Cloud Computing Lab

Safa Jahangir

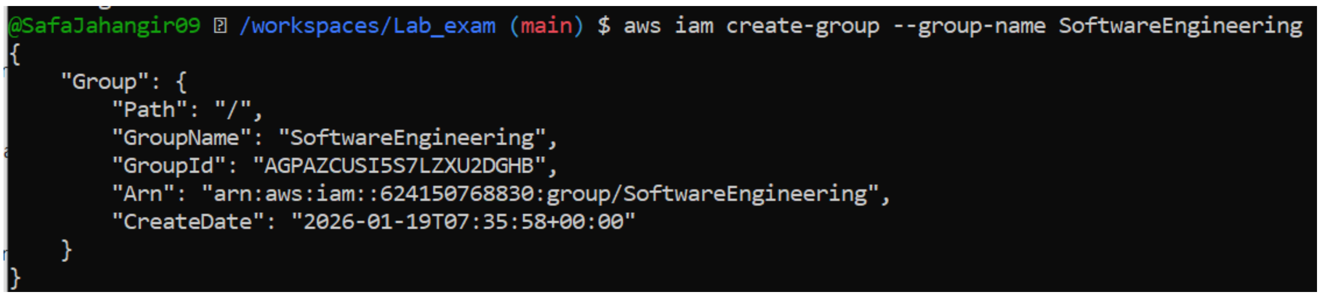
BSE-V B

2023-BSE-056

LAB Exam

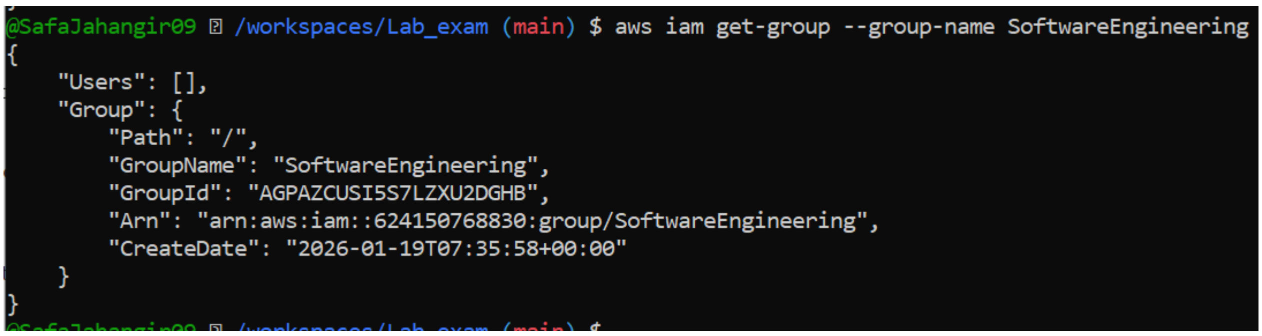
Q1 – AWS IAM Setup Using AWS CLI and Console Verification

- q1_create_group.png



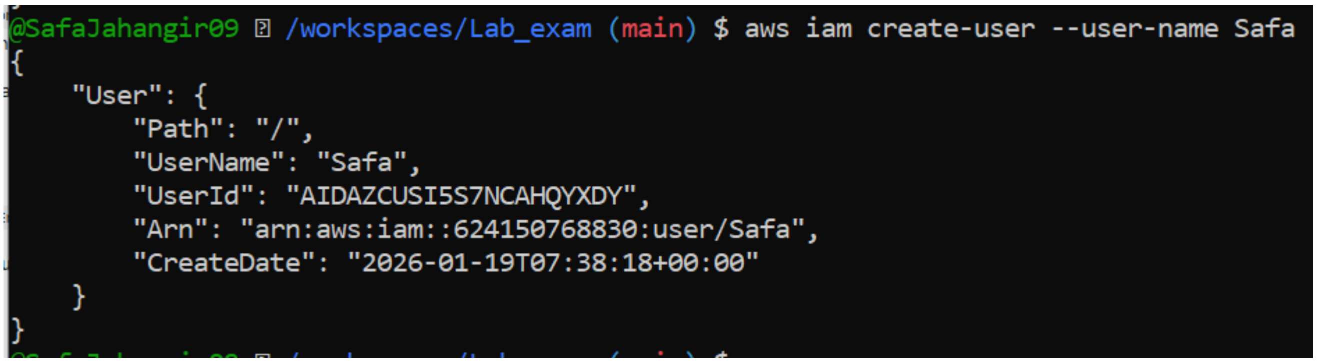
```
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam create-group --group-name SoftwareEngineering
{
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAZCUSI5S7LZXU2DGHB",
        "Arn": "arn:aws:iam::624150768830:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:35:58+00:00"
    }
}
```

- q1_group_details.png



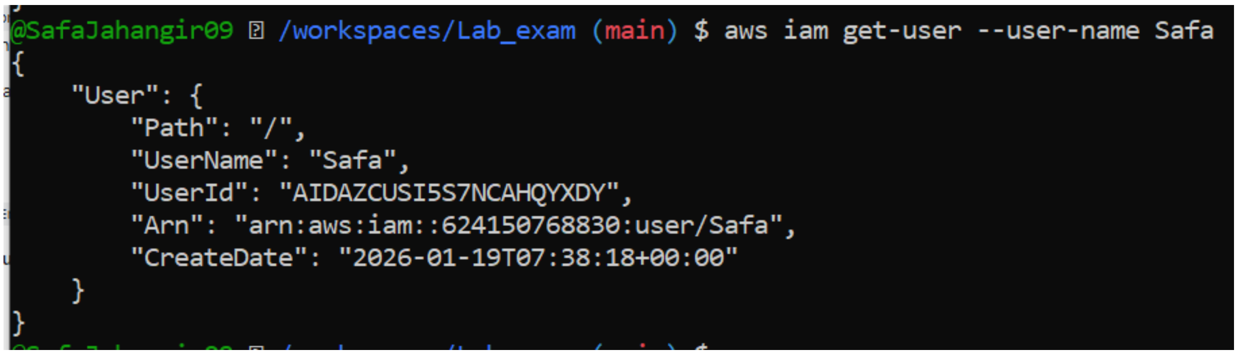
```
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAZCUSI5S7LZXU2DGHB",
        "Arn": "arn:aws:iam::624150768830:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:35:58+00:00"
    }
}
```

- q1_create_user.png



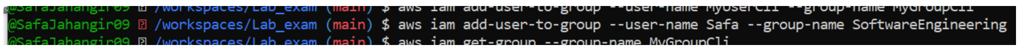
```
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam create-user --user-name Safa
{
    "User": {
        "Path": "/",
        "UserName": "Safa",
        "UserId": "AIDAZCUSI5S7NCAHQYXDY",
        "Arn": "arn:aws:iam::624150768830:user/Safa",
        "CreateDate": "2026-01-19T07:38:18+00:00"
    }
}
```

- q1_user_details.png



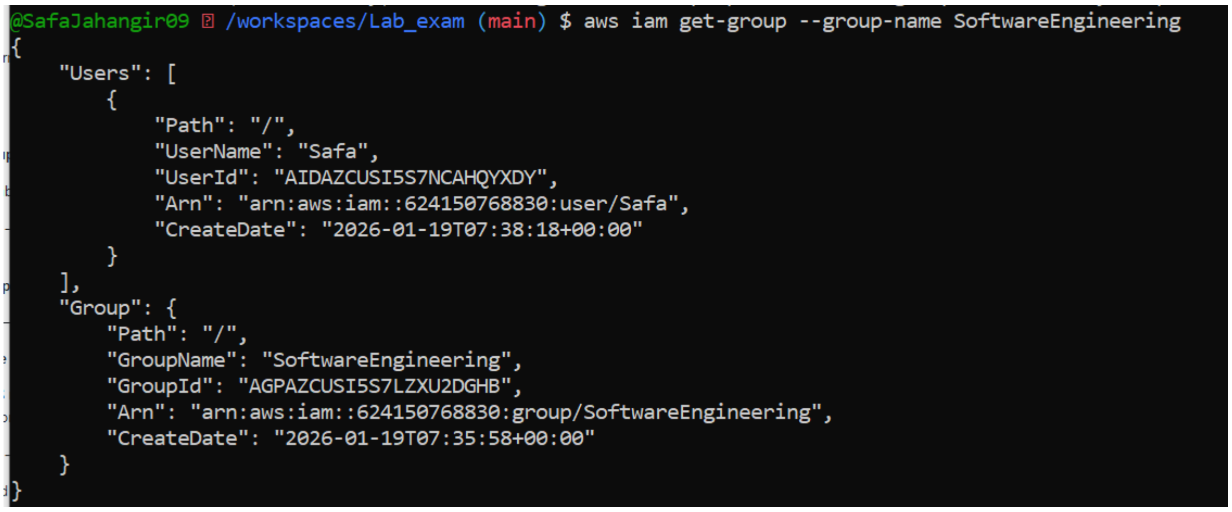
```
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam get-user --user-name Safa
{
    "User": {
        "Path": "/",
        "UserName": "Safa",
        "UserId": "AIDAZCUSI5S7NCAHQYXDY",
        "Arn": "arn:aws:iam::624150768830:user/Safa",
        "CreateDate": "2026-01-19T07:38:18+00:00"
    }
}
```

