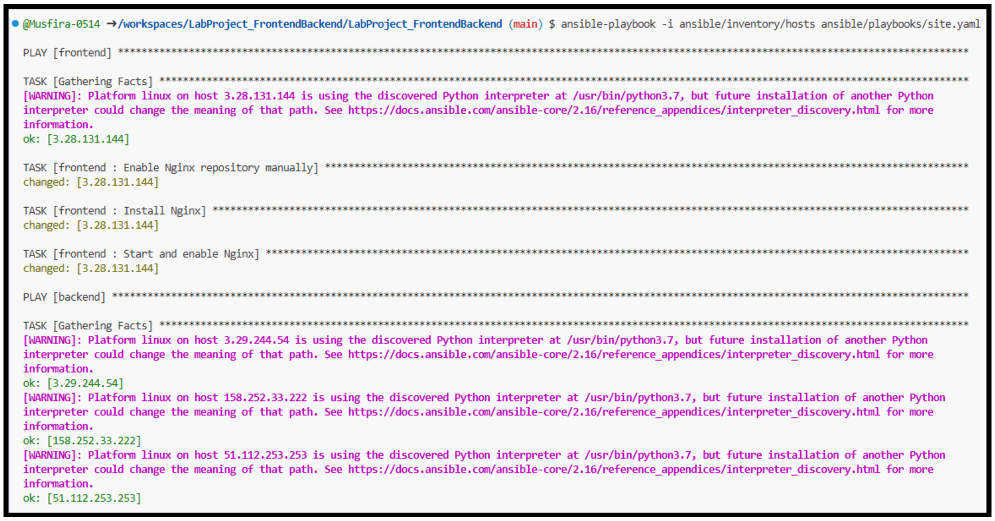


```
null_resource.ansible_run (local-exec): PLAY RECAP *********************************************************************
null_resource.ansible_run (local-exec): 158.252.33.222             : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
null_resource.ansible_run (local-exec): 3.28.131.144               : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
null_resource.ansible_run (local-exec): 3.29.244.54                : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
null_resource.ansible_run (local-exec): 51.112.253.253             : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

null_resource.ansible_run: Creation complete after 26s [id=1225888336530302270]

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

Outputs:

backend_private_ips = [
  "10.0.1.60",
  "10.0.1.210",
  "10.0.1.221",
]
backend_public_ips = [
  "3.29.244.54",
  "158.252.33.222",
  "51.112.253.253",
]
frontend_public_ip = "3.28.131.144"
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ terraform output
backend_private_ips = [
  "10.0.1.60",
  "10.0.1.210",
  "10.0.1.221",
]
backend_public_ips = [
  "3.29.244.54",
  "158.252.33.222",
  "51.112.253.253",
]
frontend_public_ip = "3.28.131.144"
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $
```

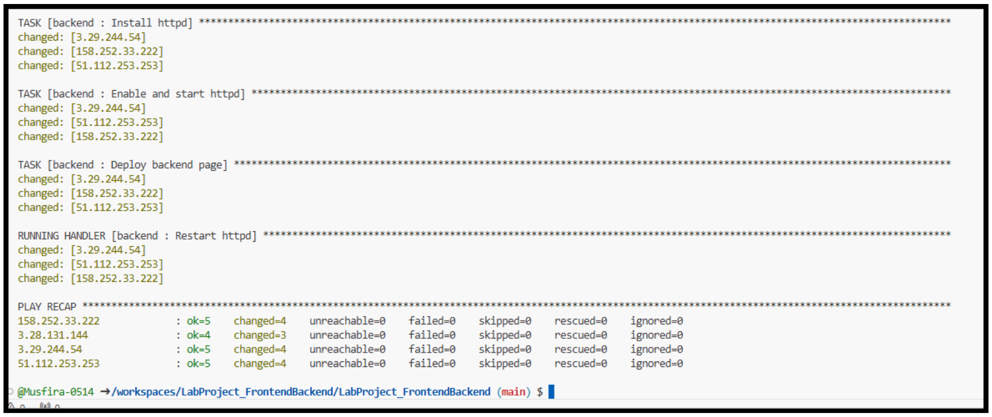
Testing ansible playbooks/roles:



```
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ cd ansible
@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend/ansible (main) $ ansible all -m ping
158.252.33.222 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
3.29.244.54 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
3.28.131.144 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
51.112.253.253 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```



