CLOUD COMPUTING
LAB PROJECT

Fatima Jinnah Women University

SUBMITTED TO

ENGR. SHOAIB

SUBMITTED BY

HAMNA MAHMOOD

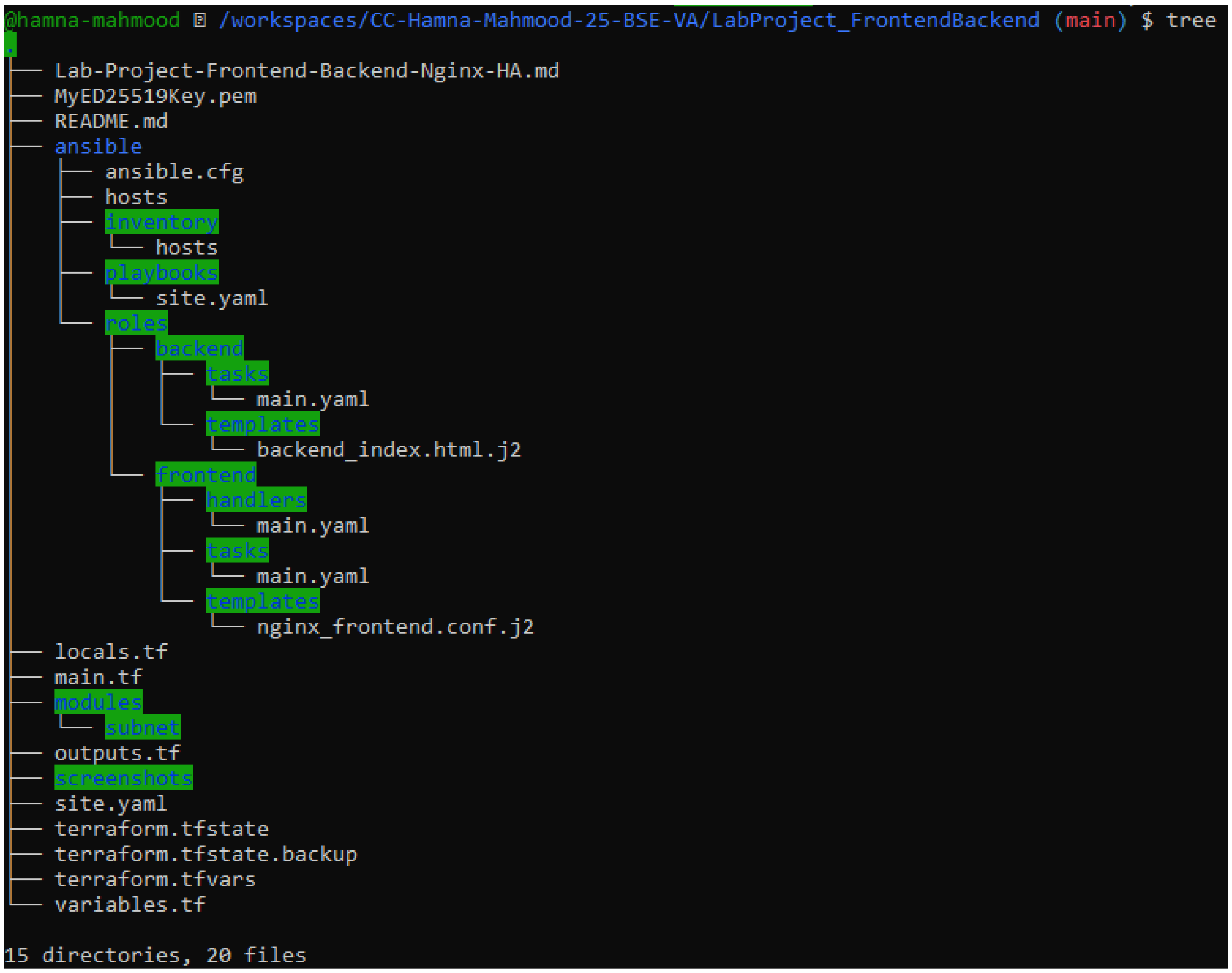
2023-BSE-025

BSE V-A

Lab Project

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

- **Required folder structure**

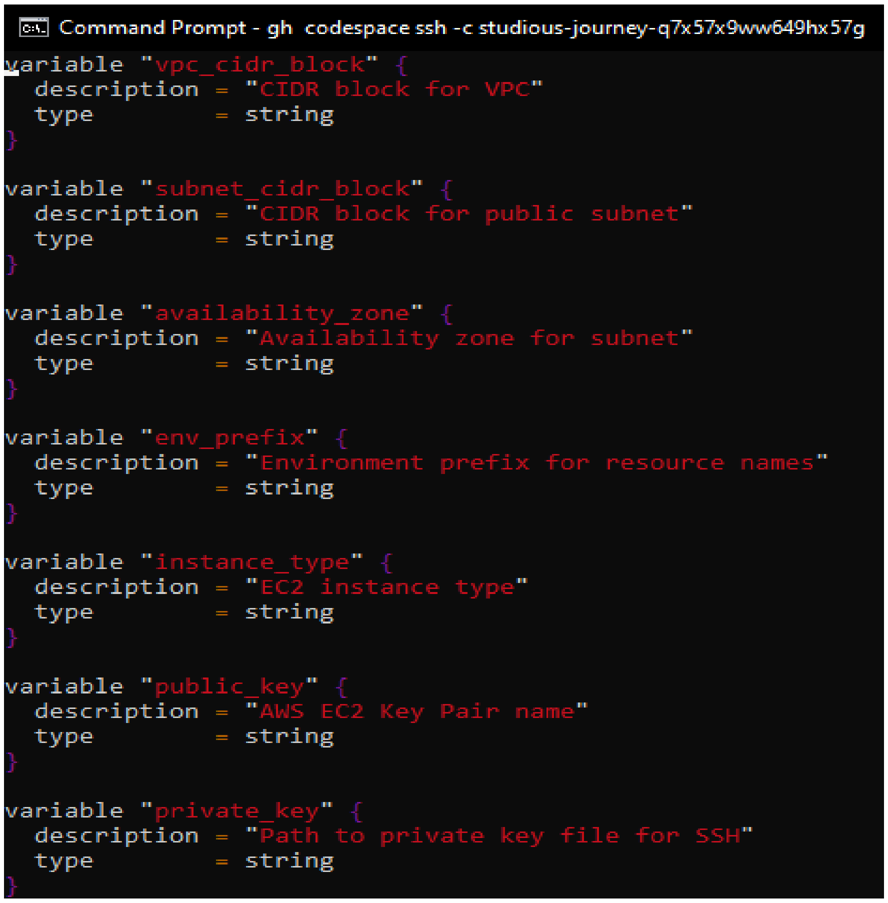


```
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA/LabProject_FrontendBackend (main) $ tree
.
├── Lab-Project-Frontend-Backend-Nginx-HA.md
├── MyED25519Key.pem
├── README.md
├── ansible
│   ├── ansible.cfg
│   ├── hosts
│   ├── inventory
│   │   └── hosts
│   ├── playbooks
│   │   └── site.yaml
│   └── roles
│       ├── backend
│       │   ├── tasks
│       │   │   └── main.yaml
│       │   └── templates
│       │       └── backend_index.html.j2
│       └── frontend
│           ├── handlers
│           │   └── main.yaml
│           ├── tasks
│           │   └── main.yaml
│           └── templates
│               └── nginx_frontend.conf.j2
├── locals.tf
├── main.tf
├── modules
│   └── subnet
├── outputs.tf
├── screenshots
├── site.yaml
├── terraform.tfstate
├── terraform.tfstate.backup
├── terraform.tfvars
└── variables.tf

15 directories, 20 files
```

- **Terraform code (*.tf, modules)**

1. **Terraform – Networking & Common Settings (Architecture Definition)**

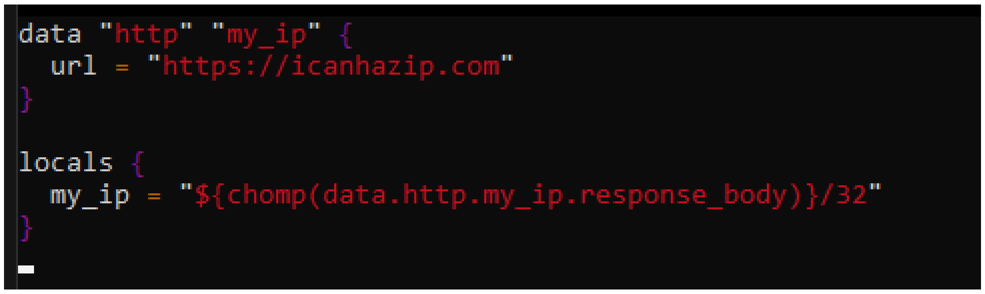
Variables setup (variables.tf):



Command Prompt - gh codespace ssh -c studious-journey-q7x57x9ww649hx57g

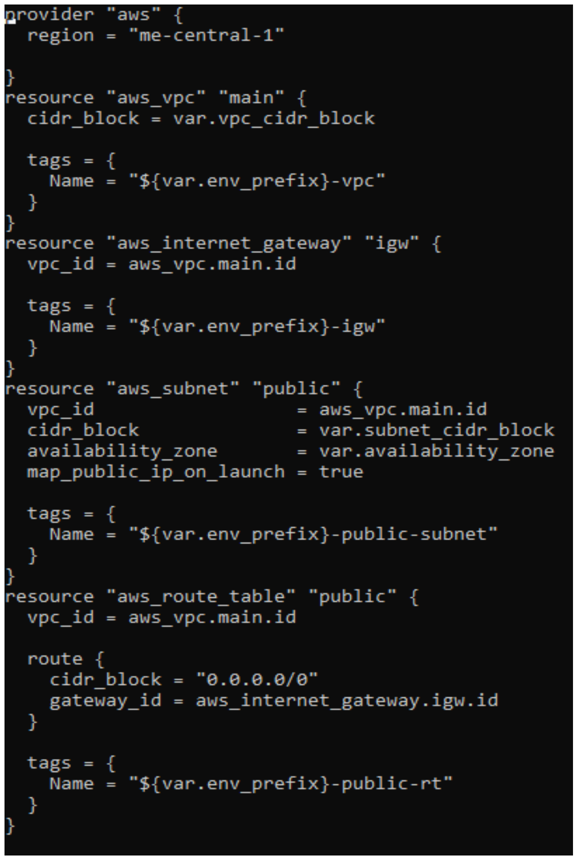
```
variable "vpc_cidr_block" {
  description = "CIDR block for VPC"
  type        = string
}

variable "subnet_cidr_block" {
  description = "CIDR block for public subnet"
  type        = string
}

variable "availability_zone" {
  description = "Availability zone for subnet"
  type        = string
}

variable "env_prefix" {
  description = "Environment prefix for resource names"
  type        = string
}

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

variable "public_key" {
  description = "AWS EC2 Key Pair name"
  type        = string
}

variable "private_key" {
  description = "Path to private key file for SSH"
  type        = string
}
```