- q1_add_user_to_group.png



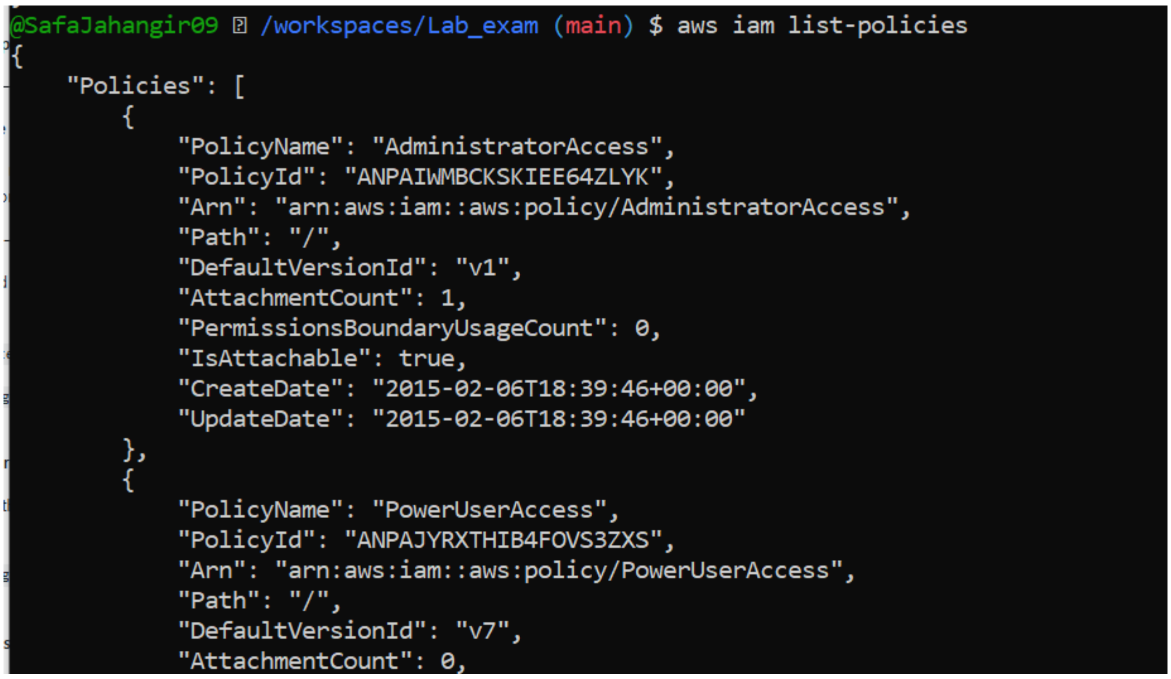
```
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam add-user-to-group --user-name MyUserCli --group-name MyGroupCli
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam add-user-to-group --user-name Safa --group-name SoftwareEngineering
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam get-group --group-name MyGroupCli
```

- q1_group_membership.png



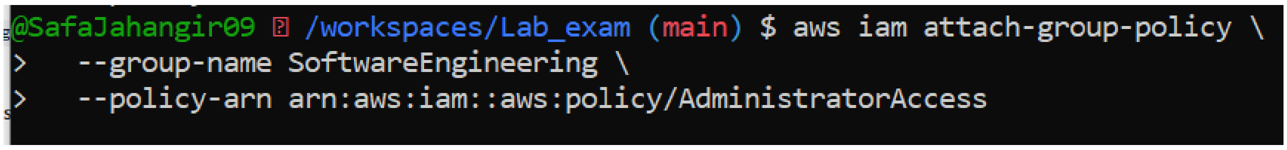
```
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Safa",
            "UserId": "AIDAZCUSI5S7NCAHQYXDY",
            "Arn": "arn:aws:iam::624150768830:user/Safa",
            "CreateDate": "2026-01-19T07:38:18+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAZCUSI5S7LZXU2DGHB",
        "Arn": "arn:aws:iam::624150768830:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:35:58+00:00"
    }
}
```

- q1_find_admin_policy.png



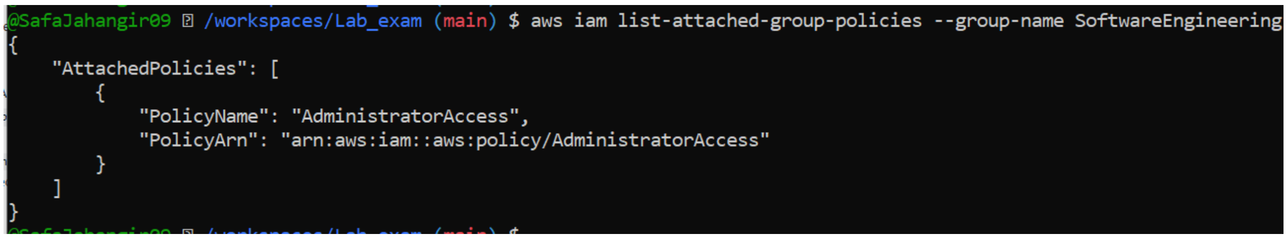
```
@SafaJahangir09 ⍰ /workspaces/Lab_exam (main) $ aws iam list-policies
{
    "Policies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 1,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:46+00:00",
            "UpdateDate": "2015-02-06T18:39:46+00:00"
        },
        {
            "PolicyName": "PowerUserAccess",
            "PolicyId": "ANPAJYRXTHIB4FOVS3ZXS",
            "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
            "Path": "/",
            "DefaultVersionId": "v7",
            "AttachmentCount": 0,
```

- q1_attach_admin_policy.png



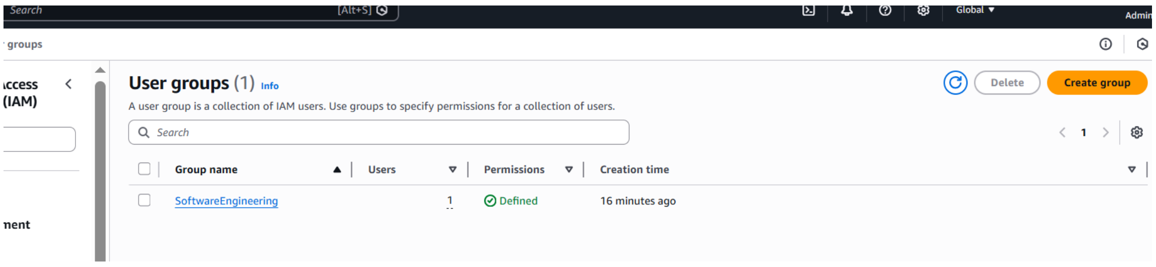
```
@SafaJahangir09 ⍰ /workspaces/Lab_exam (main) $ aws iam attach-group-policy \
>   --group-name SoftwareEngineering \
>   --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

- q1_list_group_policies.png



