Name: Rafia Fiaz

Roll No: 2023-BSE-051

Class: BSE 5-B

Submitted To: Sir Shoaib

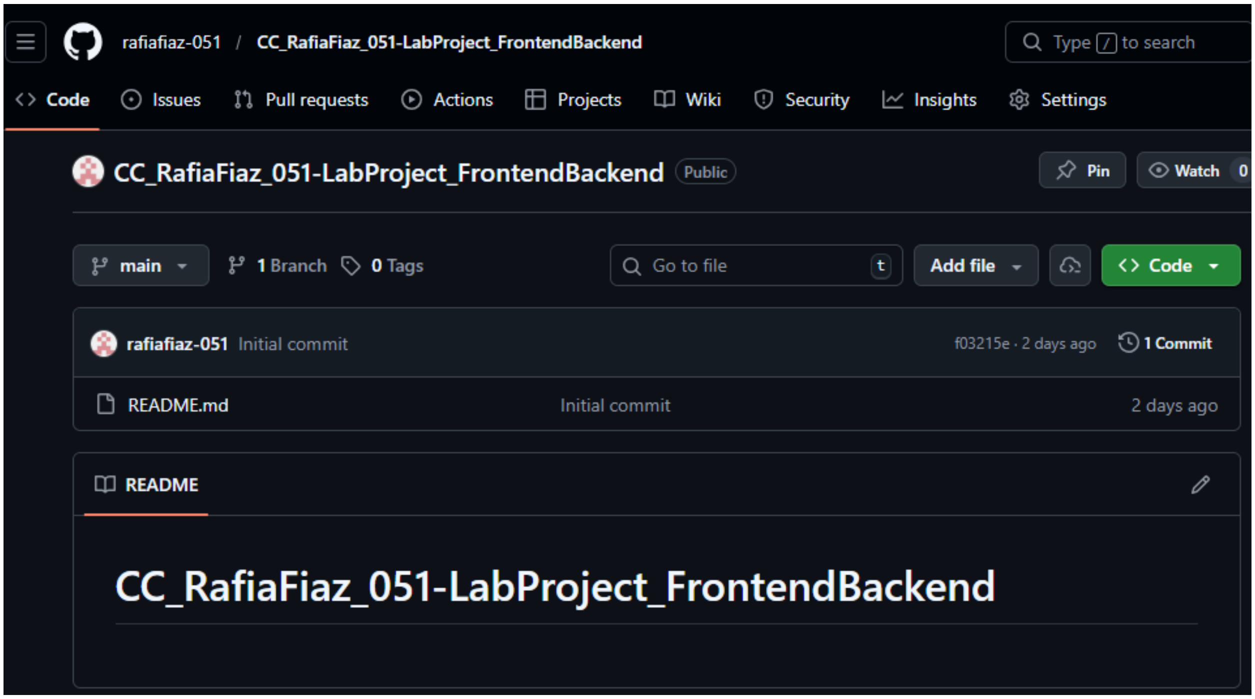
Open-Ended Lab

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

+--+

PHASE 0 — INITIAL SETUP

0.1 Create GitHub Repository



rafiafiaz-051 / CC_RafiaFiaz_051-LabProject_FrontendBackend

CC_RafiaFiaz_051-LabProject_FrontendBackend Public

main 1 Branch 0 Tags

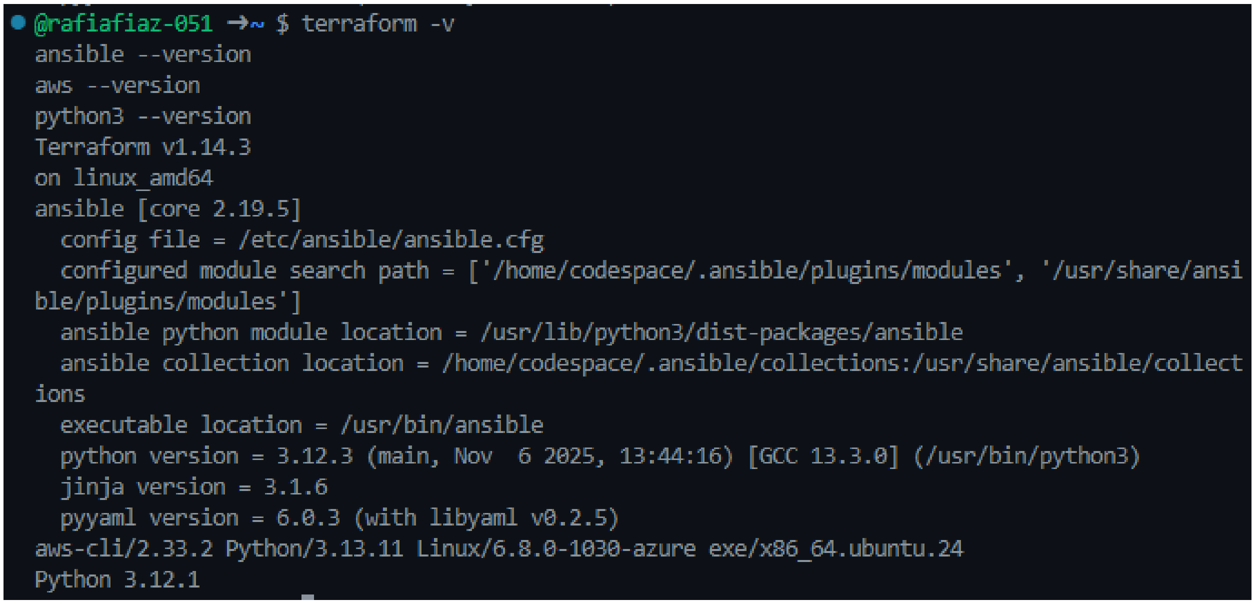
rafiafiaz-051 Initial commit f03215e · 2 days ago 1 Commit

README.md Initial commit 2 days ago

README

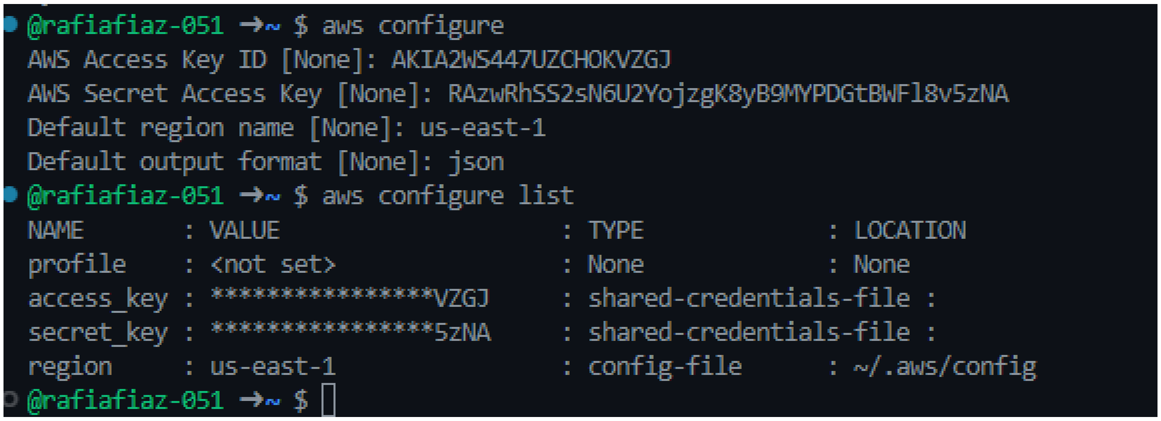
CC_RafiaFiaz_051-LabProject_FrontendBackend