Locals (locals.tf):

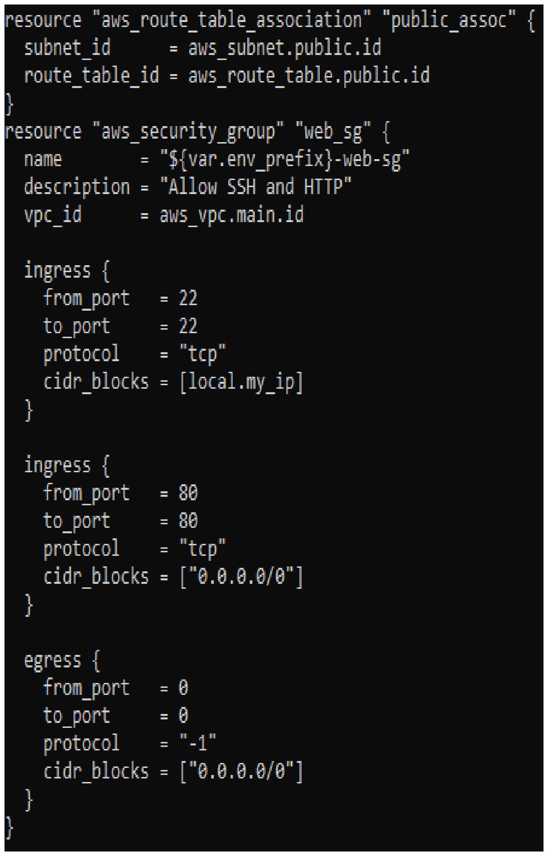


```
data "http" "my_ip" {
  url = "https://icanhazip.com"
}

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

Main.tf



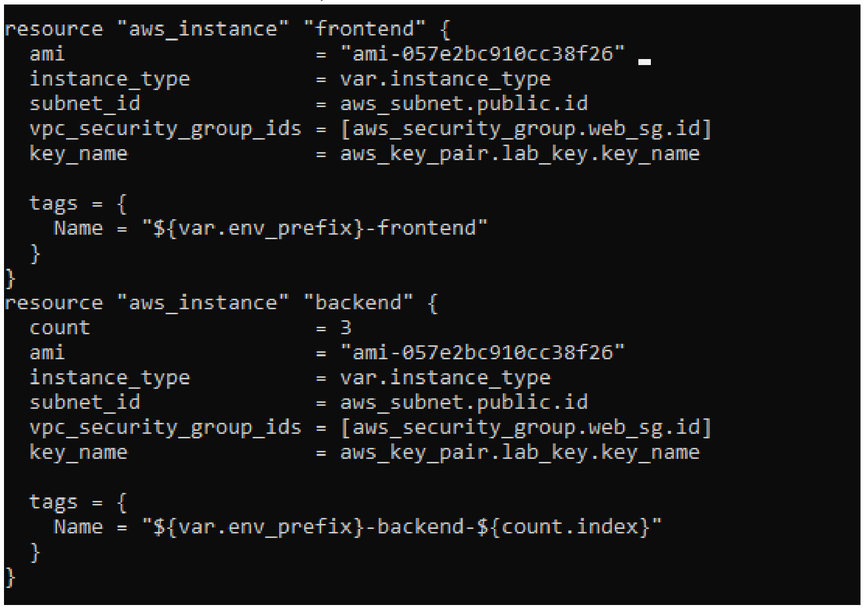
```
provider "aws" {
  region = "me-central-1"

}
resource "aws_vpc" "main" {
  cidr_block = var.vpc_cidr_block

  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name = "${var.env_prefix}-igw"
  }
}
resource "aws_subnet" "public" {
  vpc_id                  = aws_vpc.main.id
  cidr_block              = var.subnet_cidr_block
  availability_zone       = var.availability_zone
  map_public_ip_on_launch = true

  tags = {
    Name = "${var.env_prefix}-public-subnet"
  }
}
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 = {
    Name = "${var.env_prefix}-public-rt"
  }
}
```



```
resource "aws_route_table_association" "public_assoc" {
  subnet_id      = aws_subnet.public.id
  route_table_id = aws_route_table.public.id
}
resource "aws_security_group" "web_sg" {
  name        = "${var.env_prefix}-web-sg"
  description = "Allow SSH and HTTP"
  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"]
  }
}
```

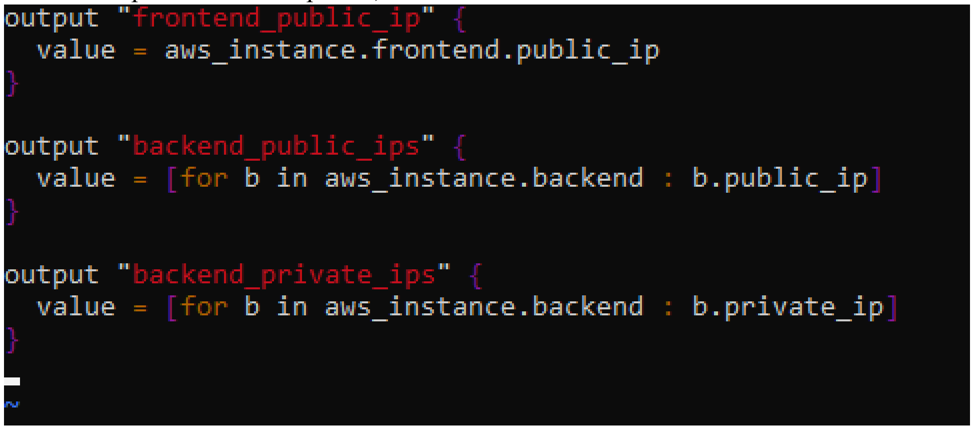
2. Terraform – Frontend & Backend EC2 Instances

EC2 frontend &backend instances in main.tf,



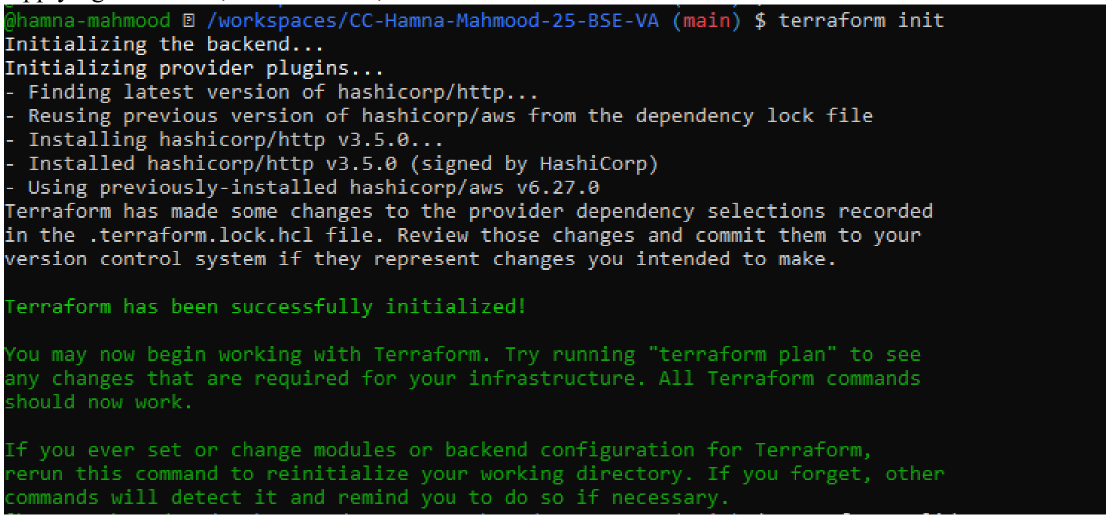
```
resource "aws_instance" "frontend" {
  ami                    = "ami-057e2bc910cc38f26"
  instance_type          = var.instance_type
  subnet_id              = aws_subnet.public.id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  key_name               = aws_key_pair.lab_key.key_name

  tags = {
    Name = "${var.env_prefix}-frontend"
  }
}
resource "aws_instance" "backend" {
  count                  = 3
  ami                    = "ami-057e2bc910cc38f26"
  instance_type          = var.instance_type
  subnet_id              = aws_subnet.public.id
  vpc_security_group_ids = [aws_security_group.web_sg.id]
  key_name               = aws_key_pair.lab_key.key_name

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

Private and public IPs in outputs.tf,



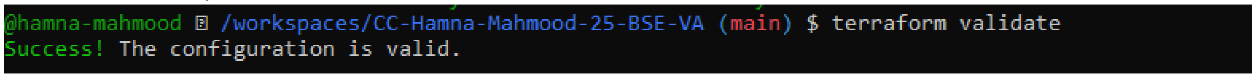
```
output "frontend_public_ip" {
  value = aws_instance.frontend.public_ip
}

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]
}

~
```