```
@SafaJahangir09 ⍰ /workspaces/Lab_exam (main) $ aws iam list-attached-group-policies --group-name SoftwareEngineering
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
```

- q1_console_group.png

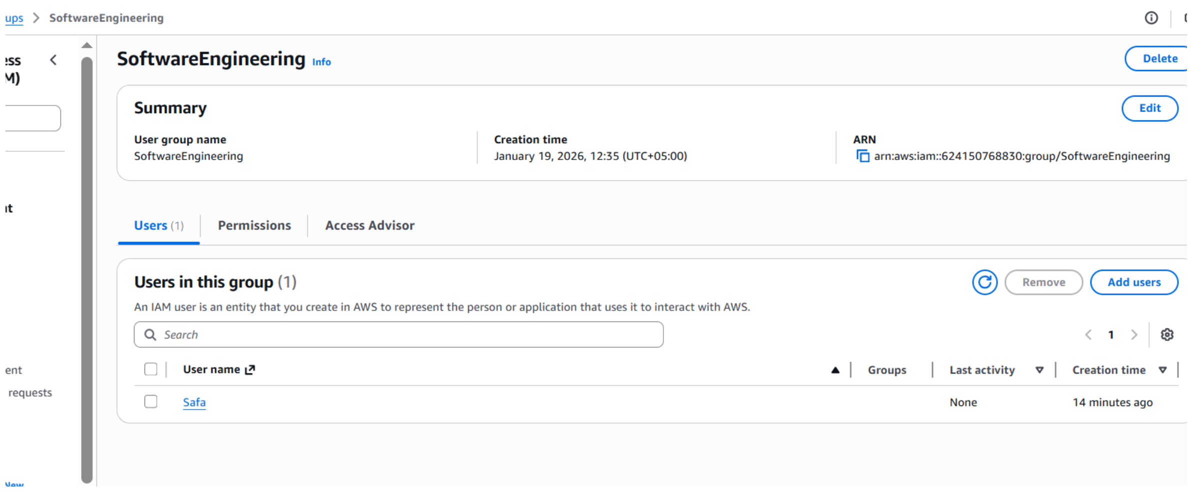


User groups (1) Info

A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

Group name	Users	Permissions	Creation time
SoftwareEngineering	1	Defined	16 minutes ago

- q1_console_user_in_group.png



ups > SoftwareEngineering

SoftwareEngineering Info

Summary

User group name
SoftwareEngineering

Creation time
January 19, 2026, 12:35 (UTC+05:00)

ARN
arn:aws:iam::624150768830:group/SoftwareEngineering

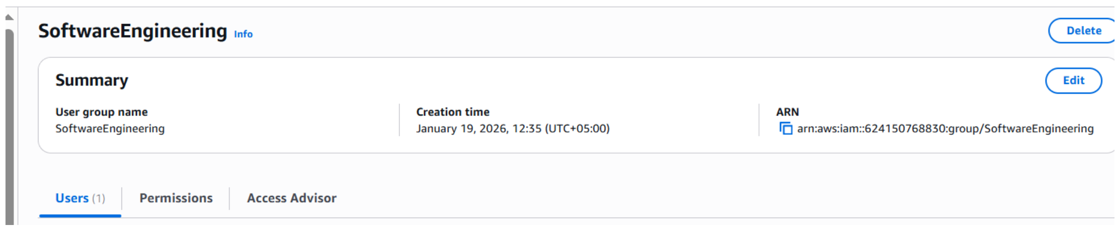
Users (1) | Permissions | Access Advisor

Users in this group (1)

An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

User name	Groups	Last activity	Creation time
Safa		None	14 minutes ago

- q1_console_group_policy.png



SoftwareEngineering Info

Summary

User group name
SoftwareEngineering

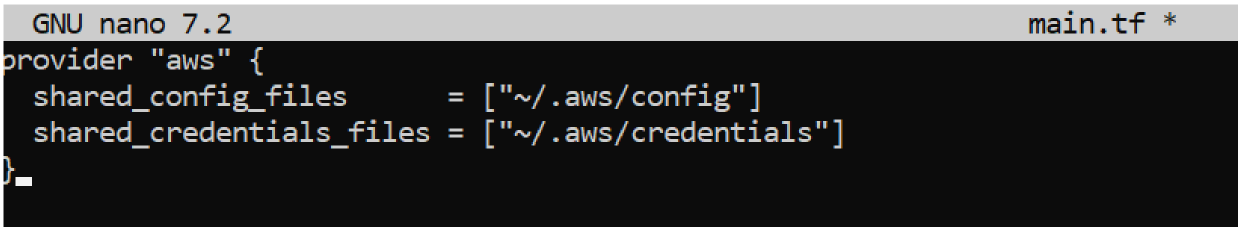
Creation time
January 19, 2026, 12:35 (UTC+05:00)

ARN
arn:aws:iam::624150768830:group/SoftwareEngineering

Users (1) | Permissions | Access Advisor

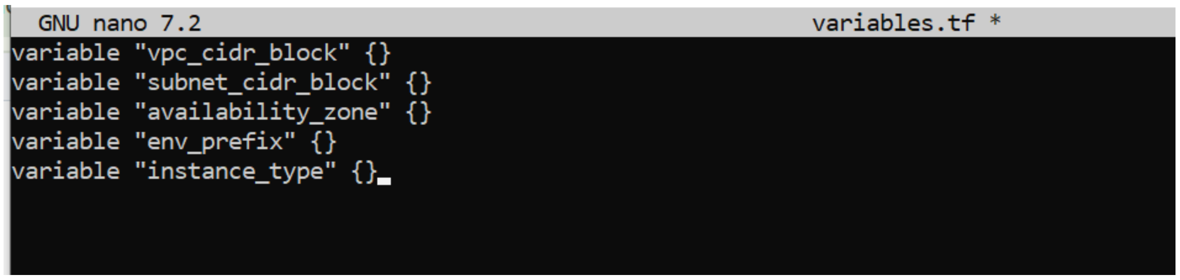
Q2 – Terraform Lab: Simple AWS Environment with Nginx over HTTPS

- q2_provider.png



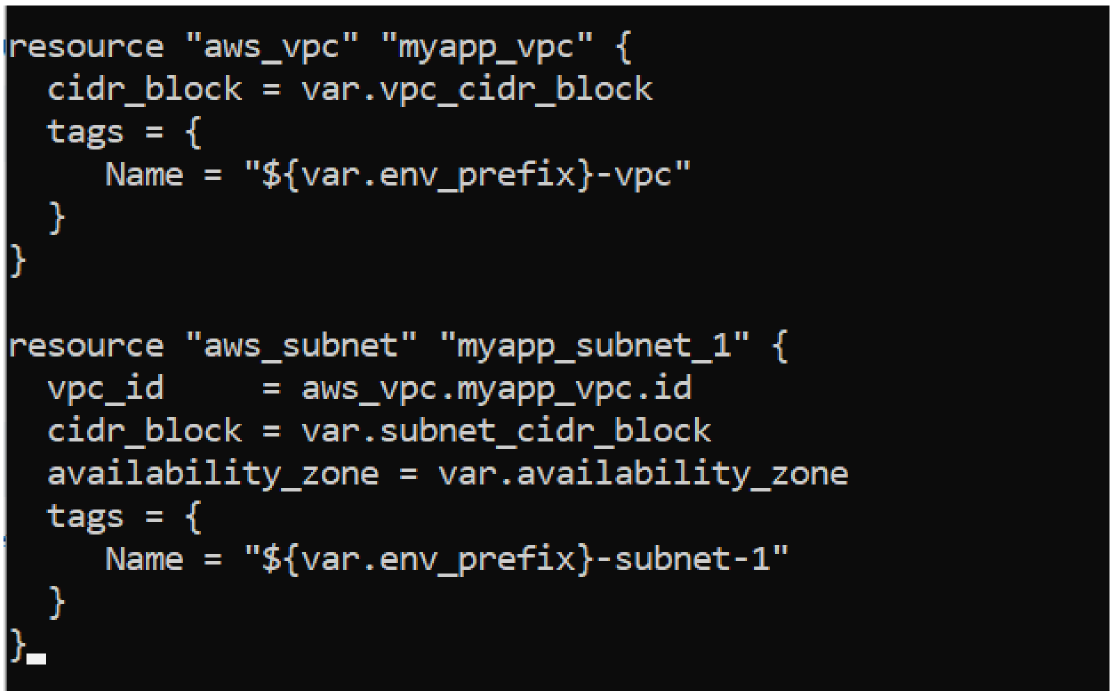
```
  GNU nano 7.2                                         main.tf *
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
```

- q2_variables.png



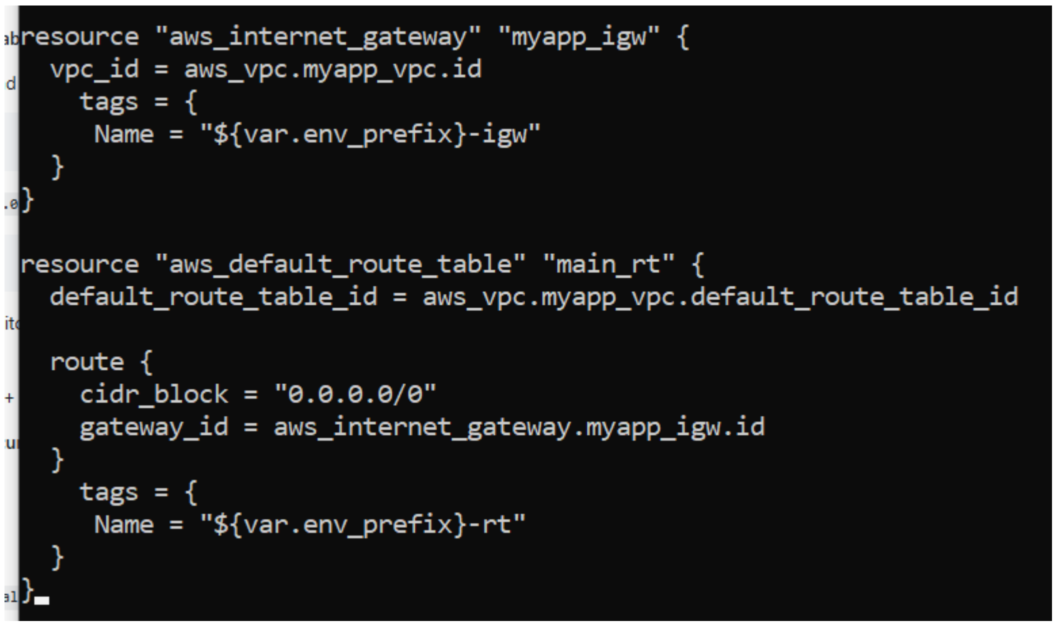
```
  GNU nano 7.2                                    variables.tf *
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
```

- q2_vpc_subnet.png



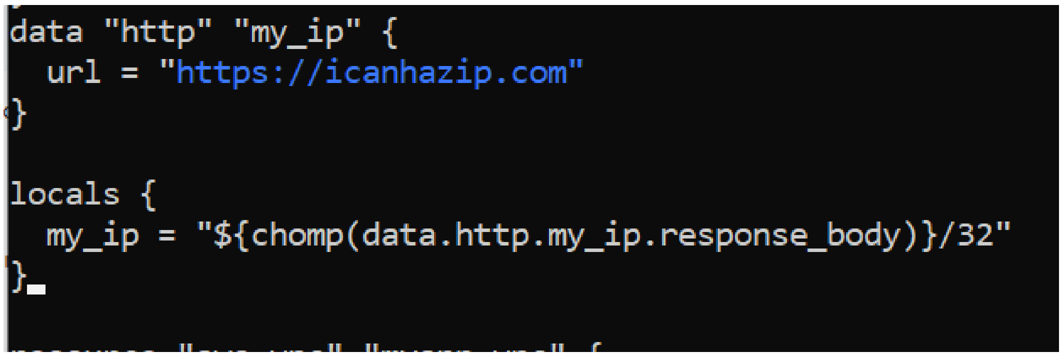
```
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
     Name = "${var.env_prefix}-vpc"
  }
}