0.2 Install Required Tools (if not already)



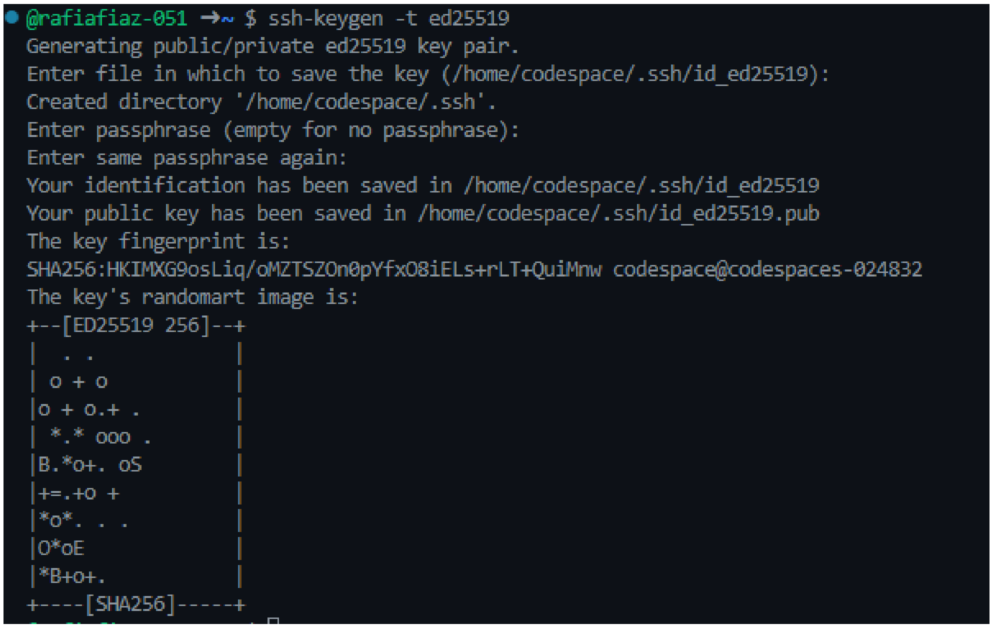
```
@rafiafiaz-051 →~ $ terraform -v
ansible --version
aws --version
python3 --version
Terraform v1.14.3
on linux_amd64
ansible [core 2.19.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/codespace/.ansible/plugins/modules', '/usr/share/ansi
ble/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/codespace/.ansible/collections:/usr/share/ansible/collect
ions
  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
  pyyaml version = 6.0.3 (with libyaml v0.2.5)
aws-cli/2.33.2 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
Python 3.12.1
```

0.3 Configure AWS Credentials



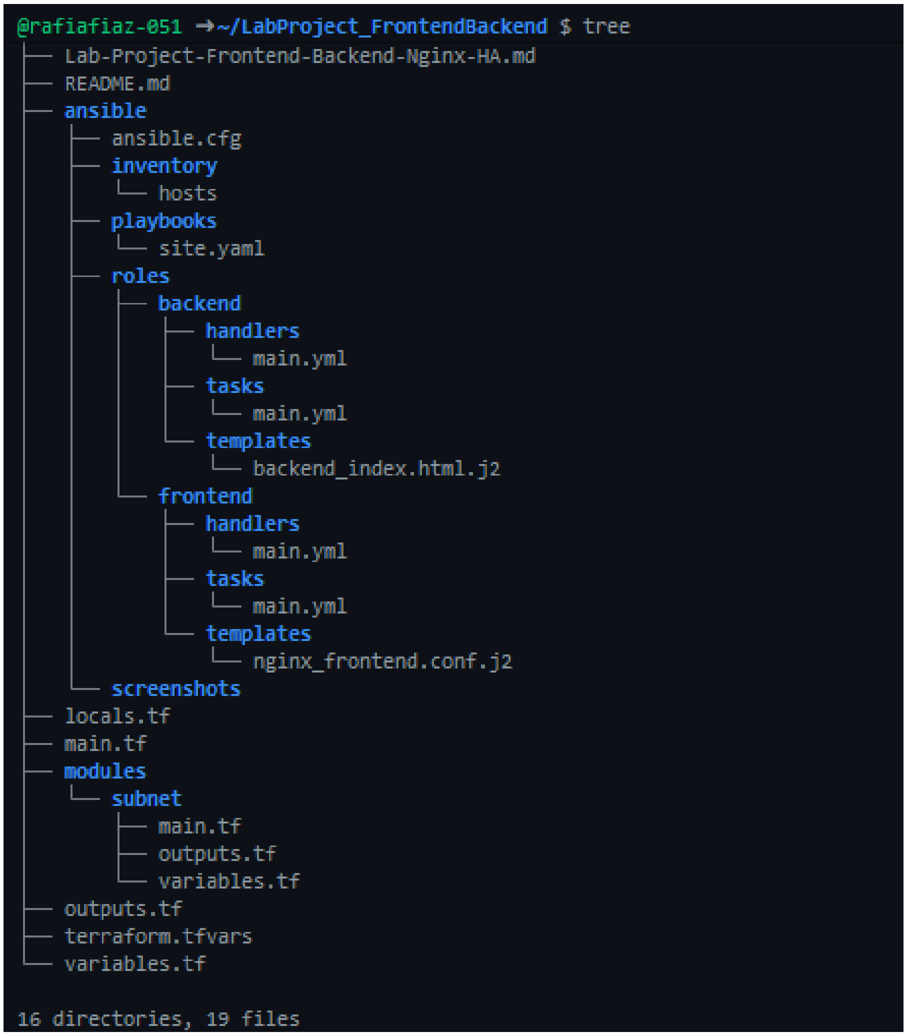
```
@rafiafiaz-051 →~ $ aws configure
AWS Access Key ID [None]: AKIA2WS447UZCHOKVZGJ
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
@rafiafiaz-051 →~ $ aws configure list
NAME       : VALUE                    : TYPE               : LOCATION
profile    : <not set>                : None               : None
access_key : ****************VZGJ     : shared-credentials-file :
secret_key : ****************5zNA     : shared-credentials-file :
region     : us-east-1                : config-file        : ~/.aws/config
@rafiafiaz-051 →~ $
```

0.4 Generate SSH Key



```
@rafiafiaz-051 ➜~ $ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/codespace/.ssh/id_ed25519):
Created directory '/home/codespace/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
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:HKIMXG9osLiq/oMZTSZOn0pYfxO8iELs+rLT+QuiMnw codespace@codespaces-024832
The key's randomart image is:
+--[ED25519 256]--+
|   . .           |
|  o + o          |
| o + o.+ .       |
|  *.* ooo .      |
|B.*o+. oS        |
|+=.+o +          |
|*o*. . .         |
|O*oE             |
|*B+o+.           |
+----[SHA256]-----+
```

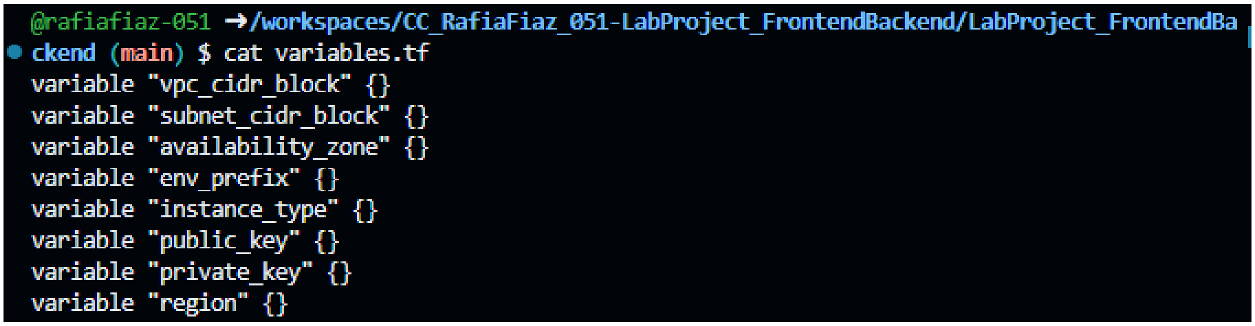
PHASE 1 — PROJECT STRUCTURE



```
@rafiafiaz-051 ➜~/LabProject_FrontendBackend $ tree
├── Lab-Project-Frontend-Backend-Nginx-HA.md
├── README.md
├── ansible
│   ├── ansible.cfg
│   ├── inventory
│   │   └── hosts
│   ├── playbooks
│   │   └── site.yaml
│   ├── roles
│   │   ├── backend
│   │   │   ├── handlers
│   │   │   │   └── main.yml
│   │   │   ├── tasks
│   │   │   │   └── main.yml
│   │   │   └── templates
│   │   │       └── backend_index.html.j2
│   │   └── frontend
│   │       ├── handlers
│   │       │   └── main.yml
│   │       ├── tasks
│   │       │   └── main.yml
│   │       └── templates
│   │           └── nginx_frontend.conf.j2
│   └── screenshots
├── locals.tf
├── main.tf
├── modules
│   └── subnet
│       ├── main.tf
│       ├── outputs.tf
│       └── variables.tf
├── outputs.tf
├── terraform.tfvars
└── variables.tf

16 directories, 19 files
```