Applying Terraform, terraform init,



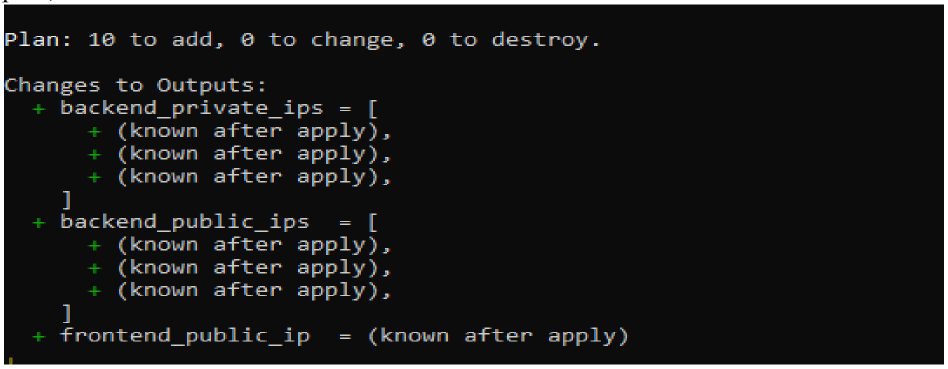
```
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/http...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Using previously-installed hashicorp/aws v6.27.0
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
```

Terraform validate,



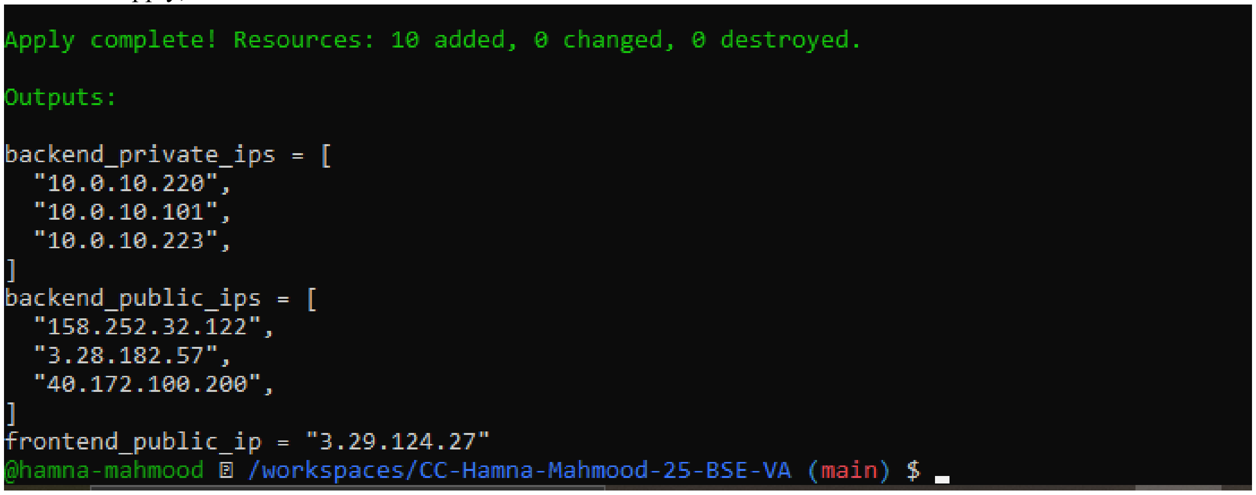
```
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $ terraform validate
Success! The configuration is valid.
```

Terraform plan,



```
Plan: 10 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + backend_private_ips = [
      + (known after apply),
      + (known after apply),
      + (known after apply),
    ]
  + backend_public_ips  = [
      + (known after apply),
      + (known after apply),
      + (known after apply),
    ]
  + frontend_public_ip  = (known after apply)
```

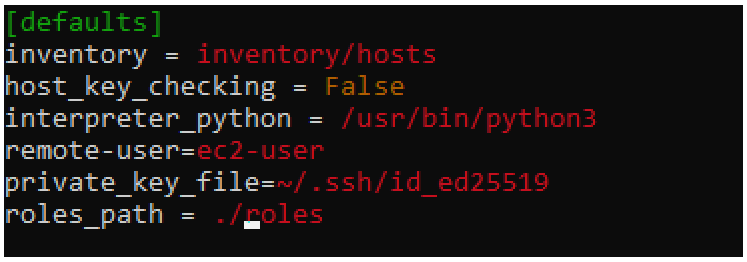
Terraform apply,



```
Apply complete! Resources: 10 added, 0 changed, 0 destroyed.

Outputs:

backend_private_ips = [
  "10.0.10.220",
  "10.0.10.101",
  "10.0.10.223",
]
backend_public_ips = [
  "158.252.32.122",
  "3.28.182.57",
  "40.172.100.200",
]
frontend_public_ip = "3.29.124.27"
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA (main) $
```

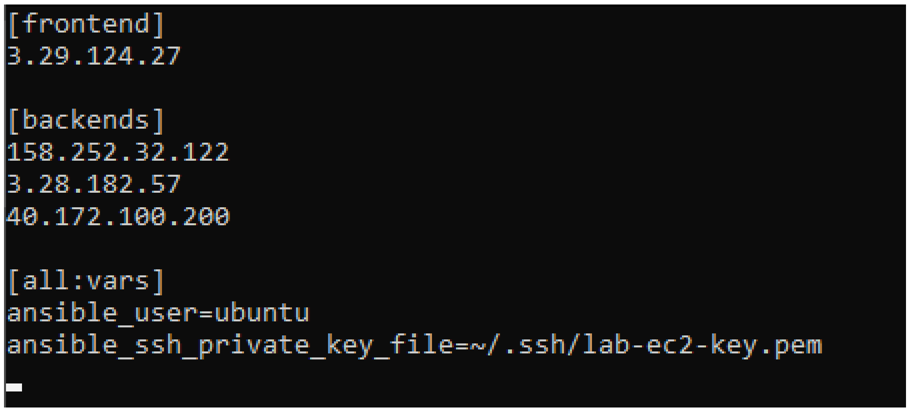
3 Ansible- Global Config & Inventory

ansible/ansible.cfg,



```
[defaults]
inventory = inventory/hosts
host_key_checking = False
interpreter_python = /usr/bin/python3
remote-user=ec2-user
private_key_file=~/.ssh/id_ed25519
roles_path = ./roles
```

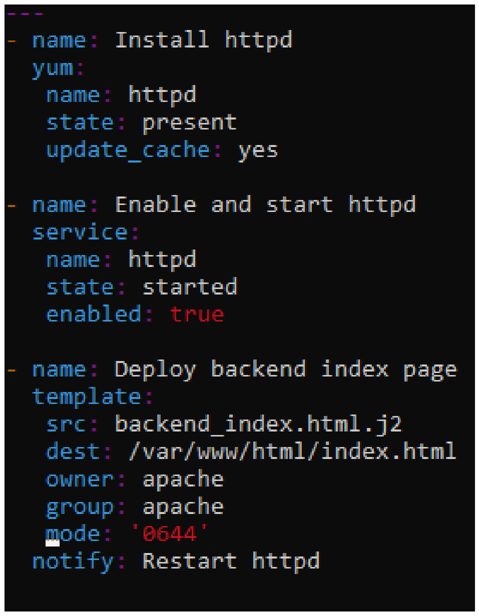
ansible/inventory/ hosts file,



```
[frontend]
3.29.124.27

[backends]
158.252.32.122
3.28.182.57
40.172.100.200

[all:vars]
ansible_user=ubuntu
ansible_ssh_private_key_file=~/.ssh/lab-ec2-key.pem
```

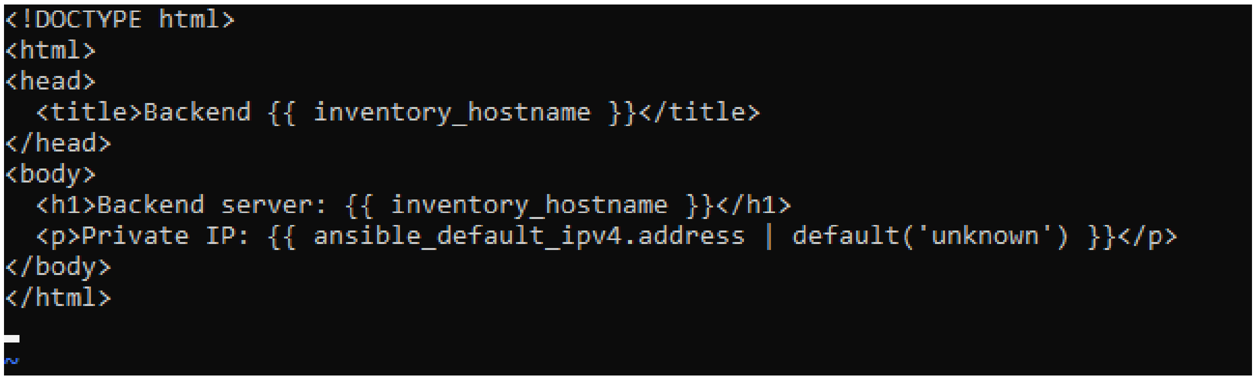
4. Ansible Roles – Backend HTTP Role

ansible/roles/backend/tasks/main.yaml,



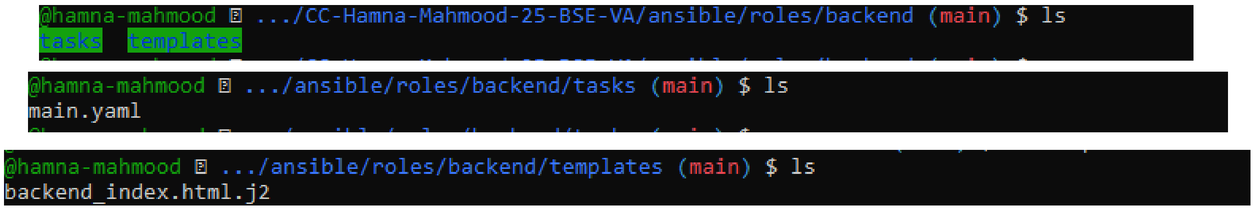
```
---
- name: Install httpd
  yum:
   name: httpd
   state: present
   update_cache: yes

- name: Enable and start httpd
  service:
   name: httpd
   state: started
   enabled: true

- name: Deploy backend index page
  template:
   src: backend_index.html.j2
   dest: /var/www/html/index.html
   owner: apache
   group: apache
   mode: '0644'
  notify: Restart httpd
```

ansible/roles/backend/tasks/ backend_index.html.j2



```
<!DOCTYPE html>
<html>
<head>
  <title>Backend {{ inventory_hostname }}</title>
</head>
<body>
  <h1>Backend server: {{ inventory_hostname }}</h1>
  <p>Private IP: {{ ansible_default_ipv4.address | default('unknown') }}</p>
</body>
</html>

~
```