resource "aws_subnet" "myapp_subnet_1" {
  vpc_id     = aws_vpc.myapp_vpc.id
  cidr_block = var.subnet_cidr_block
  availability_zone = var.availability_zone
  tags = {
     Name = "${var.env_prefix}-subnet-1"
  }
}
```

- q2_igw_route_table.png



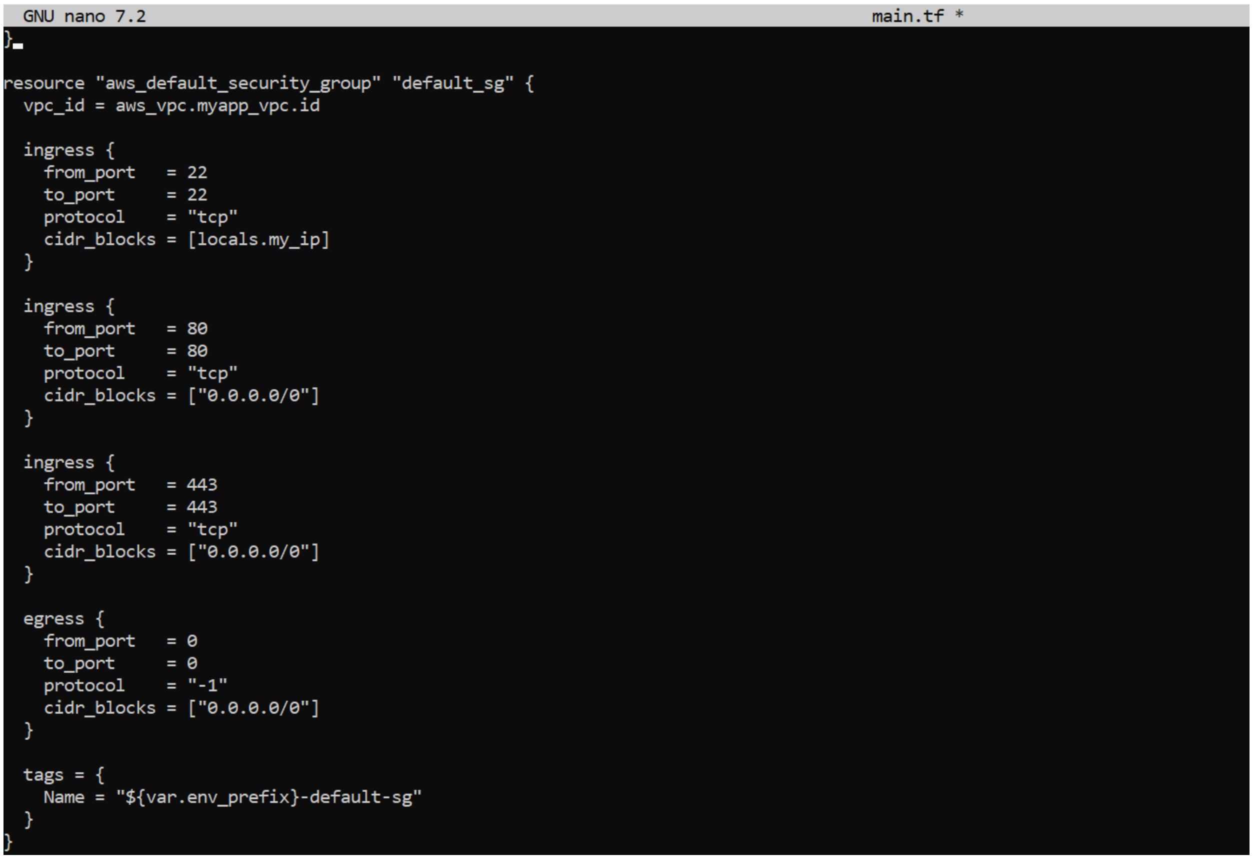
```
resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id
    tags = {
     Name = "${var.env_prefix}-igw"
  }
}

resource "aws_default_route_table" "main_rt" {
  default_route_table_id = aws_vpc.myapp_vpc.default_route_table_id

  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.myapp_igw.id
  }
    tags = {
     Name = "${var.env_prefix}-rt"
  }
}
```

- q2_http_and_locals.png



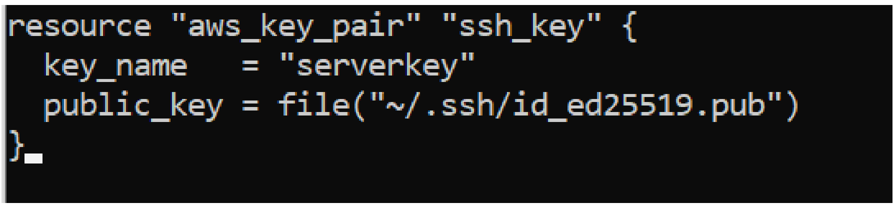
```
data "http" "my_ip" {
  url = "https://icanhazip.com"
}

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

- q2_default_sg.png



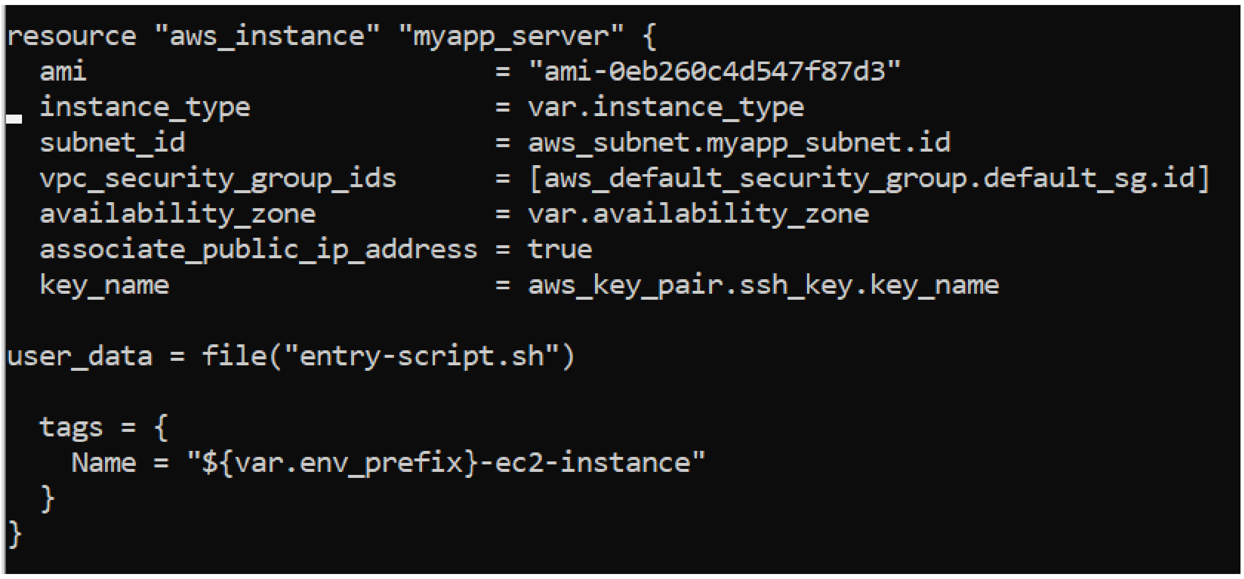
```
GNU nano 7.2                                                    main.tf *
}

resource "aws_default_security_group" "default_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

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

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

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

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

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

- q2_keypair.png



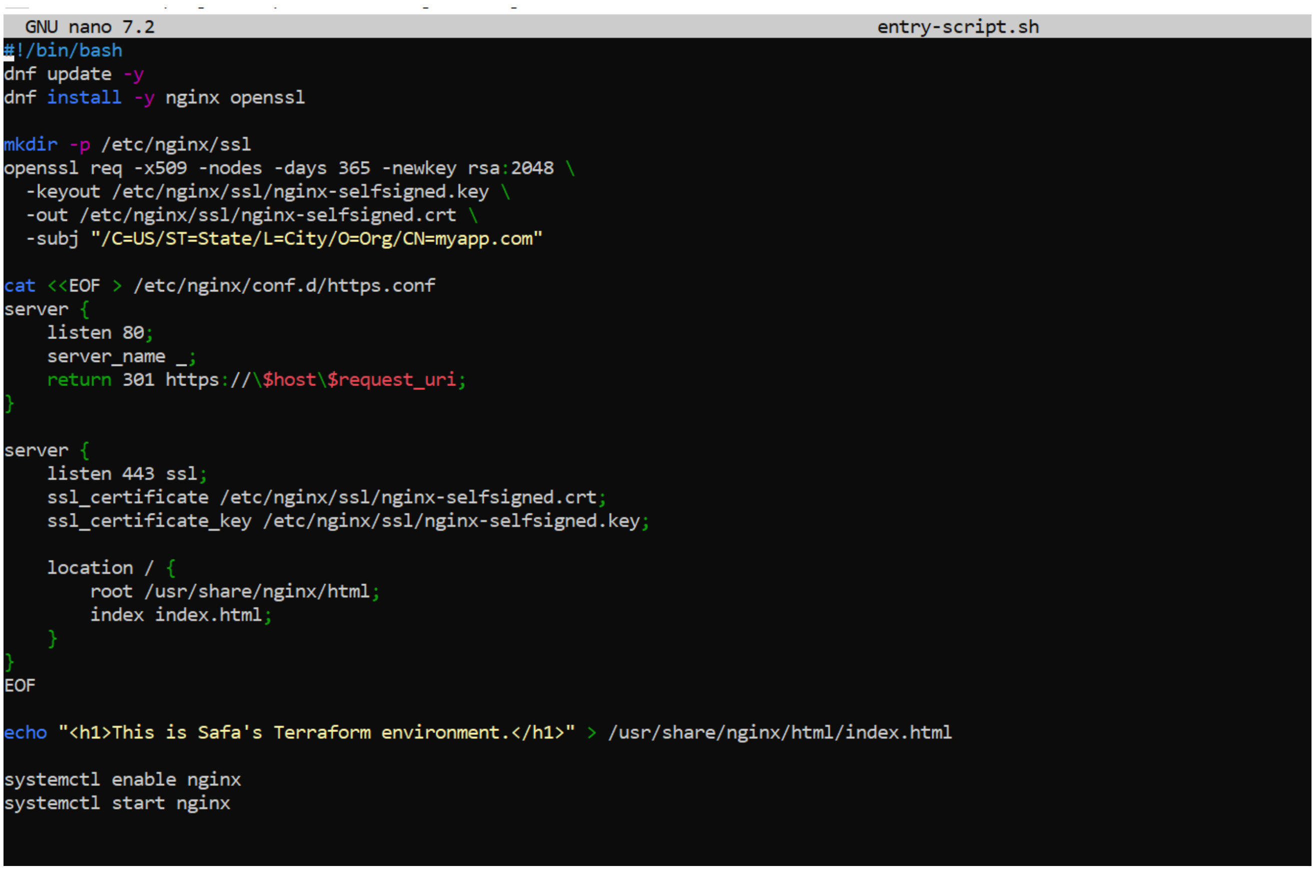
```
resource "aws_key_pair" "ssh_key" {
  key_name   = "serverkey"
  public_key = file("~/.ssh/id_ed25519.pub")
}
```

- q2_ec2_resource.png



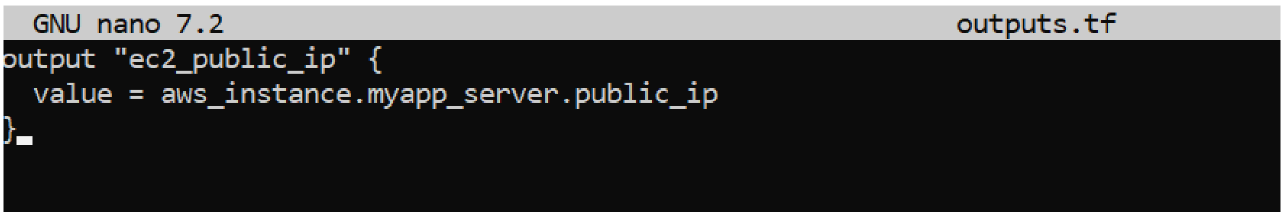
```
resource "aws_instance" "myapp_server" {
  ami                         = "ami-0eb260c4d547f87d3"
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet.id
  vpc_security_group_ids      = [aws_default_security_group.default_sg.id]
  availability_zone           = var.availability_zone
  associate_public_ip_address = true
  key_name                    = aws_key_pair.ssh_key.key_name