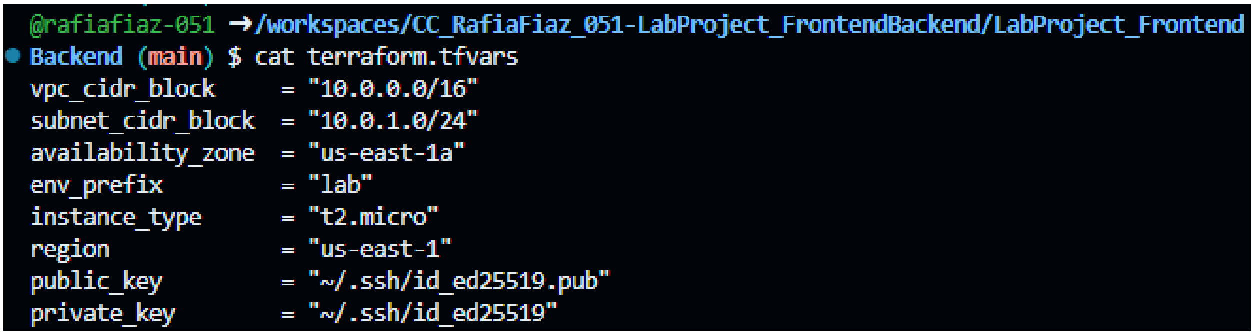
PHASE 2 — TERRAFORM NETWORKING

2.1 variables.tf



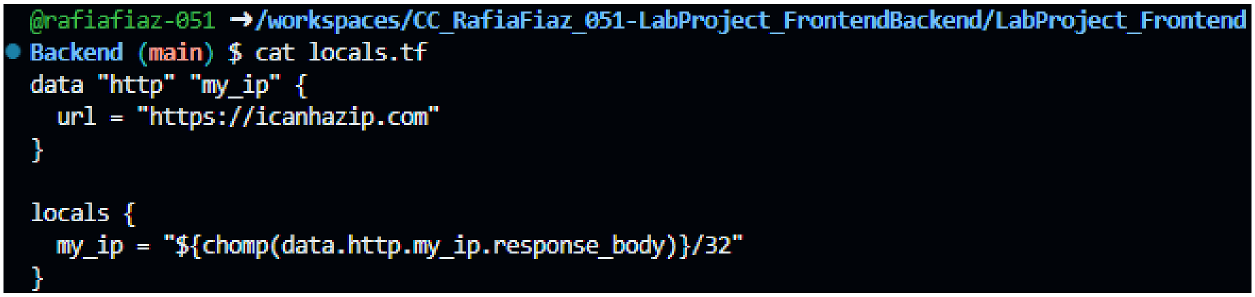
```
  @rafiafiaz-051 →/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_FrontendBa
● ckend (main) $ cat variables.tf
  variable "vpc_cidr_block" {}
  variable "subnet_cidr_block" {}
  variable "availability_zone" {}
  variable "env_prefix" {}
  variable "instance_type" {}
  variable "public_key" {}
  variable "private_key" {}
  variable "region" {}
```

2.2 terraform.tfvars



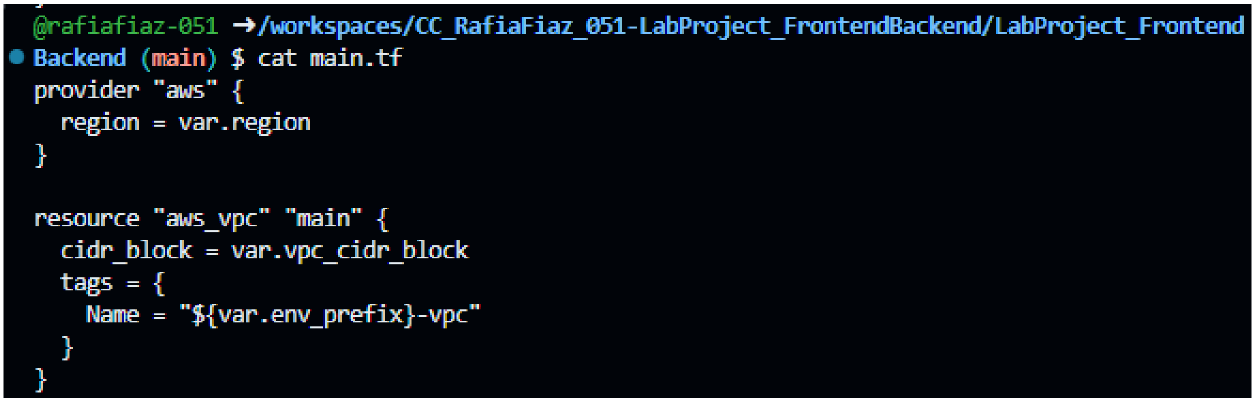
```
  @rafiafiaz-051 →/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
● Backend (main) $ cat terraform.tfvars
  vpc_cidr_block     = "10.0.0.0/16"
  subnet_cidr_block  = "10.0.1.0/24"
  availability_zone  = "us-east-1a"
  env_prefix         = "lab"
  instance_type      = "t2.micro"
  region             = "us-east-1"
  public_key         = "~/.ssh/id_ed25519.pub"
  private_key        = "~/.ssh/id_ed25519"
```

2.3 locals.tf (Your IP auto-detection)



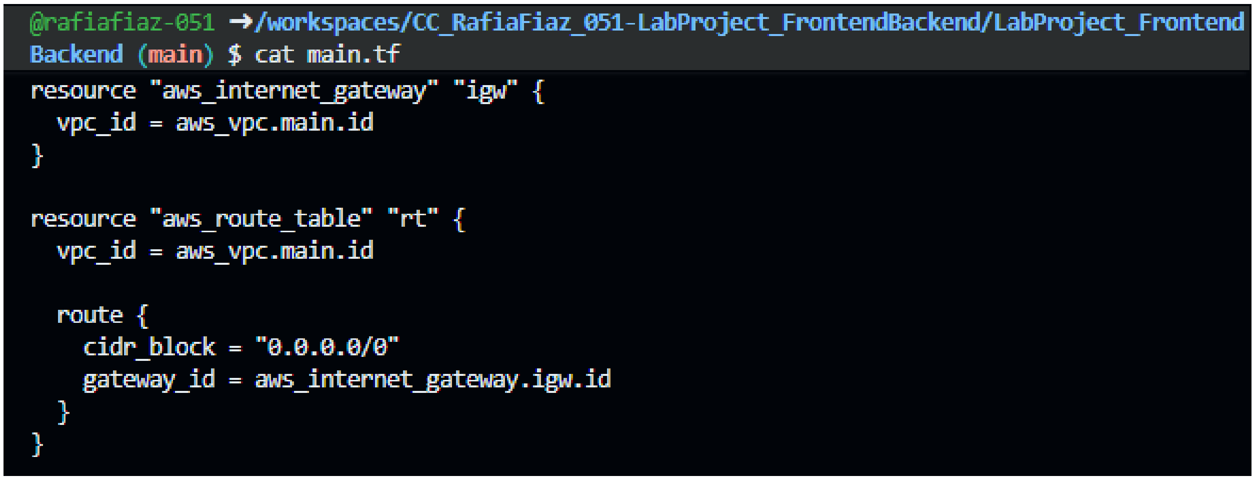
```
  @rafiafiaz-051 →/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
● Backend (main) $ cat locals.tf
  data "http" "my_ip" {
    url = "https://icanhazip.com"
  }

  locals {
    my_ip = "${chomp(data.http.my_ip.response_body)}/32"
  }
```