Folder structure yet,



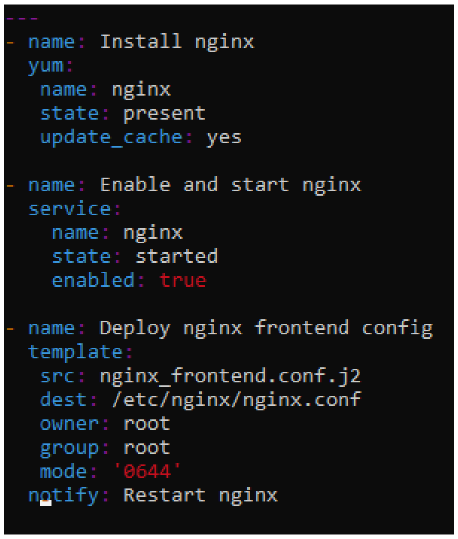
```
@hamna-mahmood ➜ .../CC-Hamna-Mahmood-25-BSE-VA/ansible/roles/backend (main) $ ls
tasks  templates
```

```
@hamna-mahmood ➜ .../ansible/roles/backend/tasks (main) $ ls
main.yaml
```

```
@hamna-mahmood ➜ .../ansible/roles/backend/templates (main) $ ls
backend_index.html.j2
```

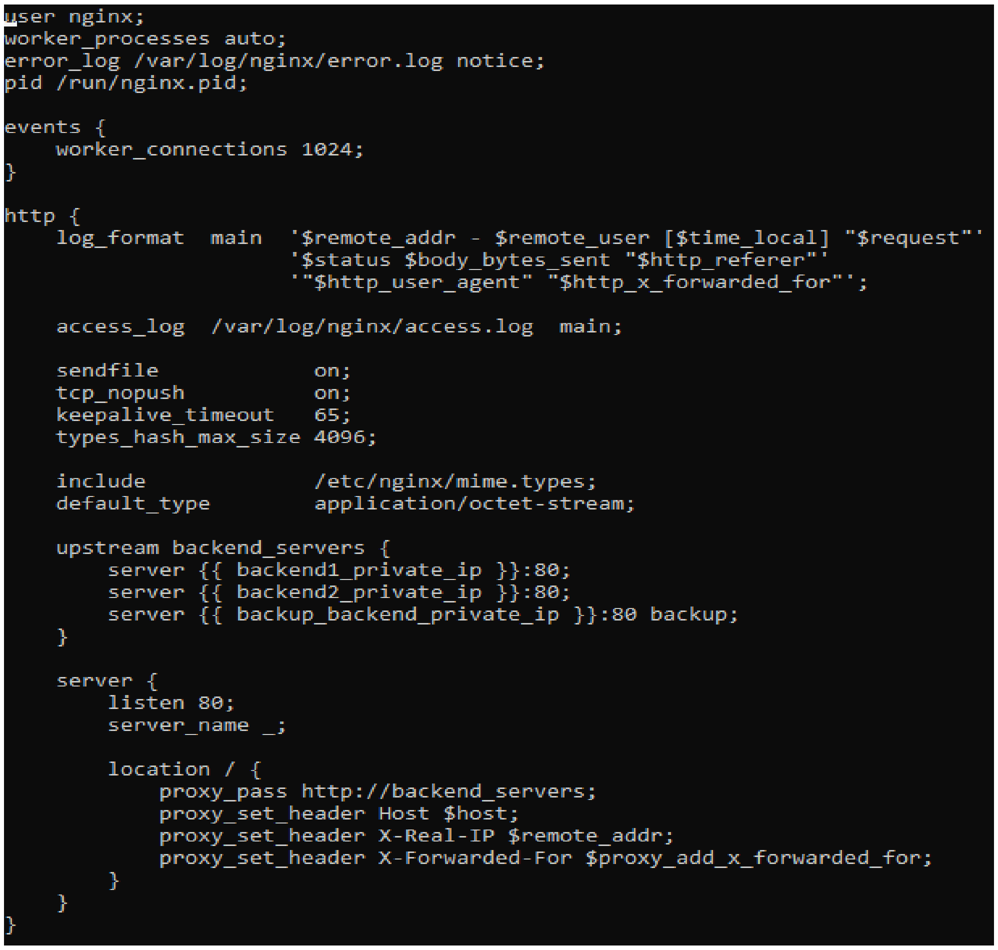
5. **Ansible Roles – Frontend Nginx Role (frontend/)**

ansible/roles/frontend/tasks/main.yaml,



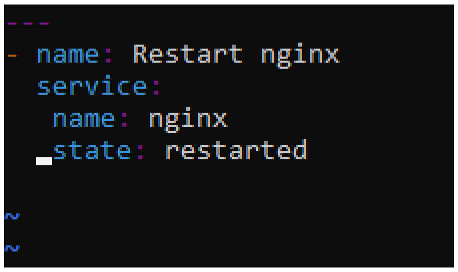
```yaml
---
- name: Install nginx
  yum:
   name: nginx
   state: present
   update_cache: yes

- name: Enable and start nginx
  service:
    name: nginx
    state: started
    enabled: true

- name: Deploy nginx frontend config
  template:
   src: nginx_frontend.conf.j2
   dest: /etc/nginx/nginx.conf
   owner: root
   group: root
   mode: '0644'
  notify: Restart nginx
```

ansible/roles/frontend/templates/ nginx_frontend.conf.j2,



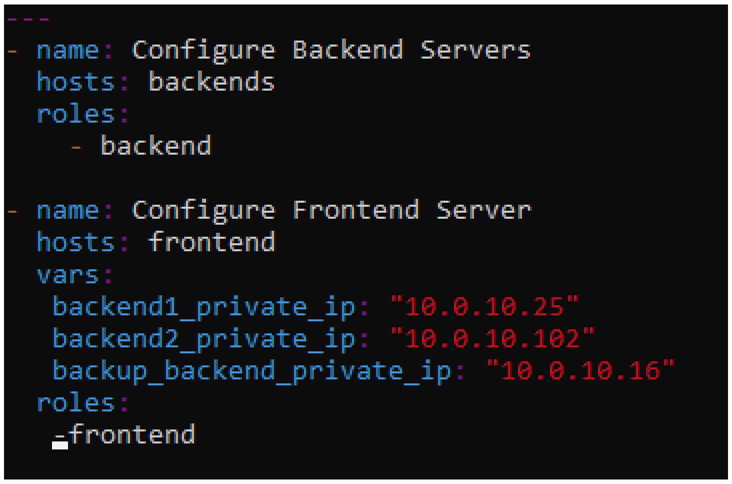
```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

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

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

    sendfile            on;
    tcp_nopush          on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

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

    upstream backend_servers {
        server {{ backend1_private_ip }}:80;
        server {{ backend2_private_ip }}:80;
        server {{ backup_backend_private_ip }}:80 backup;
    }

    server {
        listen 80;
        server_name _;

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

Ansible/roles/frontend/handlers/main.yaml,

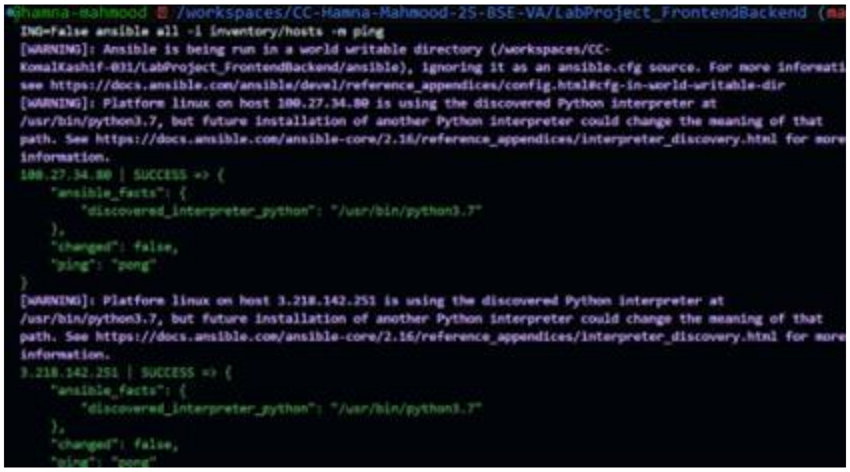