user_data = file("entry-script.sh")

  tags = {
    Name = "${var.env_prefix}-ec2-instance"
  }
}
```

- q2_entry_script.png



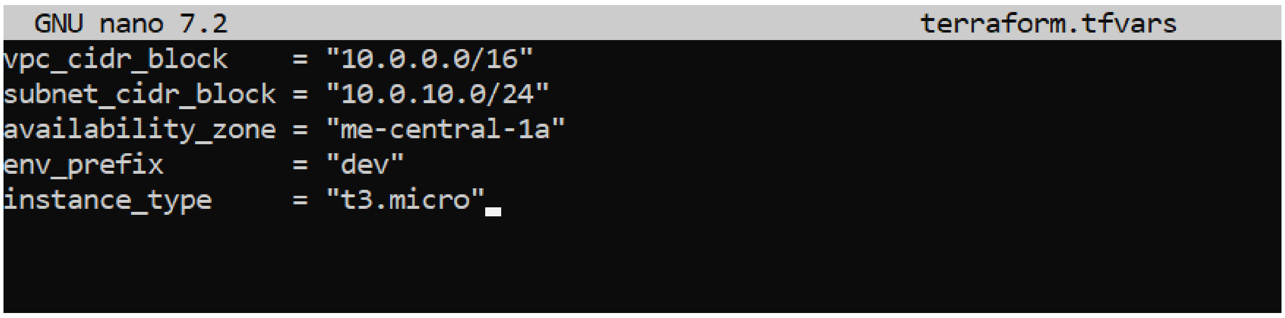
```
  GNU nano 7.2                                                                         entry-script.sh
#!/bin/bash
dnf update -y
dnf install -y nginx openssl

mkdir -p /etc/nginx/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/nginx/ssl/nginx-selfsigned.key \
  -out /etc/nginx/ssl/nginx-selfsigned.crt \
  -subj "/C=US/ST=State/L=City/O=Org/CN=myapp.com"

cat <<EOF > /etc/nginx/conf.d/https.conf
server {
    listen 80;
    server_name _;
    return 301 https://\$host\$request_uri;
}

server {
    listen 443 ssl;
    ssl_certificate /etc/nginx/ssl/nginx-selfsigned.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx-selfsigned.key;

    location / {
        root /usr/share/nginx/html;
        index index.html;
    }
}
EOF

echo "<h1>This is Safa's Terraform environment.</h1>" > /usr/share/nginx/html/index.html

systemctl enable nginx
systemctl start nginx
```

- q2_output_block.png



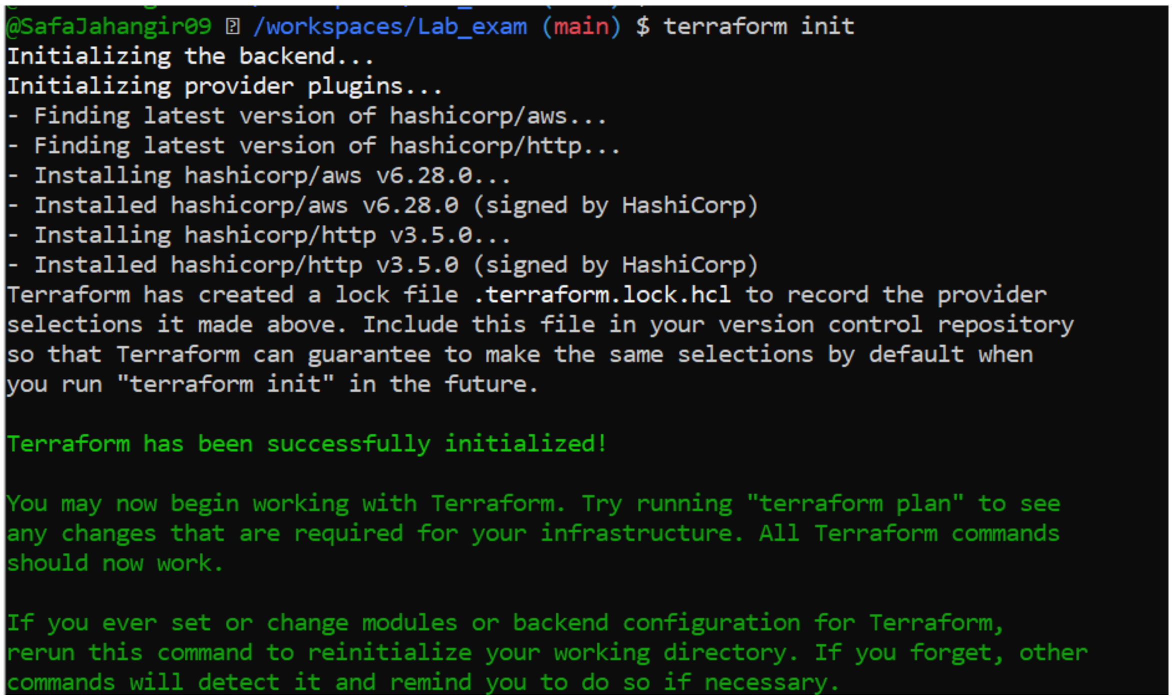
```
  GNU nano 7.2                                  outputs.tf
output "ec2_public_ip" {
  value = aws_instance.myapp_server.public_ip
}
```

- q2_tfvars_or_vars.png



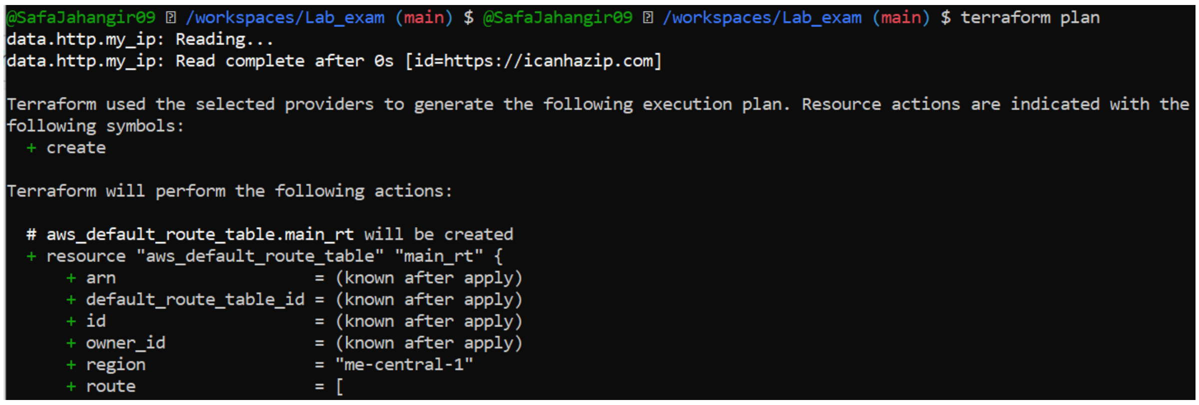
```
  GNU nano 7.2                              terraform.tfvars
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"
env_prefix         = "dev"
instance_type      = "t3.micro"
```

- q2_terraform_init.png



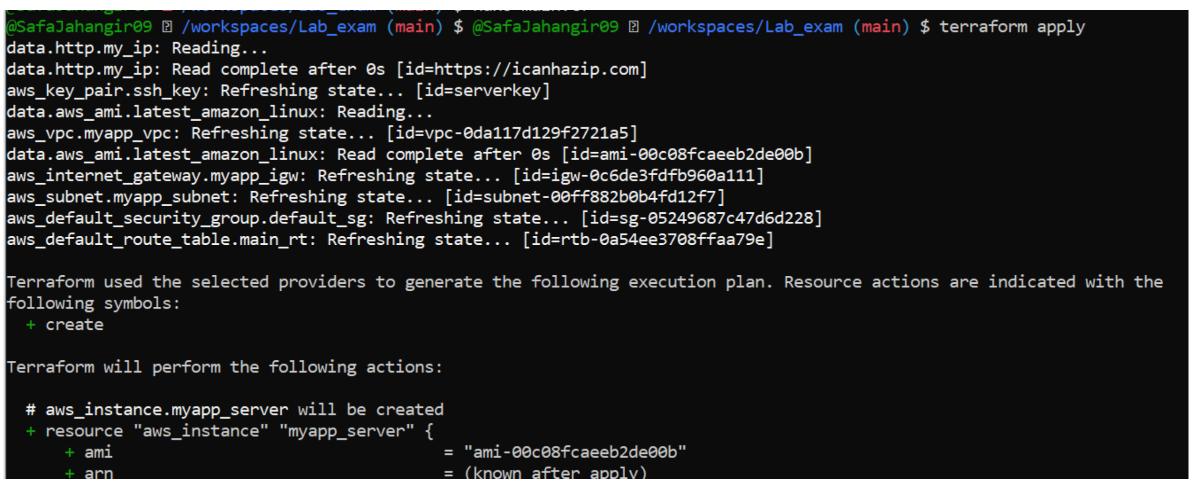
```
@SafaJahangir09 ▯ /workspaces/Lab_exam (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (signed by 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.
```

- q2_terraform_plan.png



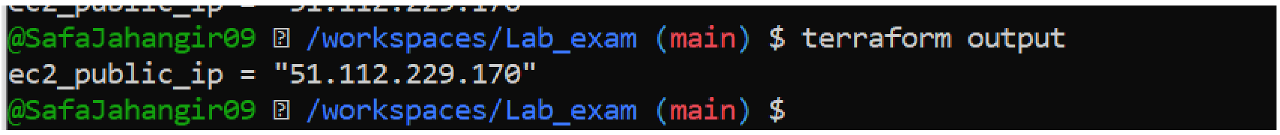
```
@SafaJahangir09 ▯ /workspaces/Lab_exam (main) $ @SafaJahangir09 ▯ /workspaces/Lab_exam (main) $ terraform plan
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]