2.4 main.tf — Provider + VPC



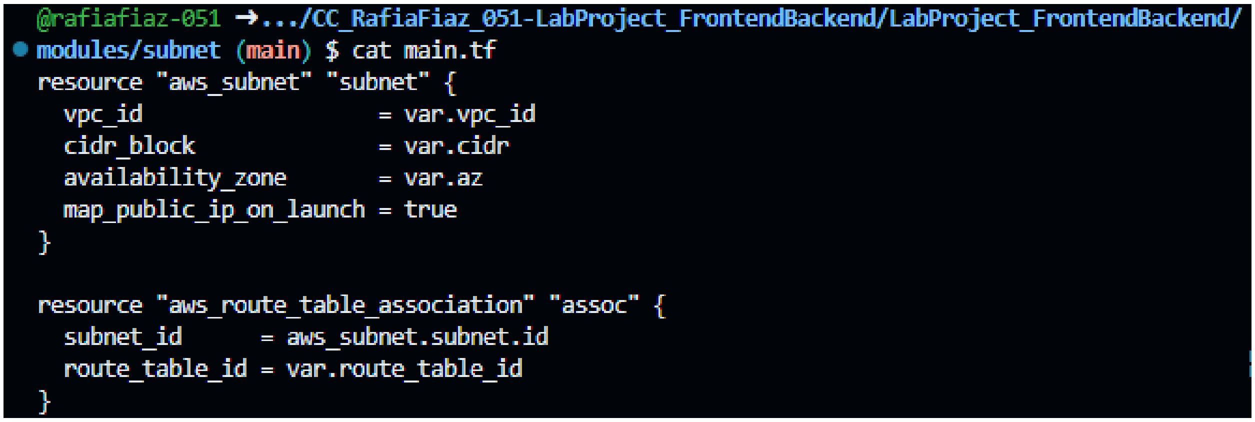
```
@rafiafiaz-051 →/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
● Backend (main) $ cat main.tf
provider "aws" {
  region = var.region
}

resource "aws_vpc" "main" {
  cidr_block = var.vpc_cidr_block
  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}
```

2.5 Internet Gateway + Route Table



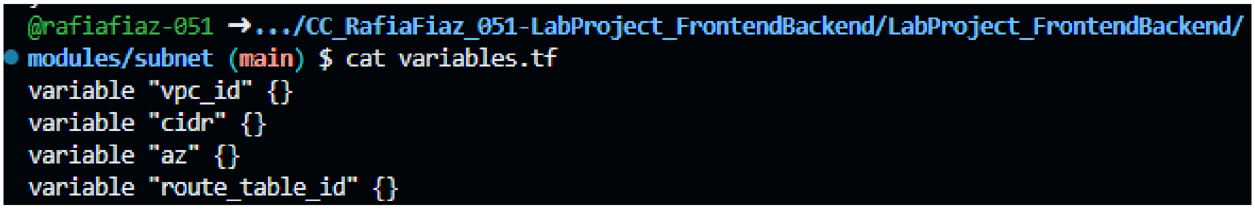
```
@rafiafiaz-051 →/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend (main) $ cat main.tf
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id
}

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

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

2.6 Subnet Module (modules/subnet/main.tf)



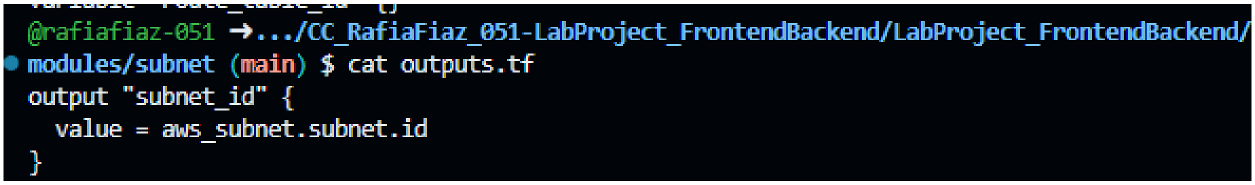
```
@rafiafiaz-051 →.../CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_FrontendBackend/
● modules/subnet (main) $ cat main.tf
resource "aws_subnet" "subnet" {
  vpc_id                  = var.vpc_id
  cidr_block              = var.cidr
  availability_zone       = var.az
  map_public_ip_on_launch = true
}

resource "aws_route_table_association" "assoc" {
  subnet_id      = aws_subnet.subnet.id
  route_table_id = var.route_table_id
}
```

(modules/subnet/variables.tf)



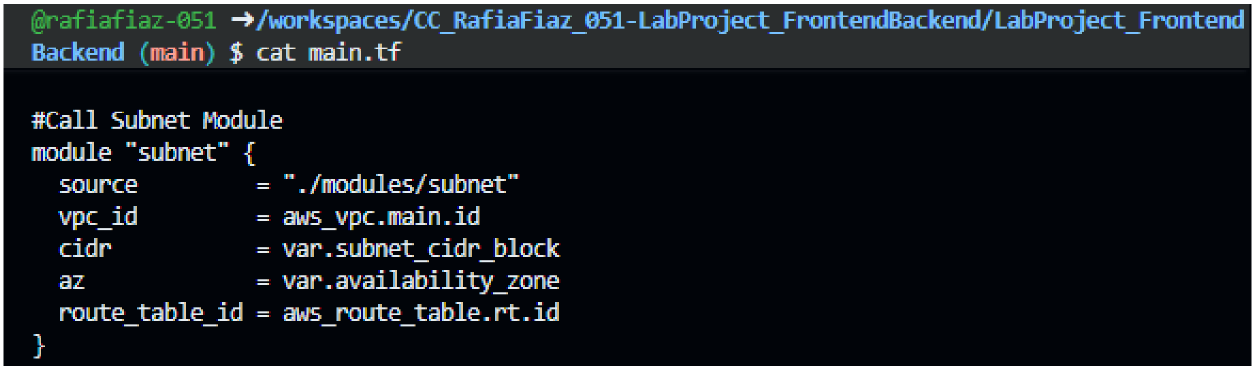
```
@rafiafiaz-051 ➜.../CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_FrontendBackend/
modules/subnet (main) $ cat variables.tf
variable "vpc_id" {}
variable "cidr" {}
variable "az" {}
variable "route_table_id" {}
```

(modules/subnet/outputs.tf)



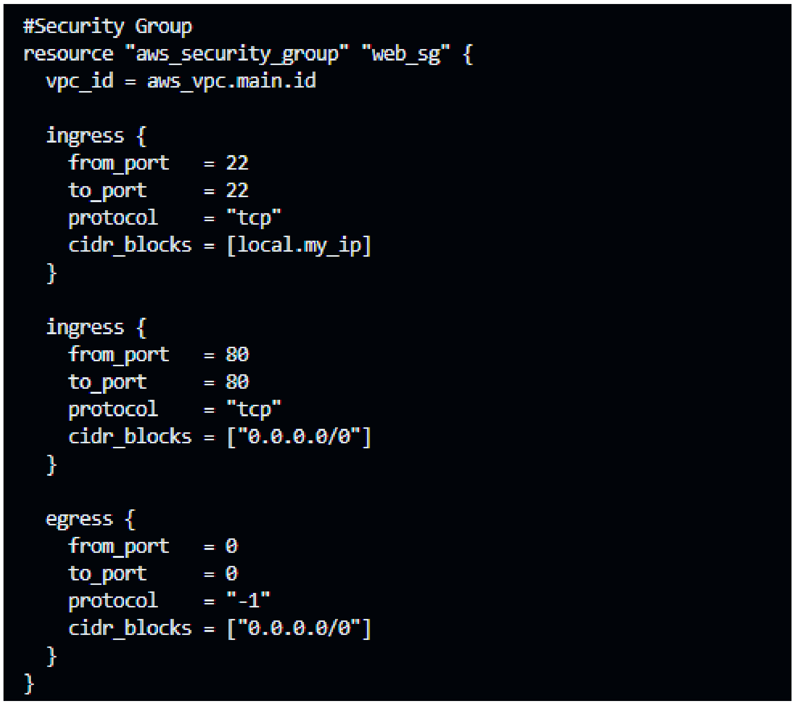
```
@rafiafiaz-051 ➜.../CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_FrontendBackend/
modules/subnet (main) $ cat outputs.tf
output "subnet_id" {
  value = aws_subnet.subnet.id
}
```

2.7 Call Subnet Module(from main.tf)