```
---
- name: Restart nginx
  service:
   name: nginx
   state: restarted

~
~
```

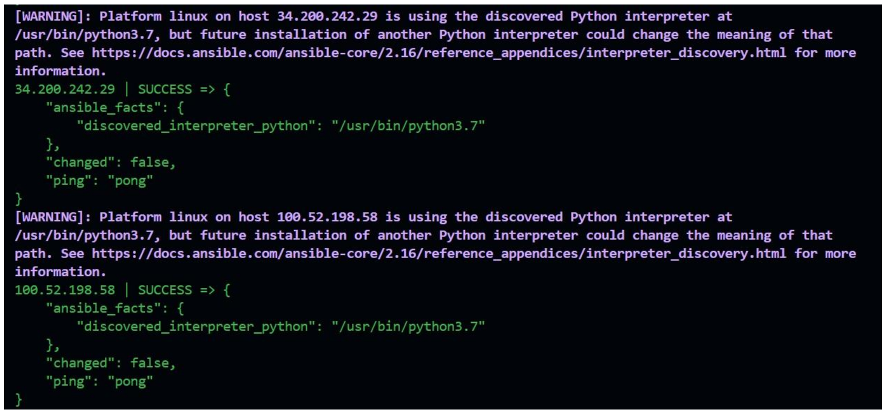
Adding IPs in ansible/playbooks/site.yaml,



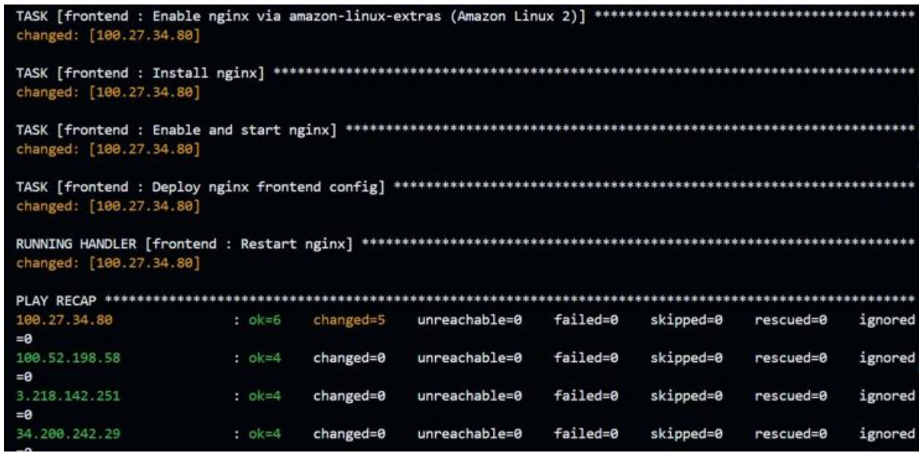
```
---
- name: Configure Backend Servers
  hosts: backends
  roles:
    - backend

- name: Configure Frontend Server
  hosts: frontend
  vars:
   backend1_private_ip: "10.0.10.25"
   backend2_private_ip: "10.0.10.102"
   backup_backend_private_ip: "10.0.10.16"
  roles:
   -frontend
```



```
@hamna-mahmood /workspaces/CC-Hamna-Mahmood-25-BSE-VA/LabProject_FrontendBackend (ma
 ING=False ansible all -i inventory/hosts -m ping
 [WARNING]: Ansible is being run in a world writable directory (/workspaces/CC-
 KomalKashif-031/LabProject_FrontendBackend/ansible), ignoring it as an ansible.cfg source. For more informati
 see https://docs.ansible.com/ansible/devel/reference_appendices/config.html#cfg-in-world-writable-dir
 [WARNING]: Platform linux on host 100.27.34.80 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.
 100.27.34.80 | SUCCESS => {
     "ansible_facts": {
         "discovered_interpreter_python": "/usr/bin/python3.7"
     },
     "changed": false,
     "ping": "pong"
 }
 [WARNING]: Platform linux on host 3.218.142.251 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.
 3.218.142.251 | SUCCESS => {
     "ansible_facts": {
         "discovered_interpreter_python": "/usr/bin/python3.7"
     },
     "changed": false,
     "ping": "pong"
```



```
[WARNING]: Platform linux on host 34.200.242.29 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.
34.200.242.29 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
[WARNING]: Platform linux on host 100.52.198.58 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.
100.52.198.58 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.7"
    },
    "changed": false,
    "ping": "pong"
}
```

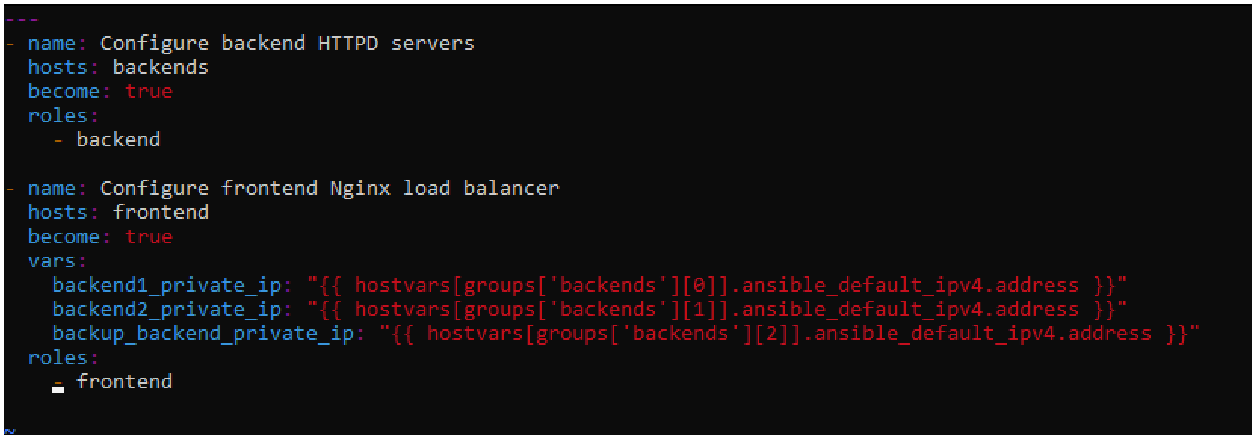


```
TASK [frontend : Enable nginx via amazon-linux-extras (Amazon Linux 2)] ****************************************
changed: [100.27.34.80]

TASK [frontend : Install nginx] ************************************************************************
changed: [100.27.34.80]

TASK [frontend : Enable and start nginx] ***************************************************************
changed: [100.27.34.80]

TASK [frontend : Deploy nginx frontend config] *********************************************************
changed: [100.27.34.80]

RUNNING HANDLER [frontend : Restart nginx] *************************************************************
changed: [100.27.34.80]

PLAY RECAP *********************************************************************************************
100.27.34.80               : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0
100.52.198.58              : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0
3.218.142.251              : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored
=0
34.200.242.29              : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored
```

6. Ansible Main Playbook Using Roles

Ansible/playbooks/site.yaml,

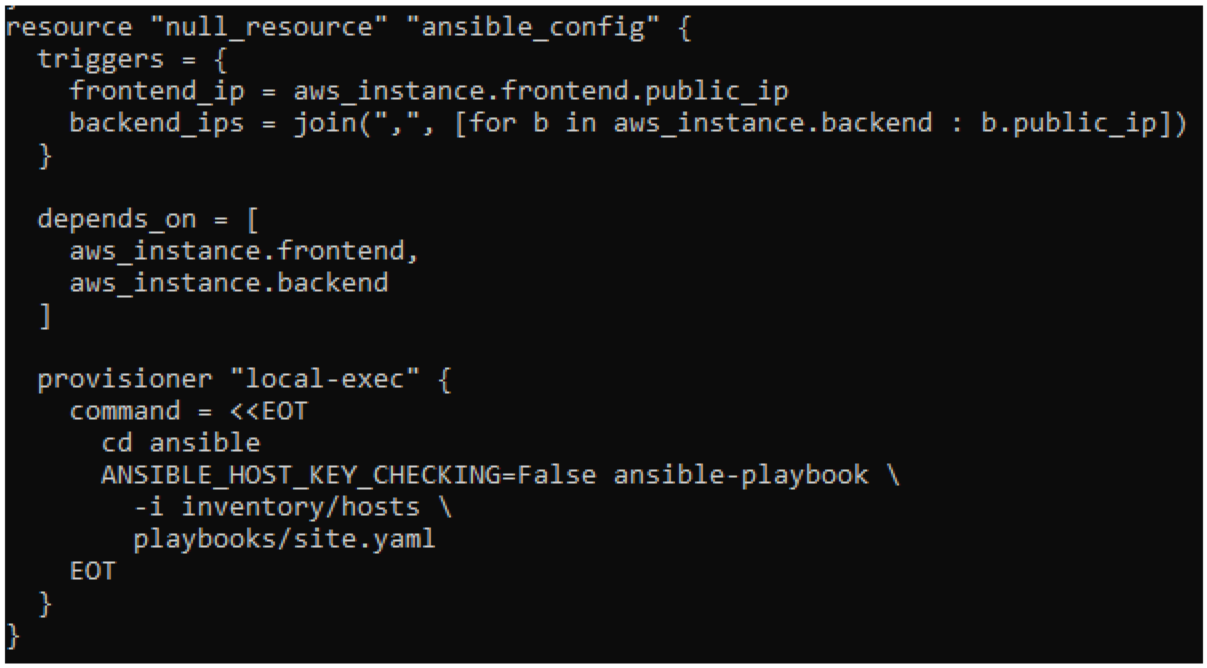
Using dynamic hostvars,



```yaml
---
- name: Configure backend HTTPD servers
  hosts: backends
  become: true
  roles:
    - backend

- name: Configure frontend Nginx load balancer
  hosts: frontend
  become: true
  vars:
    backend1_private_ip: "{{ hostvars[groups['backends'][0]].ansible_default_ipv4.address }}"
    backend2_private_ip: "{{ hostvars[groups['backends'][1]].ansible_default_ipv4.address }}"
    backup_backend_private_ip: "{{ hostvars[groups['backends'][2]].ansible_default_ipv4.address }}"
  roles:
    - frontend

~
```

7. Terraform–Ansible Integration (Automation)

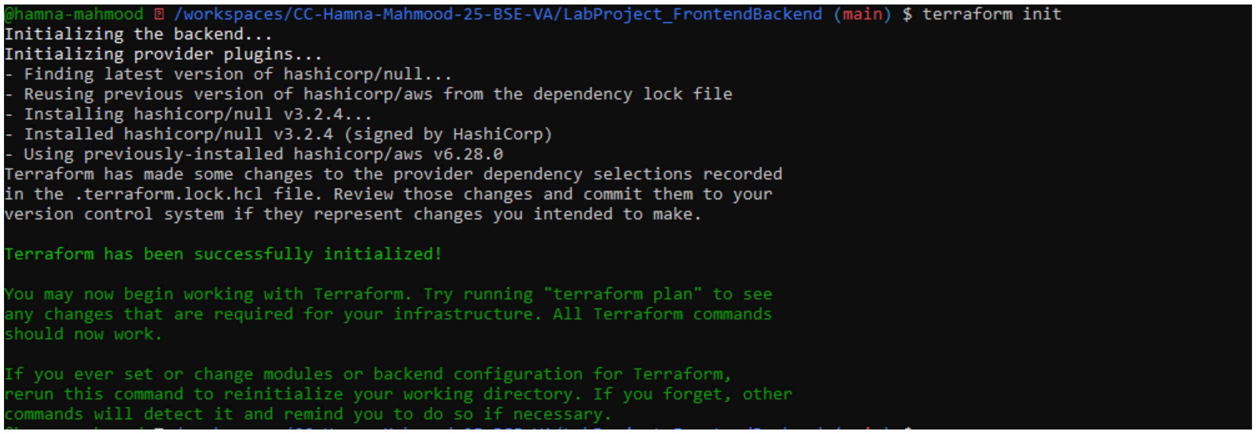
Adding null_resource,