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:

  # aws_default_route_table.main_rt will be created
  + resource "aws_default_route_table" "main_rt" {
      + arn                    = (known after apply)
      + default_route_table_id = (known after apply)
      + id                     = (known after apply)
      + owner_id               = (known after apply)
      + region                 = "me-central-1"
      + route                  = [
```

- q2_terraform_apply.png



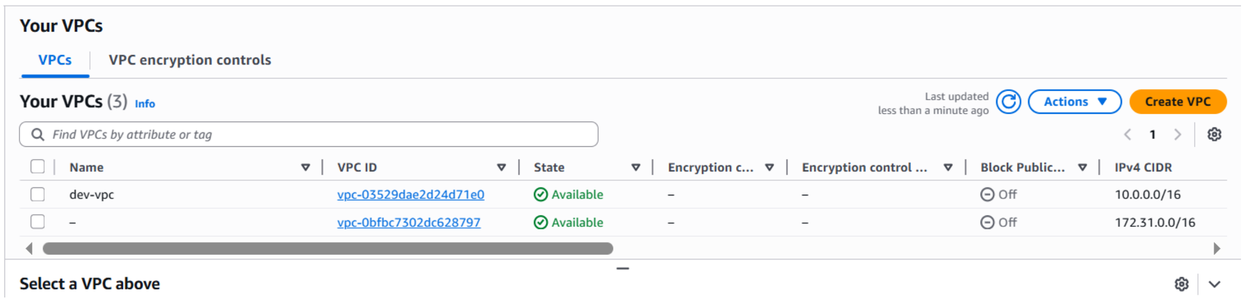
```
@SafaJahangir09 ▯ /workspaces/Lab_exam (main) $ @SafaJahangir09 ▯ /workspaces/Lab_exam (main) $ terraform apply
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]
aws_key_pair.ssh_key: Refreshing state... [id=serverkey]
data.aws_ami.latest_amazon_linux: Reading...
aws_vpc.myapp_vpc: Refreshing state... [id=vpc-0da117d129f2721a5]
data.aws_ami.latest_amazon_linux: Read complete after 0s [id=ami-00c08fcaeeb2de00b]
aws_internet_gateway.myapp_igw: Refreshing state... [id=igw-0c6de3fdfb960a111]
aws_subnet.myapp_subnet: Refreshing state... [id=subnet-00ff882b0b4fd12f7]
aws_default_security_group.default_sg: Refreshing state... [id=sg-05249687c47d6d228]
aws_default_route_table.main_rt: Refreshing state... [id=rtb-0a54ee3708ffaa79e]

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:

  # aws_instance.myapp_server will be created
  + resource "aws_instance" "myapp_server" {
      + ami                                  = "ami-00c08fcaeeb2de00b"
      + arn                                  = (known after apply)
```

- q2_terraform_output.png