```
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend (main) $ cat main.tf

#Call Subnet Module
module "subnet" {
  source         = "./modules/subnet"
  vpc_id         = aws_vpc.main.id
  cidr           = var.subnet_cidr_block
  az             = var.availability_zone
  route_table_id = aws_route_table.rt.id
}
```

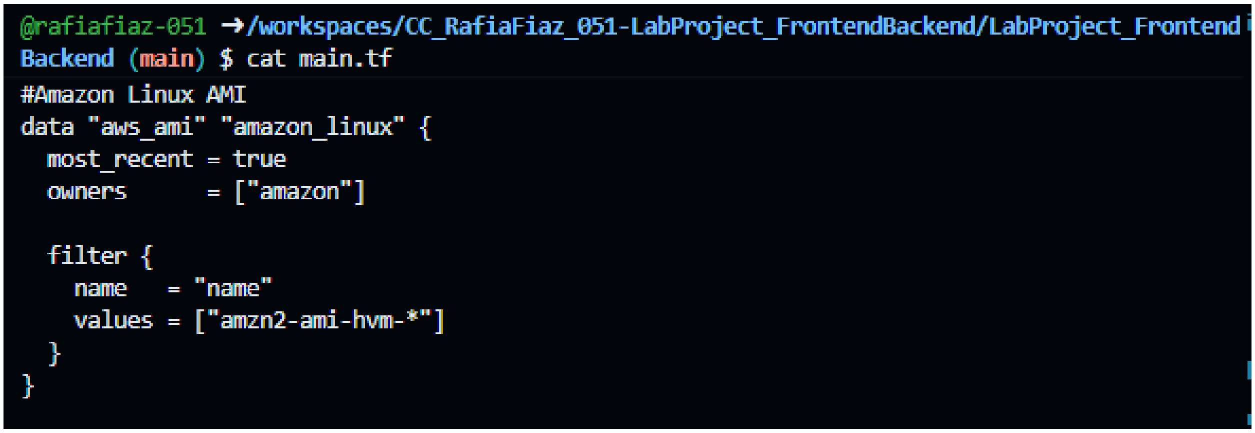
2.8 Security Group



```
#Security Group
resource "aws_security_group" "web_sg" {
  vpc_id = aws_vpc.main.id

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = [local.my_ip]
  }

  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"]
  }
}
```

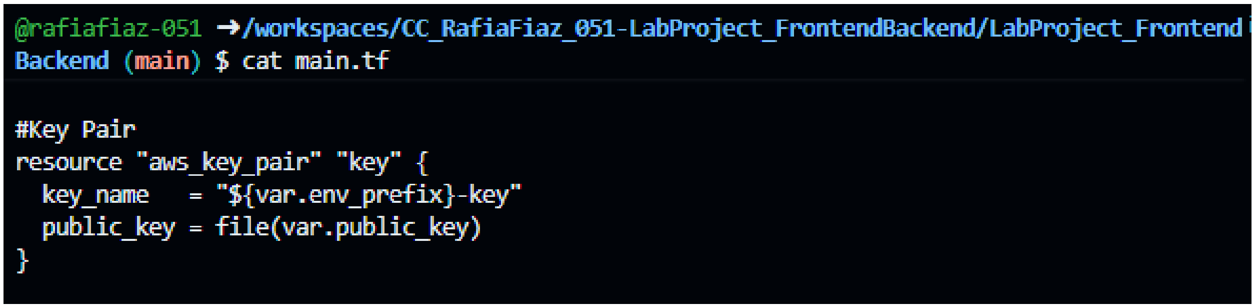
PHASE 3 — EC2 INSTANCES (FRONTEND + BACKENDS)

3.1 Find Amazon Linux AMI



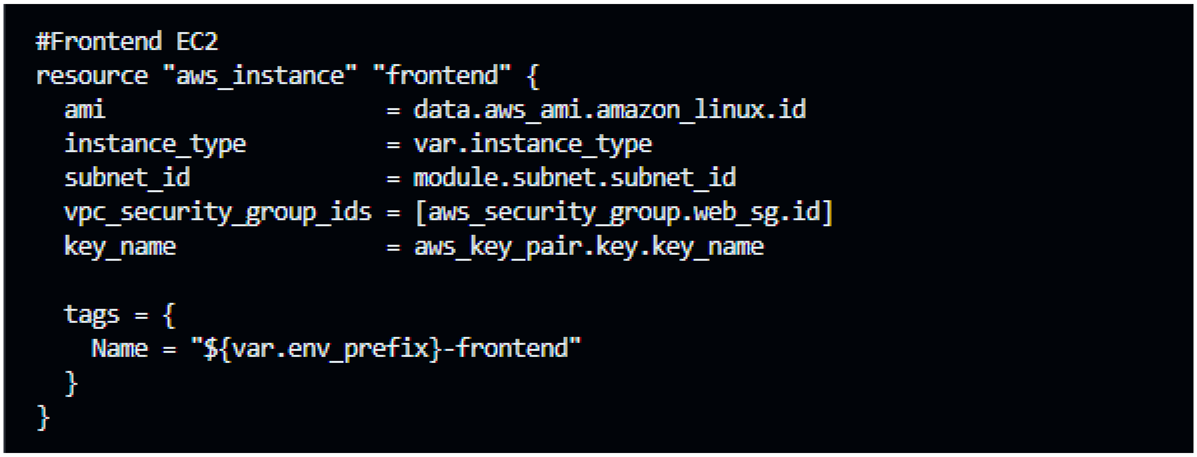
```
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend (main) $ cat main.tf
#Amazon Linux AMI
data "aws_ami" "amazon_linux" {
  most_recent = true
  owners      = ["amazon"]

  filter {
    name   = "name"
    values = ["amzn2-ami-hvm-*"]
  }
}
```

3.2 Key Pair



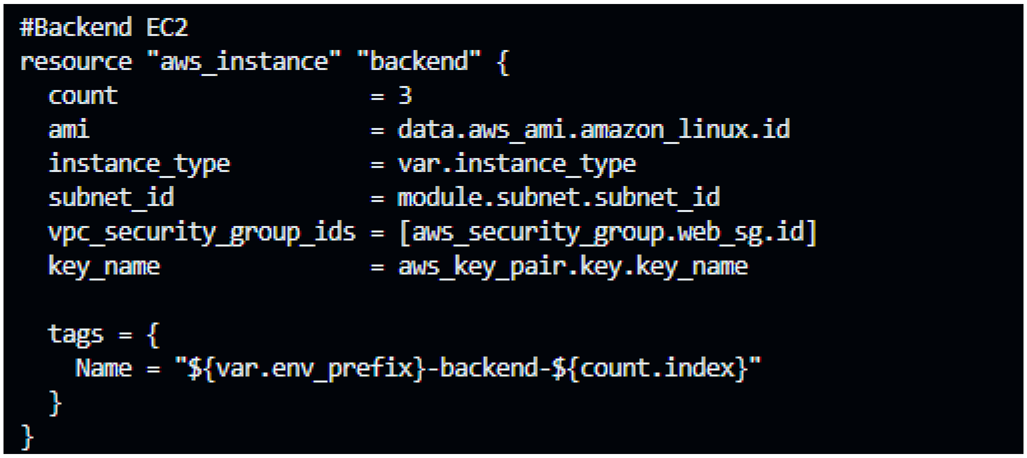
```
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend (main) $ cat main.tf