```
resource "null_resource" "ansible_config" {
  triggers = {
    frontend_ip = aws_instance.frontend.public_ip
    backend_ips = join(",", [for b in aws_instance.backend : b.public_ip])
  }

  depends_on = [
    aws_instance.frontend,
    aws_instance.backend
  ]

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

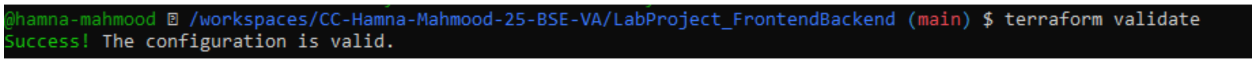
Now, running Terraform commands again,

Terraform init,



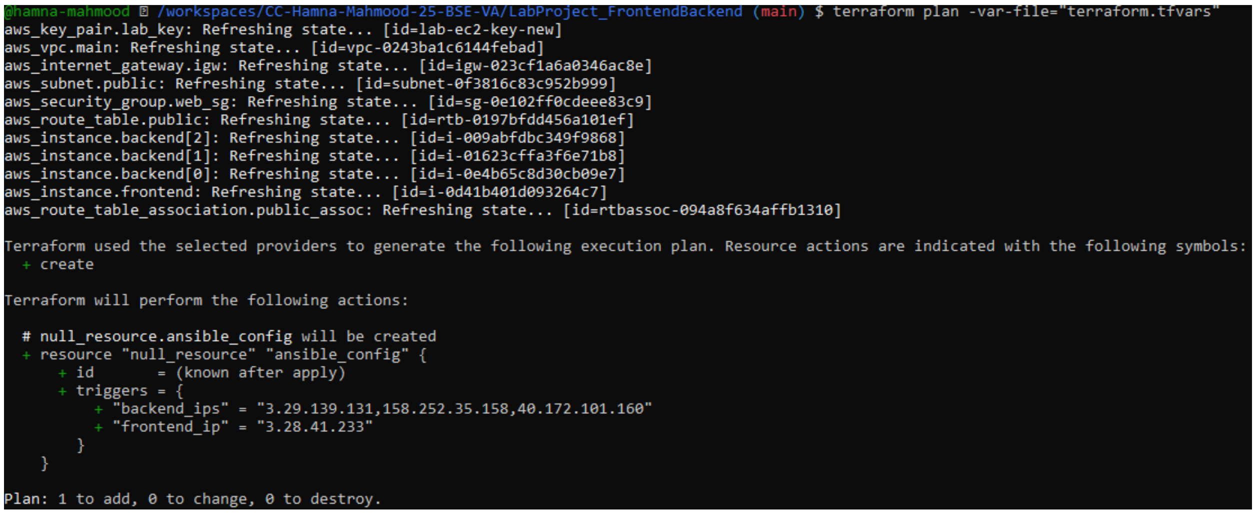
```
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA/LabProject_FrontendBackend (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/null...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Installing hashicorp/null v3.2.4...
- Installed hashicorp/null v3.2.4 (signed by HashiCorp)
- Using previously-installed hashicorp/aws v6.28.0
Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

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.
```

Terraform validate,



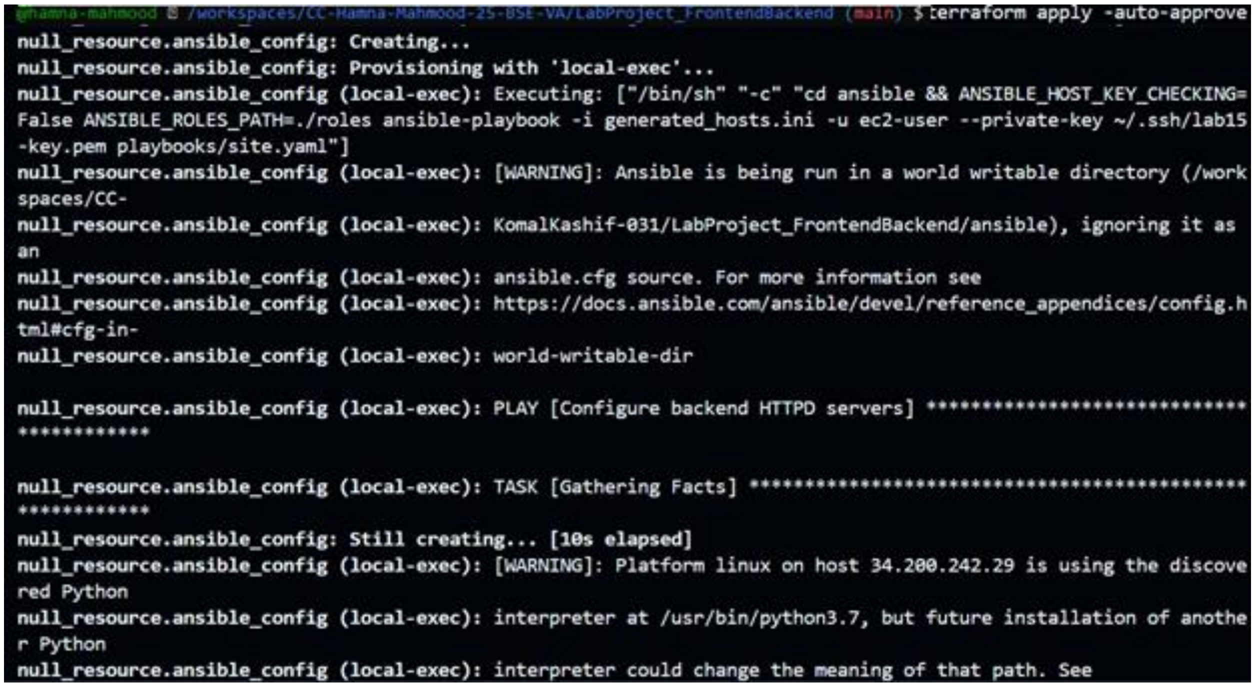
```
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA/LabProject_FrontendBackend (main) $ terraform validate
Success! The configuration is valid.
```