```
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $ terraform output
ec2_public_ip = "51.112.229.170"
@SafaJahangir09 ➜ /workspaces/Lab_exam (main) $
```

- q2_console_vpc.png



Your VPCs

VPCs | VPC encryption controls

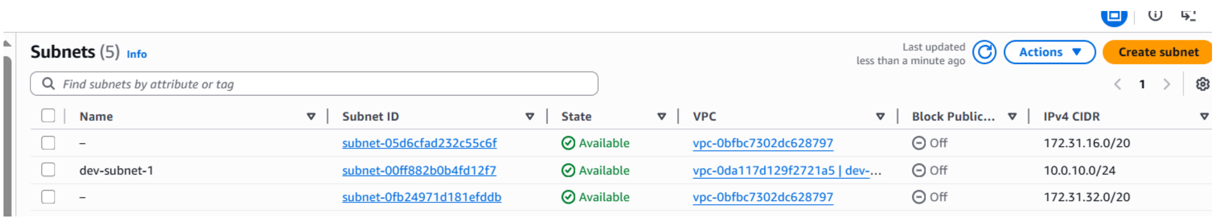
Your VPCs (3) Info

Last updated less than a minute ago

	Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR
☐	dev-vpc	vpc-03529dae2d24d71e0	Available	–	–	Off	10.0.0.0/16
☐	–	vpc-0bfbc7302dc628797	Available	–	–	Off	172.31.0.0/16

Select a VPC above

- q2_console_subnet.png

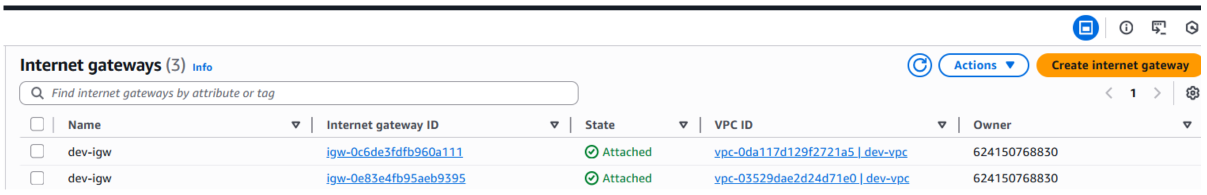


Subnets (5) Info

Last updated less than a minute ago

	Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
☐	–	subnet-05d6cfad232c55c6f	Available	vpc-0bfbc7302dc628797	Off	172.31.16.0/20
☐	dev-subnet-1	subnet-00ff882b0b4fd12f7	Available	vpc-0da117d129f2721a5 | dev-...	Off	10.0.10.0/24
☐	–	subnet-0fb24971d181efddb	Available	vpc-0bfbc7302dc628797	Off	172.31.32.0/20

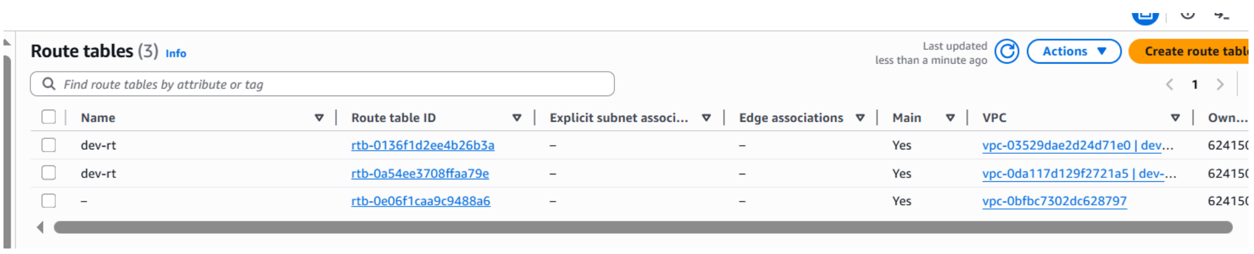
- q2_console_igw.png



Internet gateways (3) Info

	Name	Internet gateway ID	State	VPC ID	Owner
☐	dev-igw	igw-0c6de3fdfb960a111	Attached	vpc-0da117d129f2721a5 | dev-vpc	624150768830
☐	dev-igw	igw-0e83e4fb95aeb9395	Attached	vpc-03529dae2d24d71e0 | dev-vpc	624150768830

- q2_console_route_table.png

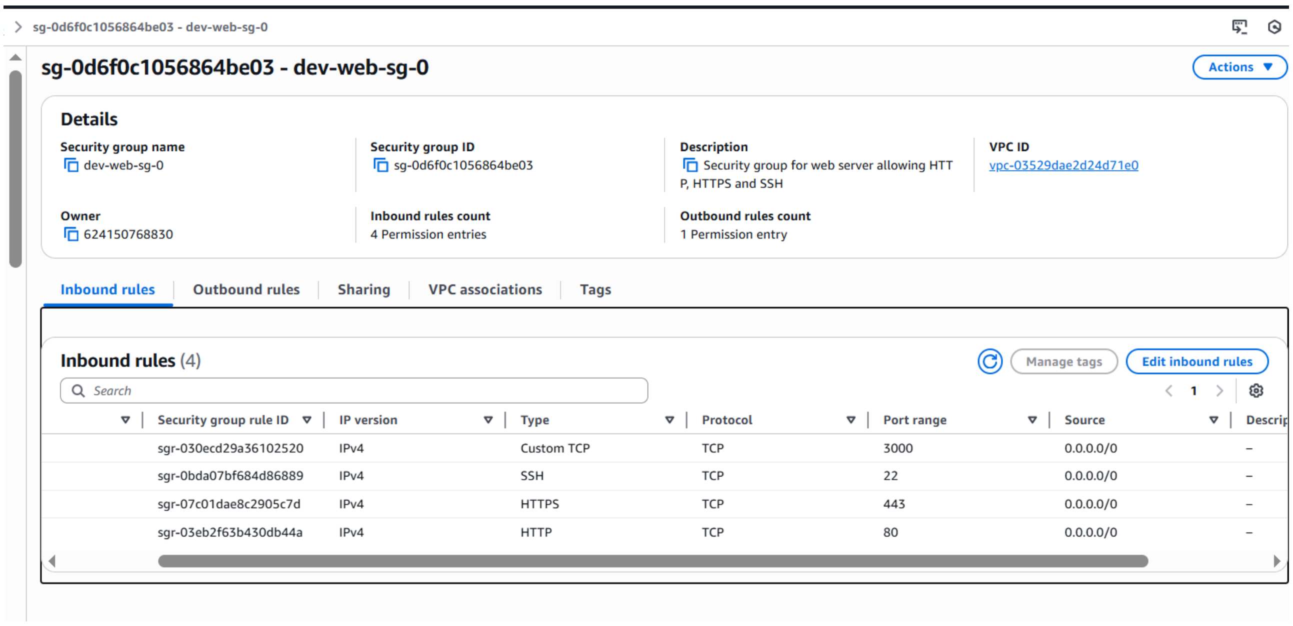


Route tables (3) Info

Last updated less than a minute ago

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Own...
☐	dev-rt	rtb-0136f1d2ee4b26b3a	–	–	Yes	vpc-03529dae2d24d71e0 | dev...	62415
☐	dev-rt	rtb-0a54ee3708ffaa79e	–	–	Yes	vpc-0da117d129f2721a5 | dev-...	62415
☐	–	rtb-0e06f1caa9c9488a6	–	–	Yes	vpc-0bfbc7302dc628797	62415

- q2_console_sg.png



sg-0d6f0c1056864be03 - dev-web-sg-0

sg-0d6f0c1056864be03 - dev-web-sg-0

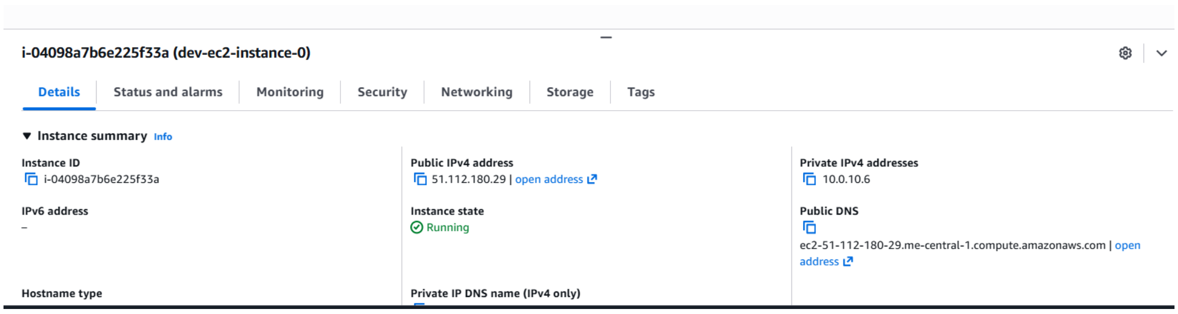
Details

Security group name	**Security group ID**	**Description**	**VPC ID**
dev-web-sg-0	sg-0d6f0c1056864be03	Security group for web server allowing HTTP, HTTPS and SSH	vpc-03529dae2d24d71e0
Owner	**Inbound rules count**	**Outbound rules count**	
624150768830	4 Permission entries	1 Permission entry	

Inbound rules (4)

Security group rule ID	IP version	Type	Protocol	Port range	Source	Descrip
sgr-030ecd29a36102520	IPv4	Custom TCP	TCP	3000	0.0.0.0/0	–
sgr-0bda07bf684d86889	IPv4	SSH	TCP	22	0.0.0.0/0	–
sgr-07c01dae8c2905c7d	IPv4	HTTPS	TCP	443	0.0.0.0/0	–
sgr-03eb2f63b430db44a	IPv4	HTTP	TCP	80	0.0.0.0/0	–

- q2_console_ec2.png



i-04098a7b6e225f33a (dev-ec2-instance-0)

Instance summary Info

Instance ID
i-04098a7b6e225f33a

Public IPv4 address
51.112.180.29 | open address

Private IPv4 addresses
10.0.10.6

IPv6 address
–

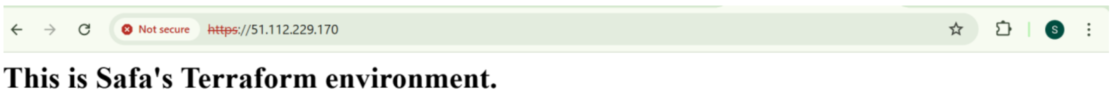
Instance state
Running

Public DNS
ec2-51-112-180-29.me-central-1.compute.amazonaws.com | open address

Hostname type

Private IP DNS name (IPv4 only)

- q2_https_browser.png

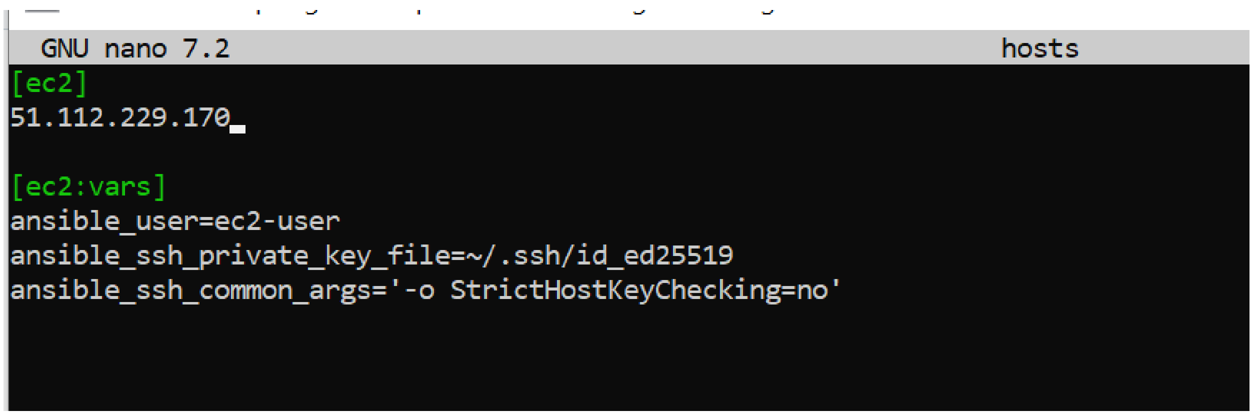


Not secure ~~https~~://51.112.229.170

This is Safa's Terraform environment.

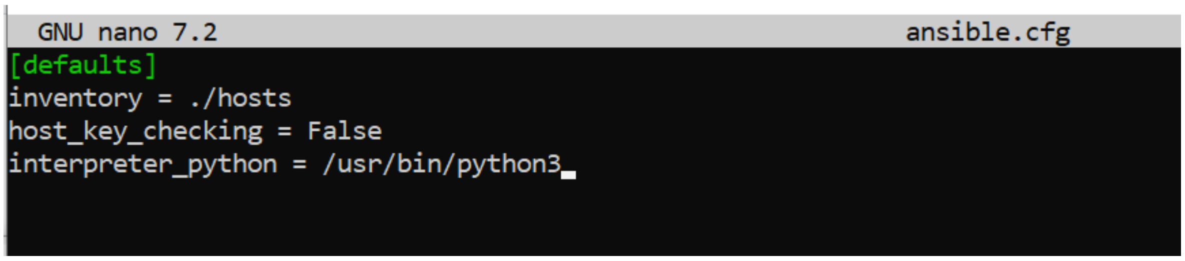
Q3 – Ansible Playbook for EC2 Web Server Using Q2 Instance

- q3_hosts.png



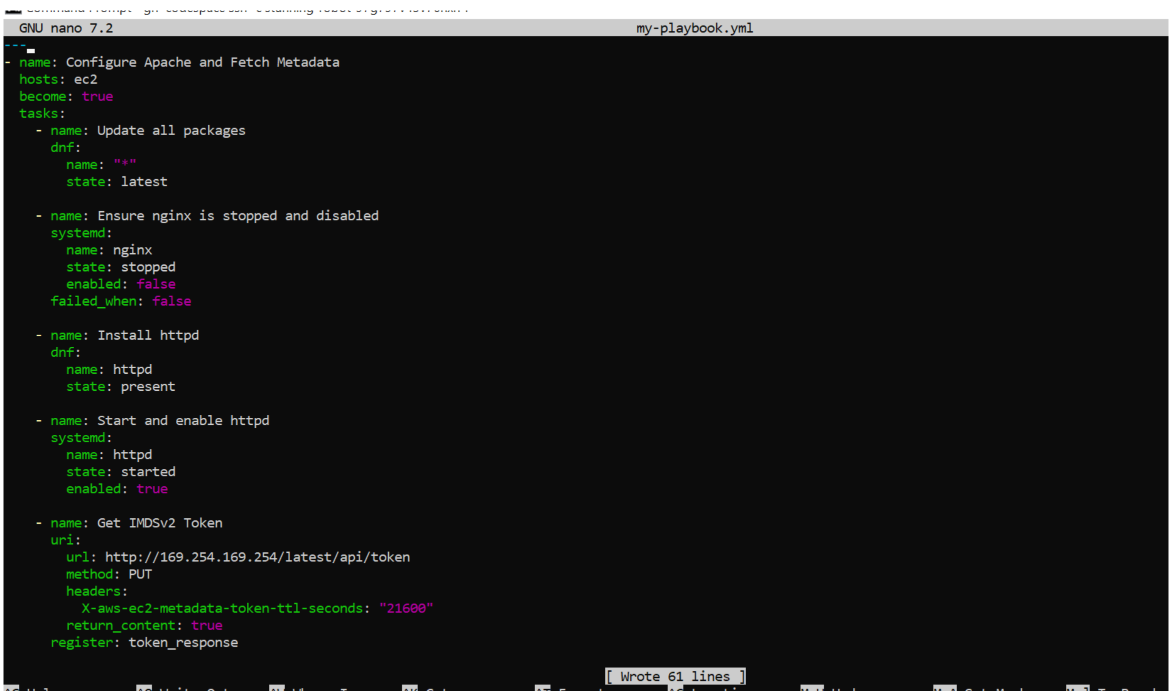
```
  GNU nano 7.2                                                    hosts
[ec2]
51.112.229.170