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

3.3 Frontend EC2



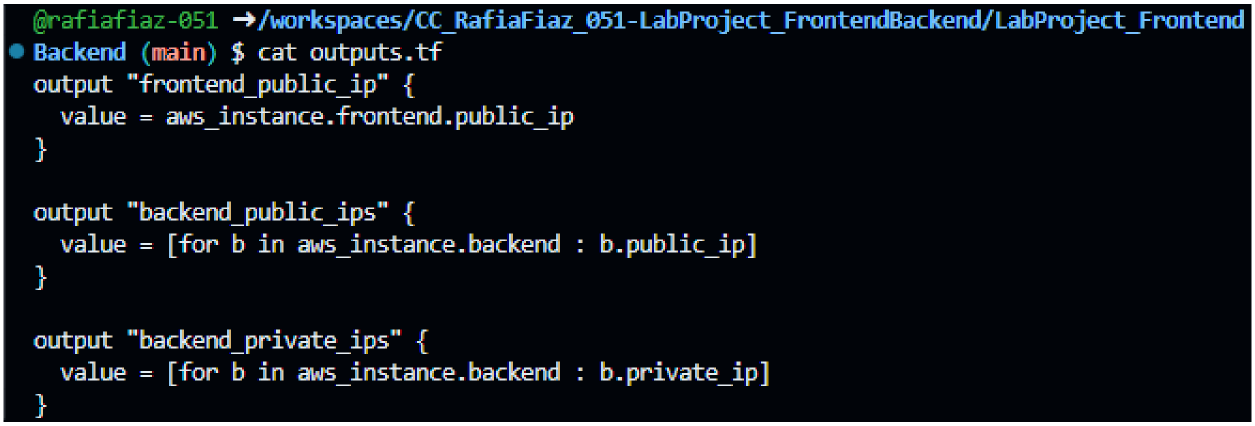
```
#Frontend EC2
resource "aws_instance" "frontend" {
  ami                    = data.aws_ami.amazon_linux.id
  instance_type          = var.instance_type
  subnet_id              = module.subnet.subnet_id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  key_name               = aws_key_pair.key.key_name

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

3.4 Backend EC2 (3 instances)



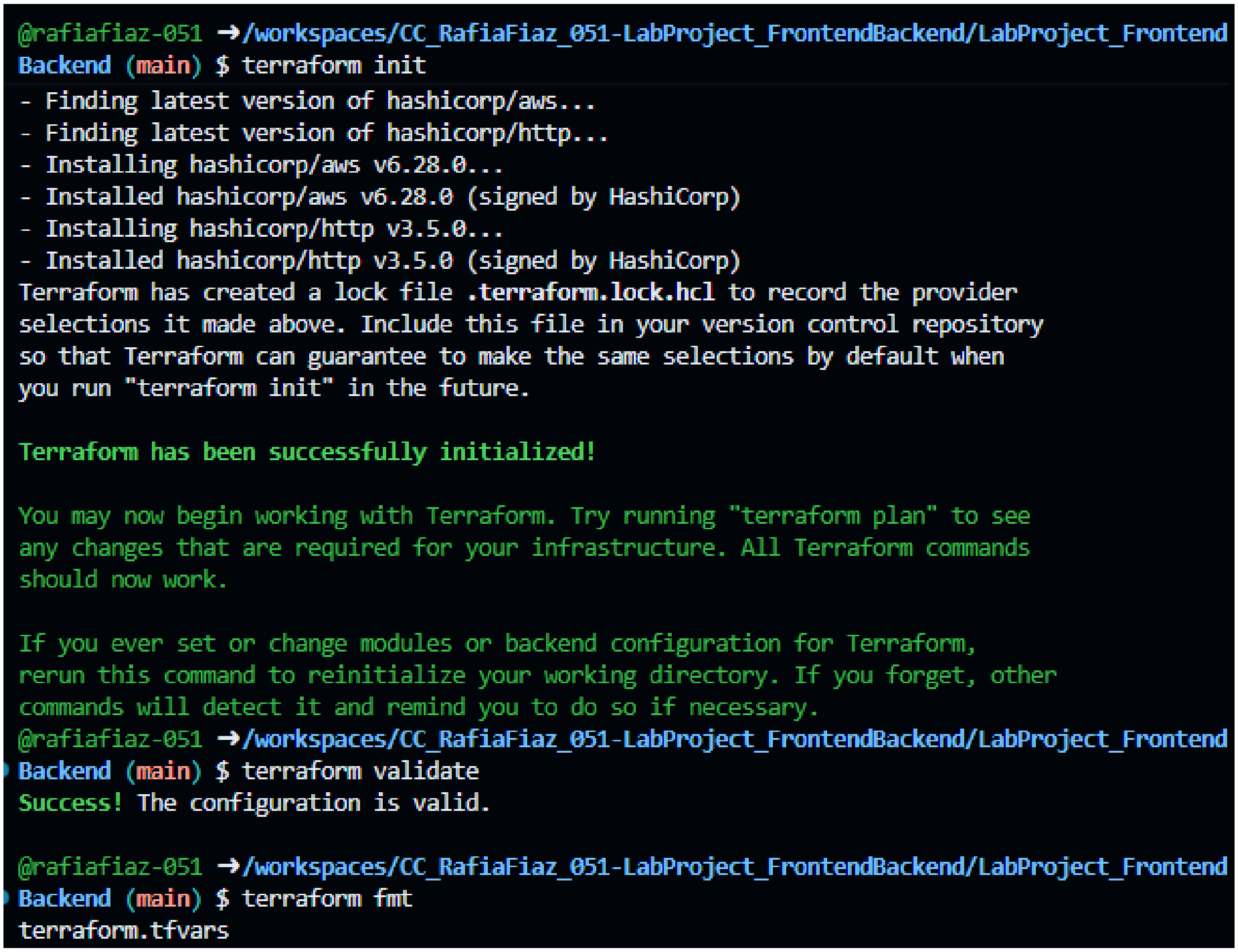
```
#Backend EC2
resource "aws_instance" "backend" {
  count                  = 3
  ami                    = data.aws_ami.amazon_linux.id
  instance_type          = var.instance_type
  subnet_id              = module.subnet.subnet_id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  key_name               = aws_key_pair.key.key_name

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

3.5 outputs.tf



```
  @rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
● Backend (main) $ cat outputs.tf
  output "frontend_public_ip" {
    value = aws_instance.frontend.public_ip
  }

  output "backend_public_ips" {
    value = [for b in aws_instance.backend : b.public_ip]
  }

  output "backend_private_ips" {
    value = [for b in aws_instance.backend : b.private_ip]
  }
```

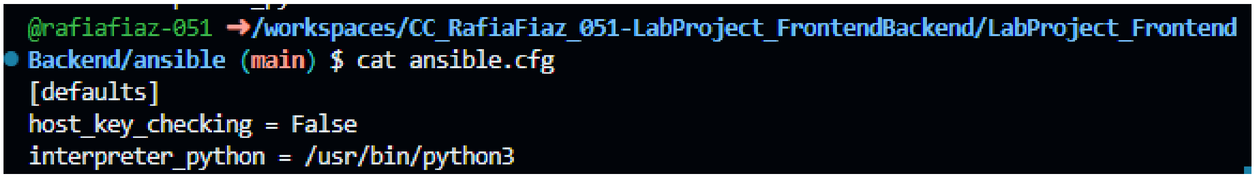
3.6 Terraform format + validate



```
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend (main) $ terraform init
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (signed by HashiCorp)
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend (main) $ terraform validate
Success! The configuration is valid.

@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend (main) $ terraform fmt
terraform.tfvars
```