Terraform plan,

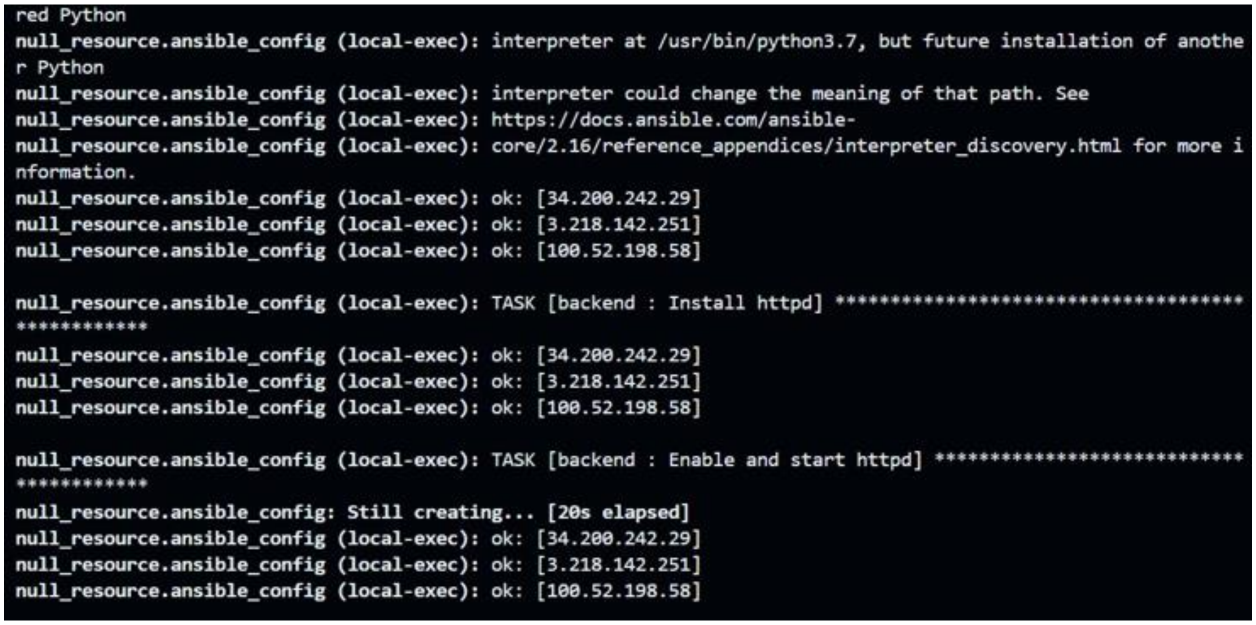


```
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA/LabProject_FrontendBackend (main) $ terraform plan -var-file="terraform.tfvars"
aws_key_pair.lab_key: Refreshing state... [id=lab-ec2-key-new]
aws_vpc.main: Refreshing state... [id=vpc-0243ba1c6144febad]
aws_internet_gateway.igw: Refreshing state... [id=igw-023cf1a6a0346ac8e]
aws_subnet.public: Refreshing state... [id=subnet-0f3816c83c952b999]
aws_security_group.web_sg: Refreshing state... [id=sg-0e102ff0cdeee83c9]
aws_route_table.public: Refreshing state... [id=rtb-0197bfdd456a101ef]
aws_instance.backend[2]: Refreshing state... [id=i-009abfdbc349f9868]
aws_instance.backend[1]: Refreshing state... [id=i-01623cffa3f6e71b8]
aws_instance.backend[0]: Refreshing state... [id=i-0e4b65c8d30cb09e7]
aws_instance.frontend: Refreshing state... [id=i-0d41b401d093264c7]
aws_route_table_association.public_assoc: Refreshing state... [id=rtbassoc-094a8f634affb1310]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # null_resource.ansible_config will be created
  + resource "null_resource" "ansible_config" {
      + id       = (known after apply)
      + triggers = {
          + "backend_ips" = "3.29.139.131,158.252.35.158,40.172.101.160"
          + "frontend_ip" = "3.28.41.233"
        }
    }

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

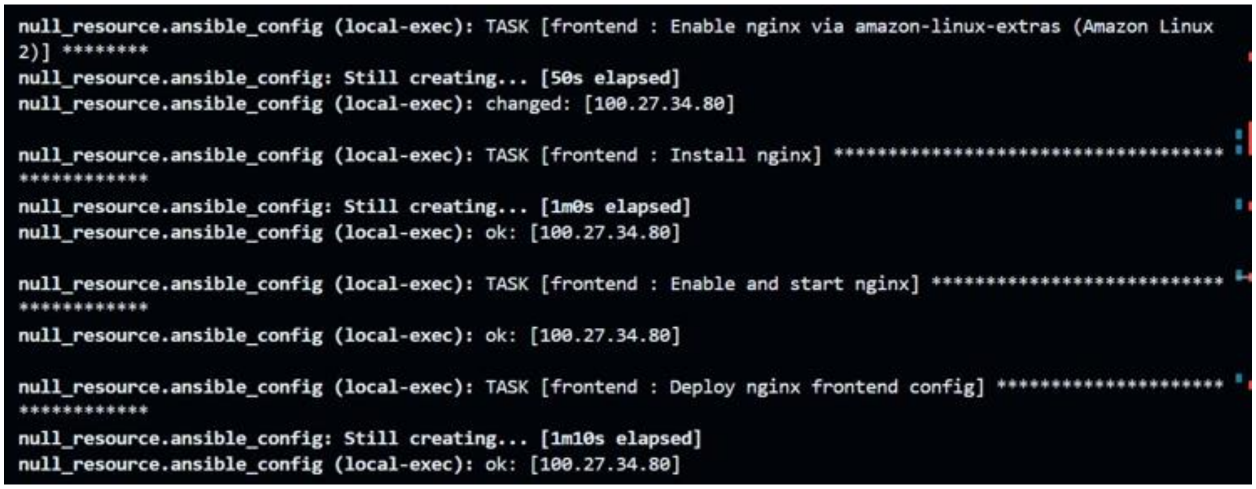
Terraform apply,



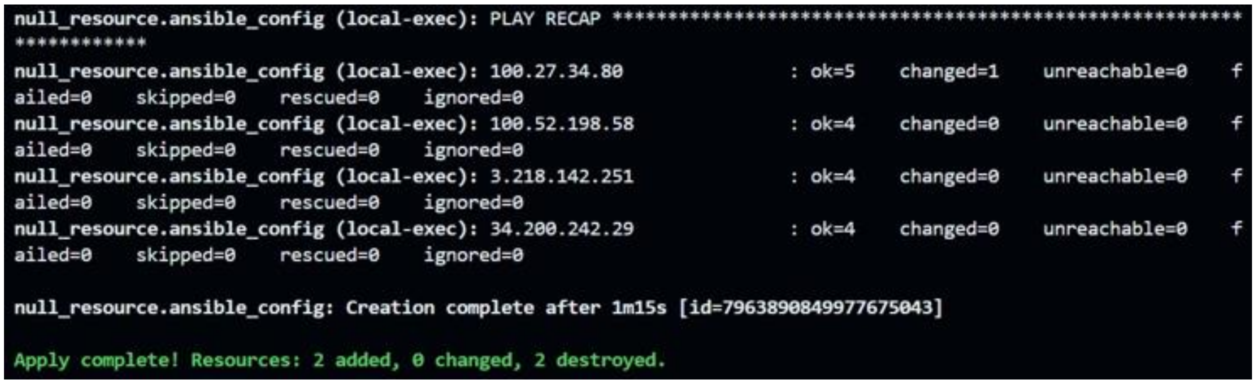
```
@hamna-mahmood ➜ /workspaces/CC-Hamna-Mahmood-25-BSE-VA/LabProject_FrontendBackend (main) $ terraform apply -auto-approve
null_resource.ansible_config: Creating...
null_resource.ansible_config: Provisioning with 'local-exec'...
null_resource.ansible_config (local-exec): Executing: ["/bin/sh" "-c" "cd ansible && ANSIBLE_HOST_KEY_CHECKING=
False ANSIBLE_ROLES_PATH=./roles ansible-playbook -i generated_hosts.ini -u ec2-user --private-key ~/.ssh/lab15
-key.pem playbooks/site.yaml"]
null_resource.ansible_config (local-exec): [WARNING]: Ansible is being run in a world writable directory (/work
spaces/CC-
null_resource.ansible_config (local-exec): KomalKashif-031/LabProject_FrontendBackend/ansible), ignoring it as
an
null_resource.ansible_config (local-exec): ansible.cfg source. For more information see
null_resource.ansible_config (local-exec): https://docs.ansible.com/ansible/devel/reference_appendices/config.h
tml#cfg-in-
null_resource.ansible_config (local-exec): world-writable-dir

null_resource.ansible_config (local-exec): PLAY [Configure backend HTTPD servers] ******************************
************

null_resource.ansible_config (local-exec): TASK [Gathering Facts] ***********************************************
************
null_resource.ansible_config: Still creating... [10s elapsed]
null_resource.ansible_config (local-exec): [WARNING]: Platform linux on host 34.200.242.29 is using the discove
red Python
null_resource.ansible_config (local-exec): interpreter at /usr/bin/python3.7, but future installation of anothe
r Python
null_resource.ansible_config (local-exec): interpreter could change the meaning of that path. See
```



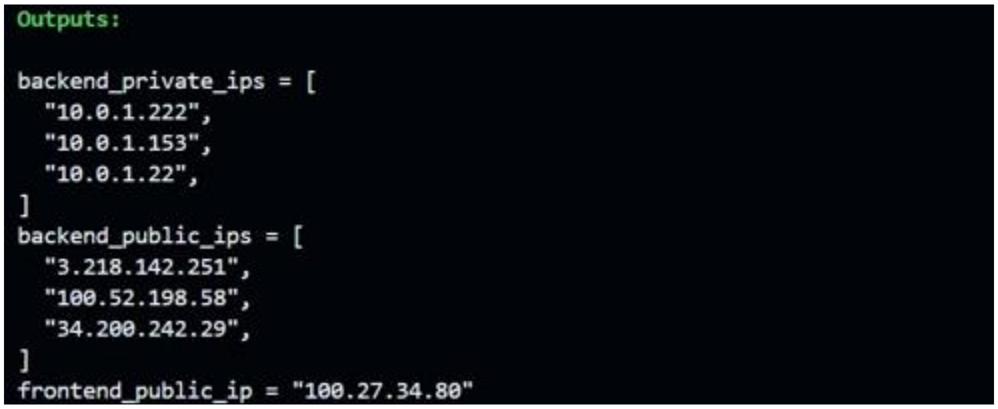
```
red Python
null_resource.ansible_config (local-exec): interpreter at /usr/bin/python3.7, but future installation of anothe
r Python
null_resource.ansible_config (local-exec): interpreter could change the meaning of that path. See
null_resource.ansible_config (local-exec): https://docs.ansible.com/ansible-
null_resource.ansible_config (local-exec): core/2.16/reference_appendices/interpreter_discovery.html for more i
nformation.
null_resource.ansible_config (local-exec): ok: [34.200.242.29]
null_resource.ansible_config (local-exec): ok: [3.218.142.251]
null_resource.ansible_config (local-exec): ok: [100.52.198.58]

null_resource.ansible_config (local-exec): TASK [backend : Install httpd] **************************************
************
null_resource.ansible_config (local-exec): ok: [34.200.242.29]
null_resource.ansible_config (local-exec): ok: [3.218.142.251]
null_resource.ansible_config (local-exec): ok: [100.52.198.58]

null_resource.ansible_config (local-exec): TASK [backend : Enable and start httpd] *****************************
************
null_resource.ansible_config: Still creating... [20s elapsed]
null_resource.ansible_config (local-exec): ok: [34.200.242.29]
null_resource.ansible_config (local-exec): ok: [3.218.142.251]
null_resource.ansible_config (local-exec): ok: [100.52.198.58]
```