[ec2:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```

- q3_ansible_cfg.png



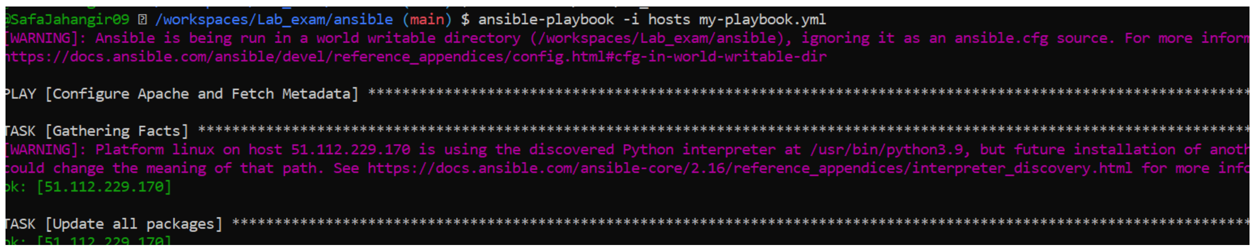
```
GNU nano 7.2                                          ansible.cfg
[defaults]
inventory = ./hosts
host_key_checking = False
interpreter_python = /usr/bin/python3
```

- q3_playbook.png

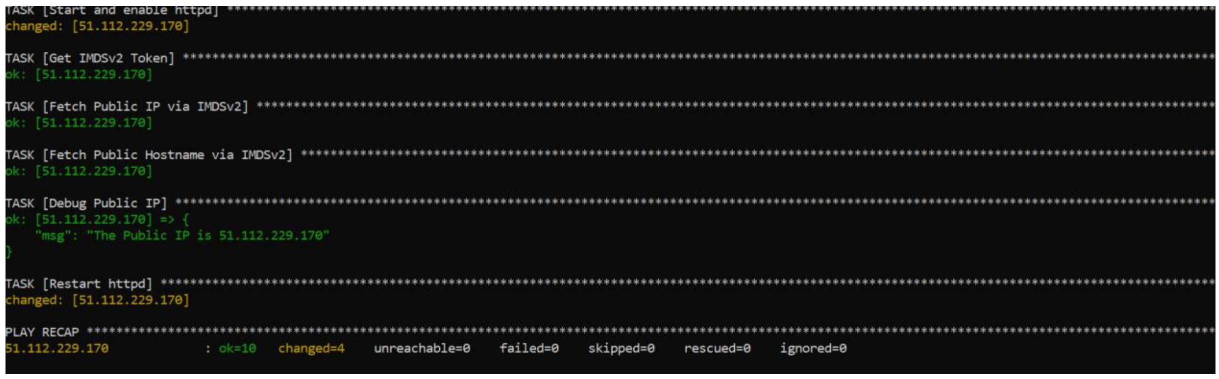


```
GNU nano 7.2                                          my-playbook.yml
---
- name: Configure Apache and Fetch Metadata
  hosts: ec2
  become: true
  tasks:
    - name: Update all packages
      dnf:
        name: "*"
        state: latest

    - name: Ensure nginx is stopped and disabled
      systemd:
        name: nginx
        state: stopped
        enabled: false
      failed_when: false

    - name: Install httpd
      dnf:
        name: httpd
        state: present

    - name: Start and enable httpd
      systemd:
        name: httpd
        state: started
        enabled: true

    - name: Get IMDSv2 Token
      uri:
        url: http://169.254.169.254/latest/api/token
        method: PUT
        headers:
          X-aws-ec2-metadata-token-ttl-seconds: "21600"
        return_content: true
      register: token_response

                                            [ Wrote 61 lines ]
```

- q3_play_run.png

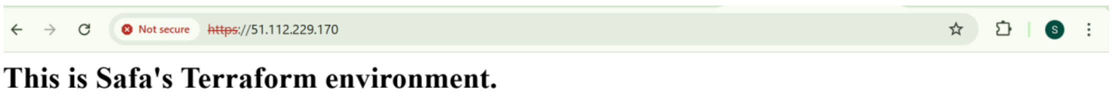


```
@SafaJahangir09 ➜ /workspaces/Lab_exam/ansible (main) $ ansible-playbook -i hosts my-playbook.yml
[WARNING]: Ansible is being run in a world writable directory (/workspaces/Lab_exam/ansible), ignoring it as an ansible.cfg source. For more inform
https://docs.ansible.com/ansible/devel/reference_appendices/config.html#cfg-in-world-writable-dir

PLAY [Configure Apache and Fetch Metadata] *****************************************************************************************

TASK [Gathering Facts] *************************************************************************************************************
[WARNING]: Platform linux on host 51.112.229.170 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of anoth
could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more info
ok: [51.112.229.170]

TASK [Update all packages] *********************************************************************************************************
ok: [51.112.229.170]
```



```
TASK [Start and enable httpd] ******************************************************************************************************
changed: [51.112.229.170]

TASK [Get IMDSv2 Token] ************************************************************************************************************
ok: [51.112.229.170]

TASK [Fetch Public IP via IMDSv2] **************************************************************************************************
ok: [51.112.229.170]

TASK [Fetch Public Hostname via IMDSv2] ********************************************************************************************
ok: [51.112.229.170]

TASK [Debug Public IP] *************************************************************************************************************
ok: [51.112.229.170] => {
    "msg": "The Public IP is 51.112.229.170"
}

TASK [Restart httpd] ***************************************************************************************************************
changed: [51.112.229.170]

PLAY RECAP *************************************************************************************************************************
51.112.229.170             : ok=10   changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

- q3_http_browser.png


Not secure ~~https~~://51.112.229.170

This is Safa's Terraform environment.