PHASE 4 — ANSIBLE CONFIGURATION

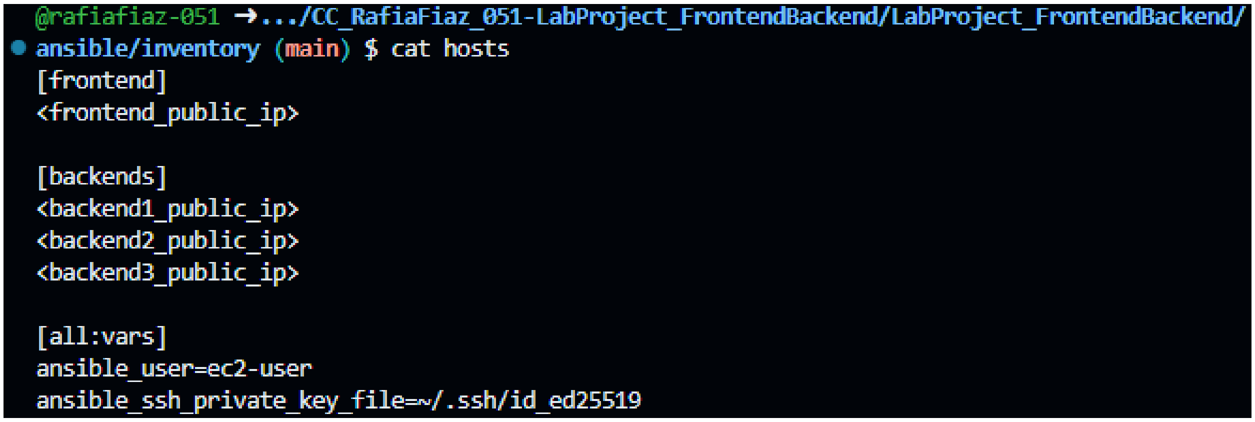
4.1 ansible/ansible.cfg



```
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend/ansible (main) $ cat ansible.cfg
[defaults]
host_key_checking = False
interpreter_python = /usr/bin/python3
```

4.2 Inventory (STATIC)

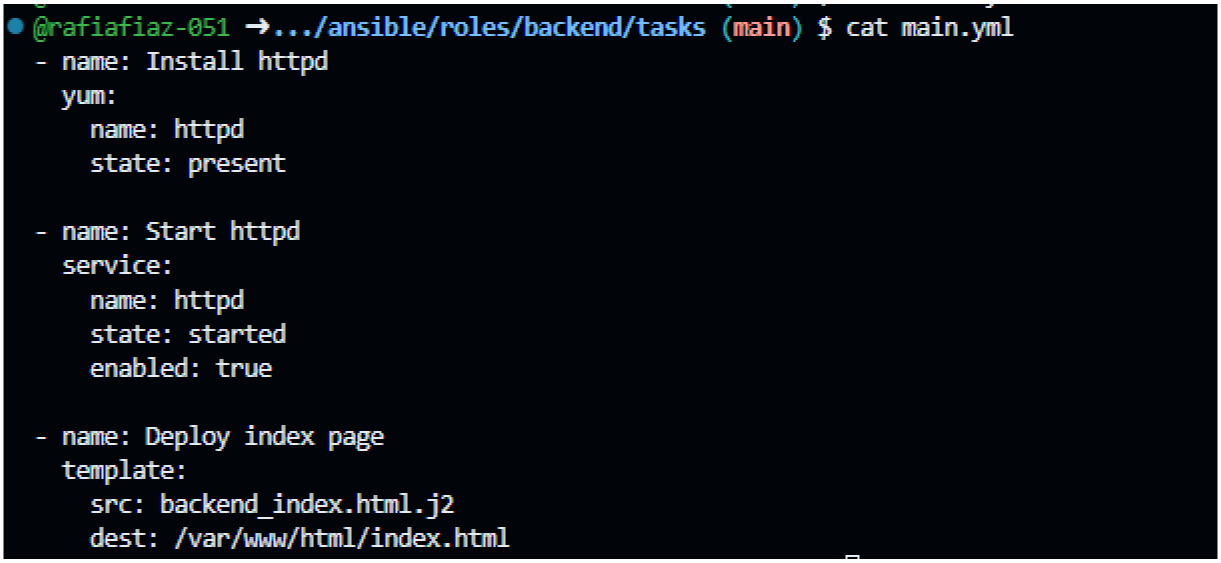
File: ansible/inventory/hosts



```
@rafiafiaz-051 ➜.../CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_FrontendBackend/
ansible/inventory (main) $ cat hosts
[frontend]
<frontend_public_ip>

[backends]
<backend1_public_ip>
<backend2_public_ip>
<backend3_public_ip>

[all:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
```

4.3 Backend Role — Apache install

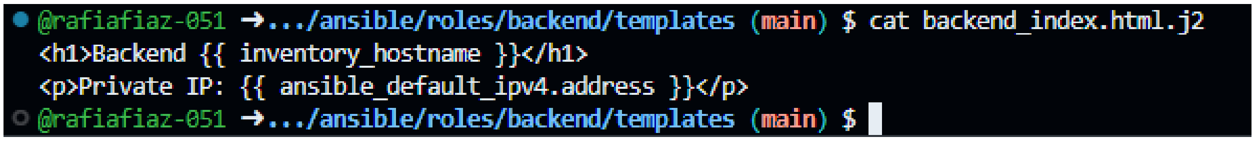
File: ansible/roles/backend/tasks/main.yml



```
@rafiafiaz-051 ➜.../ansible/roles/backend/tasks (main) $ cat main.yml
- name: Install httpd
  yum:
    name: httpd
    state: present

- name: Start httpd
  service:
    name: httpd
    state: started
    enabled: true

- name: Deploy index page
  template:
    src: backend_index.html.j2
    dest: /var/www/html/index.html
```

4.4 Backend HTML Template

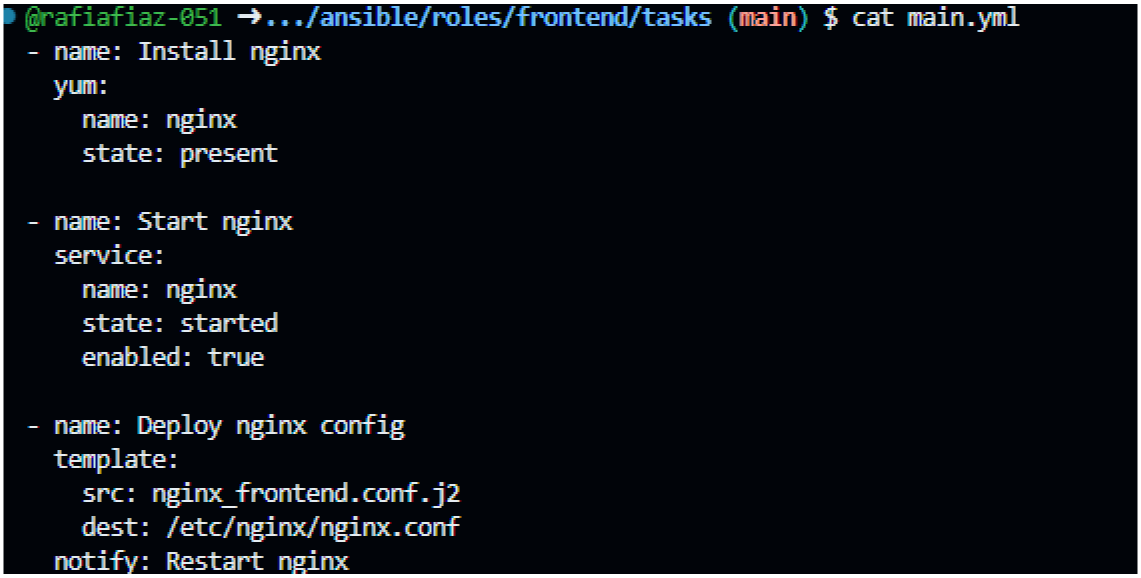
File: ansible/roles/backend/templates/backend_index.html.j2



```
@rafiafiaz-051 ➜.../ansible/roles/backend/templates (main) $ cat backend_index.html.j2
<h1>Backend {{ inventory_hostname }}</h1>
<p>Private IP: {{ ansible_default_ipv4.address }}</p>
@rafiafiaz-051 ➜.../ansible/roles/backend/templates (main) $
```