```
● @Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ ansible-playbook -i ansible/inventory/hosts ansible/playbooks/site.yaml

PLAY [frontend] *********************************************************************************************************************

TASK [Gathering Facts] **************************************************************************************************************
[WARNING]: Platform linux on host 3.28.131.144 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: [3.28.131.144]

TASK [frontend : Enable Nginx repository manually] **********************************************************************************
changed: [3.28.131.144]

TASK [frontend : Install Nginx] *****************************************************************************************************
changed: [3.28.131.144]

TASK [frontend : Start and enable Nginx] ********************************************************************************************
changed: [3.28.131.144]

PLAY [backend] **********************************************************************************************************************

TASK [Gathering Facts] **************************************************************************************************************
[WARNING]: Platform linux on host 3.29.244.54 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: [3.29.244.54]
[WARNING]: Platform linux on host 158.252.33.222 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: [158.252.33.222]
[WARNING]: Platform linux on host 51.112.253.253 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: [51.112.253.253]
```



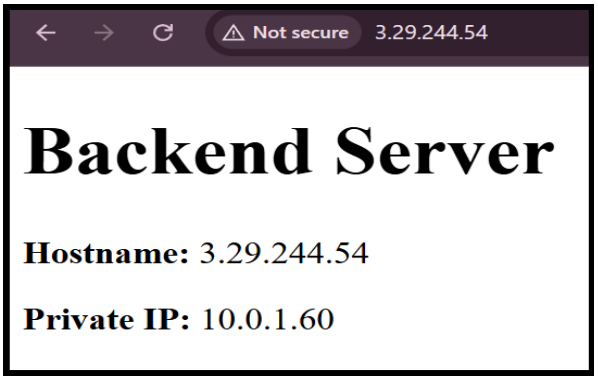
```
TASK [backend : Install httpd] ******************************************************************************************************
changed: [3.29.244.54]
changed: [158.252.33.222]
changed: [51.112.253.253]

TASK [backend : Enable and start httpd] *********************************************************************************************
changed: [3.29.244.54]
changed: [51.112.253.253]
changed: [158.252.33.222]

TASK [backend : Deploy backend page] ************************************************************************************************
changed: [3.29.244.54]
changed: [158.252.33.222]
changed: [51.112.253.253]

RUNNING HANDLER [backend : Restart httpd] *******************************************************************************************
changed: [3.29.244.54]
changed: [51.112.253.253]
changed: [158.252.33.222]

PLAY RECAP **************************************************************************************************************************
158.252.33.222             : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
3.28.131.144               : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
3.29.244.54                : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
51.112.253.253             : ok=5    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