```
null_resource.ansible_config (local-exec): TASK [frontend : Enable nginx via amazon-linux-extras (Amazon Linux
2)] ********
null_resource.ansible_config: Still creating... [50s elapsed]
null_resource.ansible_config (local-exec): changed: [100.27.34.80]

null_resource.ansible_config (local-exec): TASK [frontend : Install nginx] *************************************
************
null_resource.ansible_config: Still creating... [1m0s elapsed]
null_resource.ansible_config (local-exec): ok: [100.27.34.80]

null_resource.ansible_config (local-exec): TASK [frontend : Enable and start nginx] ***************************
************
null_resource.ansible_config (local-exec): ok: [100.27.34.80]

null_resource.ansible_config (local-exec): TASK [frontend : Deploy nginx frontend config] *********************
************
null_resource.ansible_config: Still creating... [1m10s elapsed]
null_resource.ansible_config (local-exec): ok: [100.27.34.80]
```



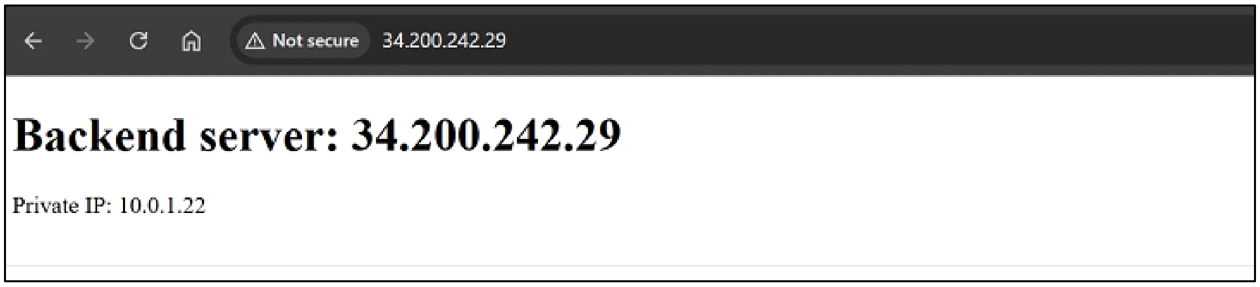
```
null_resource.ansible_config (local-exec): PLAY RECAP ***********************************************************
************
null_resource.ansible_config (local-exec): 100.27.34.80              : ok=5    changed=1    unreachable=0    f
ailed=0    skipped=0    rescued=0    ignored=0
null_resource.ansible_config (local-exec): 100.52.198.58             : ok=4    changed=0    unreachable=0    f
ailed=0    skipped=0    rescued=0    ignored=0
null_resource.ansible_config (local-exec): 3.218.142.251             : ok=4    changed=0    unreachable=0    f
ailed=0    skipped=0    rescued=0    ignored=0
null_resource.ansible_config (local-exec): 34.200.242.29             : ok=4    changed=0    unreachable=0    f
ailed=0    skipped=0    rescued=0    ignored=0

null_resource.ansible_config: Creation complete after 1m15s [id=7963890849977675043]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.
```



```
Outputs:

backend_private_ips = [
  "10.0.1.222",
  "10.0.1.153",
  "10.0.1.22",
]
backend_public_ips = [
  "3.218.142.251",
  "100.52.198.58",
  "34.200.242.29",
]
frontend_public_ip = "100.27.34.80"
```

Since, Ansible runs from Terraform, therefore, our automation is complete.

Run time Behaviour
Backend 1

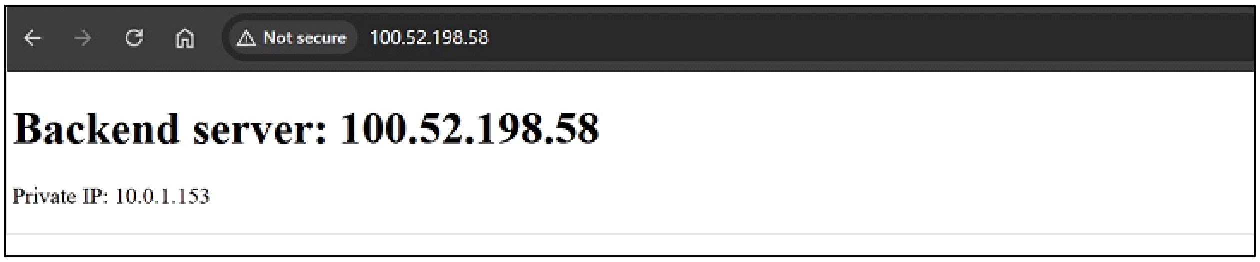


Not secure 34.200.242.29

Backend server: 34.200.242.29

Private IP: 10.0.1.22

Backend 2

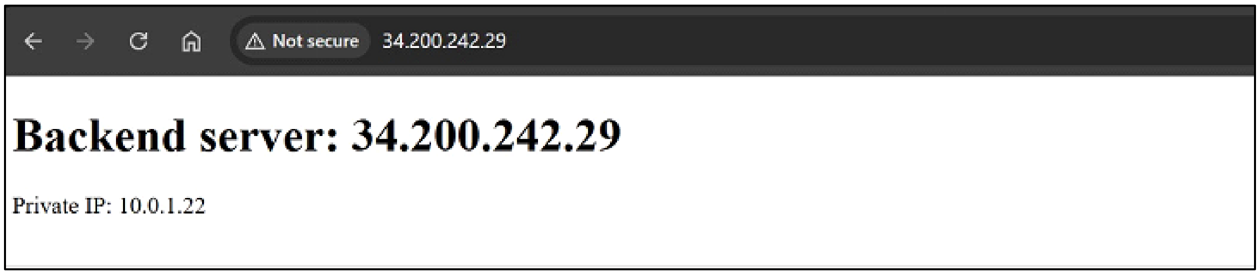


Not secure 100.52.198.58

Backend server: 100.52.198.58

Private IP: 10.0.1.153

Backend 3

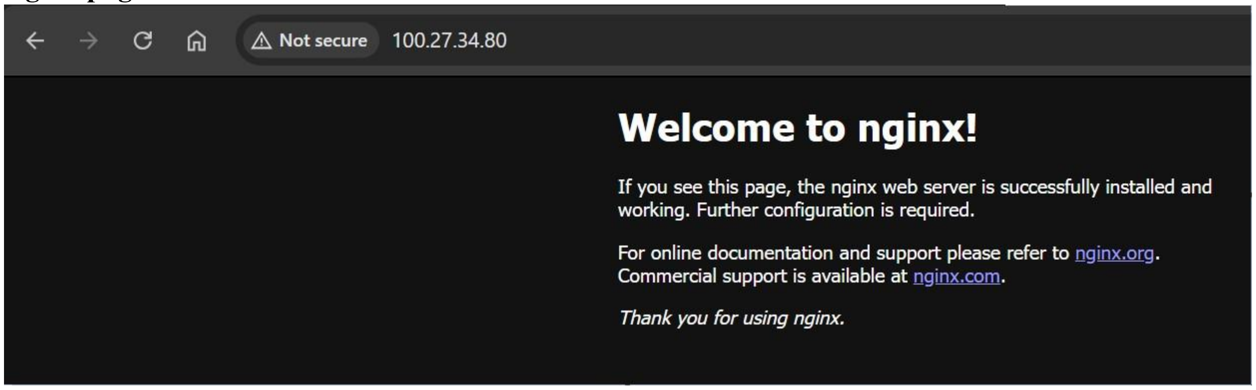


Not secure 34.200.242.29

Backend server: 34.200.242.29

Private IP: 10.0.1.22

Nginx page



Not secure 100.27.34.80

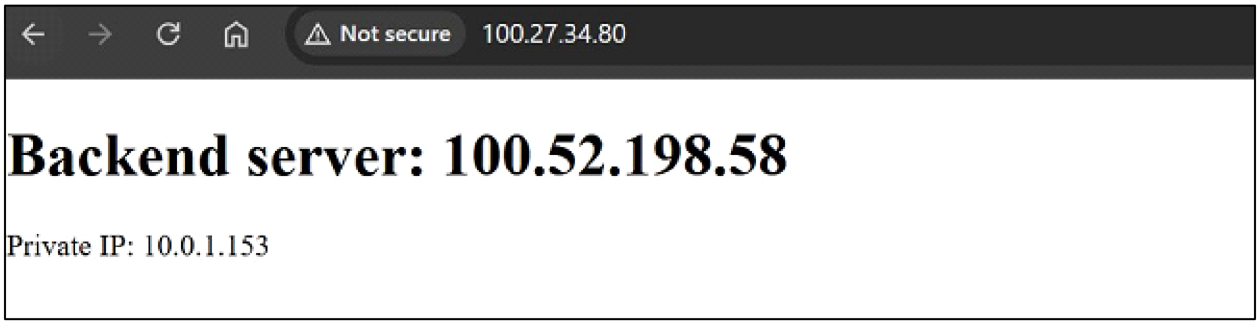
Welcome to nginx!

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.

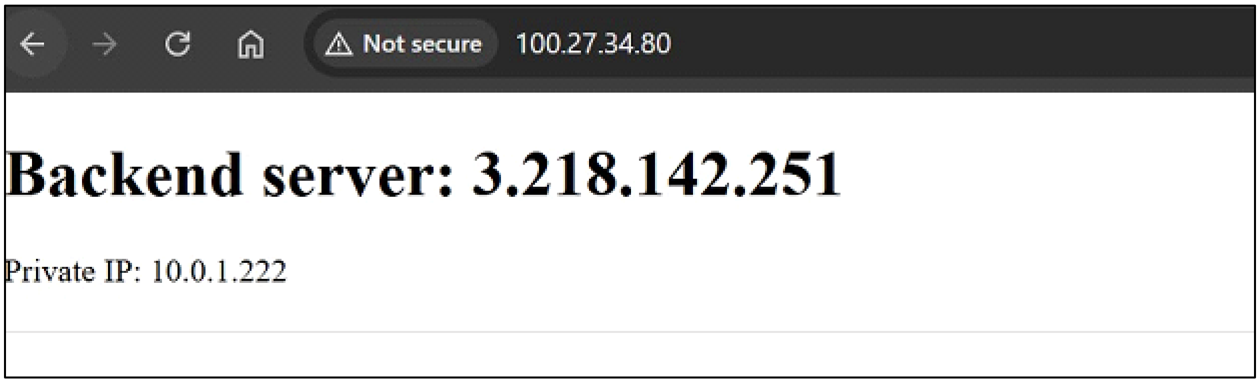
Frontend -> Backend



Not secure 100.27.34.80

Backend server: 100.52.198.58

Private IP: 10.0.1.153



Not secure 100.27.34.80

Backend server: 3.218.142.251

Private IP: 10.0.1.222

Therefore, in this lab I have designed a small **multi-tier AWS architecture** using **Terraform**.

Used **Ansible roles** to separate responsibilities for:

- Frontend Nginx configuration.
- Backend HTTPD configuration.
- (Optional but recommended) Common base configuration.

Configured **Nginx** as a reverse proxy / load balancer with:

- **2 active backend HTTPD servers**.
- **1 backup backend** (used only on primary failure).