4.5 Frontend Role — Nginx

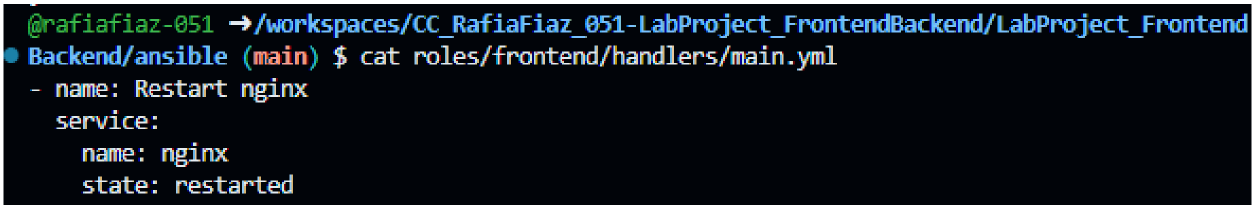
File: ansible/roles/frontend/tasks/main.yml



```
@rafiafiaz-051 ➜.../ansible/roles/frontend/tasks (main) $ cat main.yml
- name: Install nginx
  yum:
    name: nginx
    state: present

- name: Start nginx
  service:
    name: nginx
    state: started
    enabled: true

- name: Deploy nginx config
  template:
    src: nginx_frontend.conf.j2
    dest: /etc/nginx/nginx.conf
  notify: Restart nginx
```

4.6 Frontend Handler

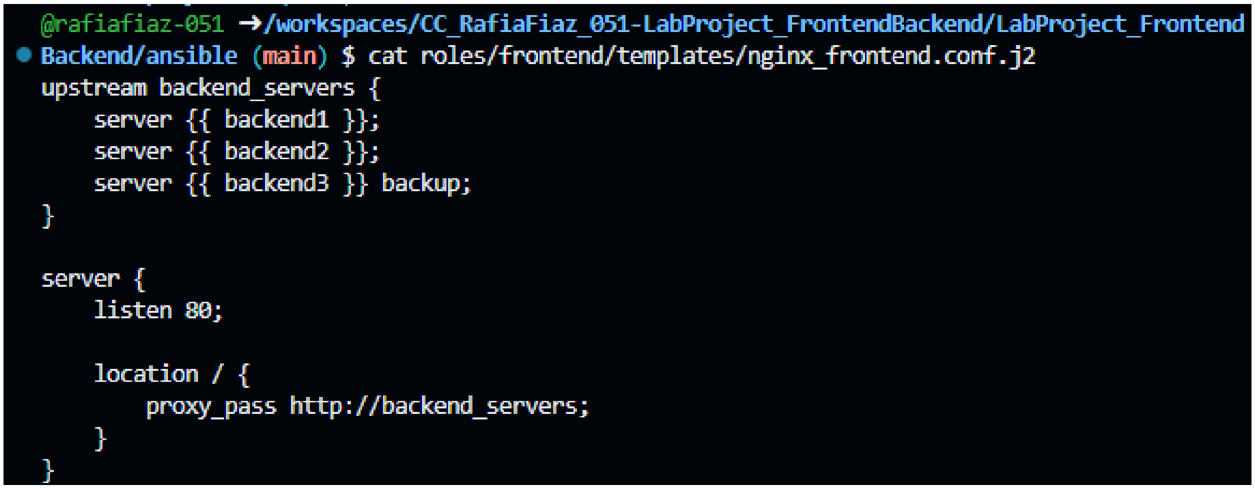
File: ansible/roles/frontend/handlers/main.yml



```
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend/ansible (main) $ cat roles/frontend/handlers/main.yml
- name: Restart nginx
  service:
    name: nginx
    state: restarted
```

4.7 Nginx Template (Load Balancer)

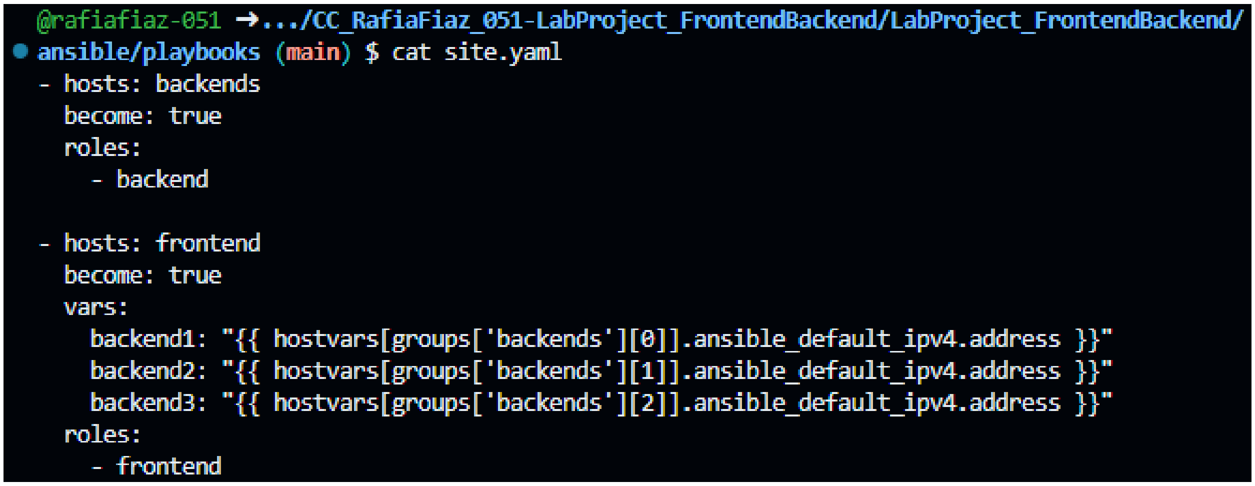
File: ansible/roles/frontend/templates/nginx_frontend.conf.j2



```
@rafiafiaz-051 ➜/workspaces/CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_Frontend
Backend/ansible (main) $ cat roles/frontend/templates/nginx_frontend.conf.j2
upstream backend_servers {
    server {{ backend1 }};
    server {{ backend2 }};
    server {{ backend3 }} backup;
}

server {
    listen 80;

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

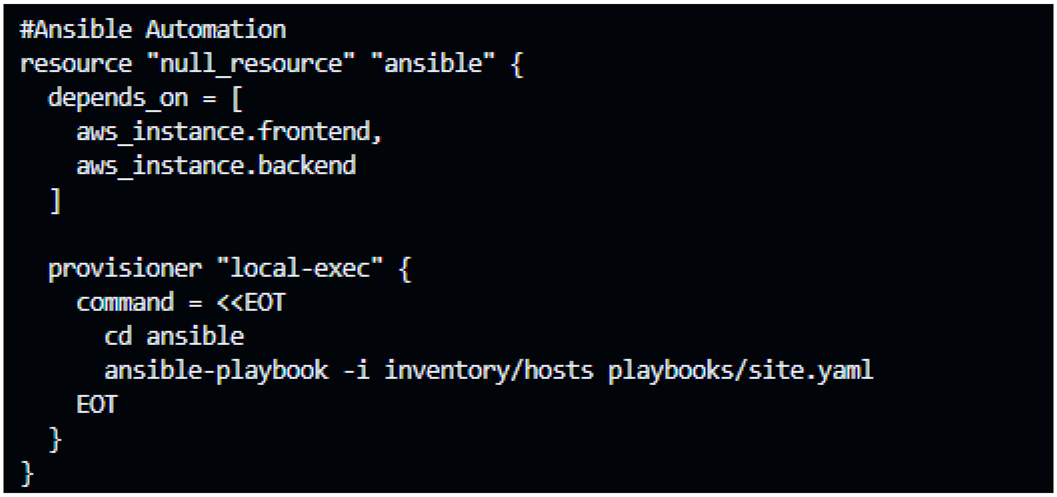
4.8 Main Playbook

File: ansible/playbooks/site.yaml



```
 @rafiafiaz-051 ➜.../CC_RafiaFiaz_051-LabProject_FrontendBackend/LabProject_FrontendBackend/
● ansible/playbooks (main) $ cat site.yaml
  - hosts: backends
    become: true
    roles:
      - backend

  - hosts: frontend
    become: true
    vars:
      backend1: "{{ hostvars[groups['backends'][0]].ansible_default_ipv4.address }}"
      backend2: "{{ hostvars[groups['backends'][1]].ansible_default_ipv4.address }}"
      backend3: "{{ hostvars[groups['backends'][2]].ansible_default_ipv4.address }}"
    roles:
      - frontend
```

PHASE 5 — TERRAFORM + ANSIBLE AUTOMATION



```
#Ansible Automation
resource "null_resource" "ansible" {
  depends_on = [
    aws_instance.frontend,
    aws_instance.backend
  ]

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