@Musfira-0514 →/workspaces/LabProject_FrontendBackend/LabProject_FrontendBackend (main) $ 
```

Verification Results

Backend URLs

- http://3.29.244.54
- http://158.252.33.222
- http://51.112.253.253

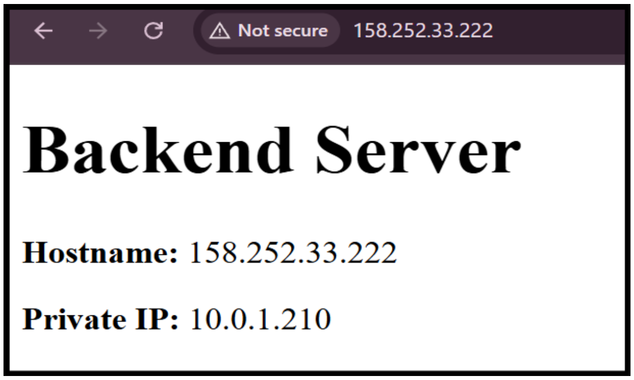
Each backend displays a unique page.


⚠ Not secure 3.29.244.54

Backend Server

Hostname: 3.29.244.54

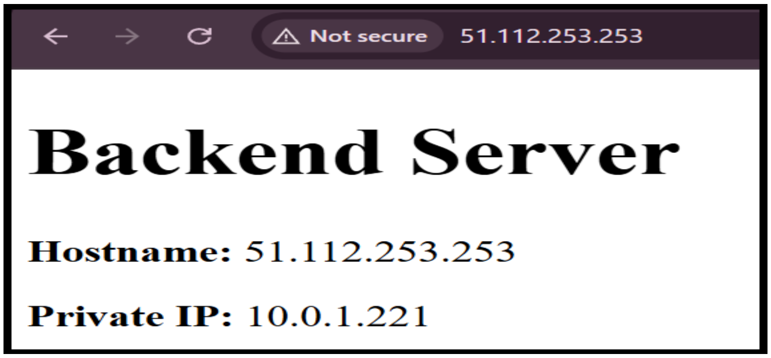
Private IP: 10.0.1.60


⚠ Not secure 158.252.33.222

Backend Server

Hostname: 158.252.33.222

Private IP: 10.0.1.210


⚠ Not secure 51.112.253.253

Backend Server

Hostname: 51.112.253.253

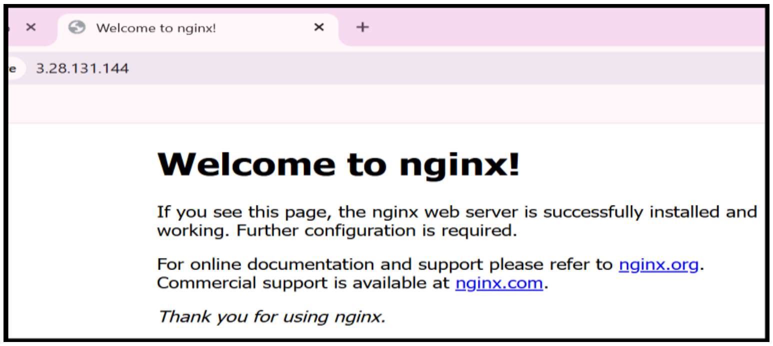
Private IP: 10.0.1.221

Frontend URL

http://3.28.131.144

Alternates between backend 1 and backend 2

Serves backup backend when primaries are down


Welcome to nginx!

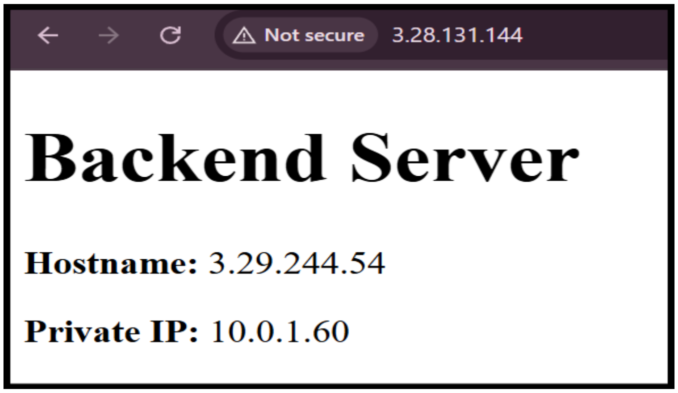
3.28.131.144

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

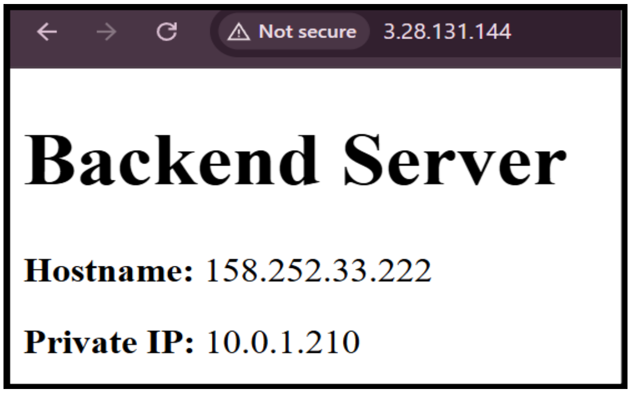
Thank you for using nginx.


⚠ Not secure 3.28.131.144

Backend Server

Hostname: 3.29.244.54

Private IP: 10.0.1.60


⚠ Not secure 3.28.131.144

Backend Server

Hostname: 158.252.33.222

Private IP: 10.0.1.210

Conclusion

This project successfully demonstrates a fully automated, highly available web architecture using industry-standard tools. The use of Terraform and Ansible roles ensures scalability, maintainability, and repeatability of deployments.

Repository link

https://github.com/Musfira-0514/LabProject_FrontendBackend

https://github.com/codespaces/probable-space-invention-qv9q7jx5rgr249j4