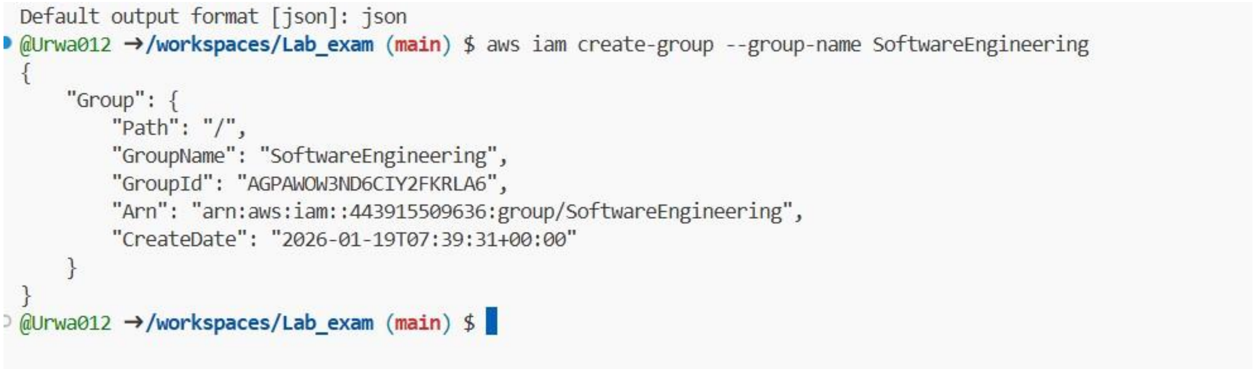
Name Urwa Zahra

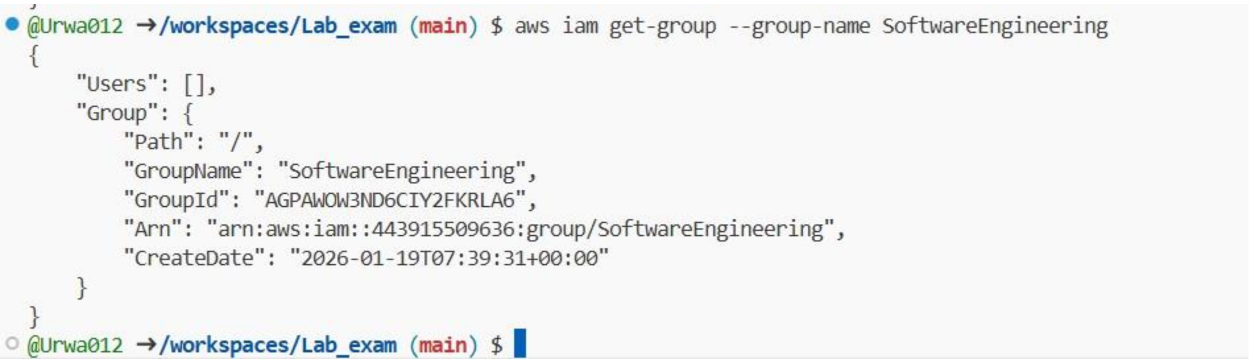
Registration#:2023-BSE-068

Section:B

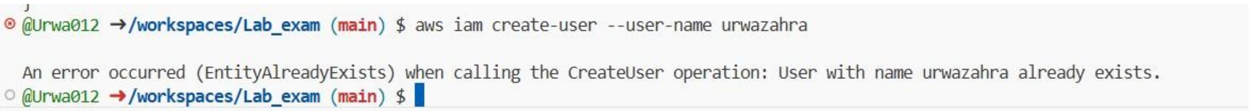
Question 1:



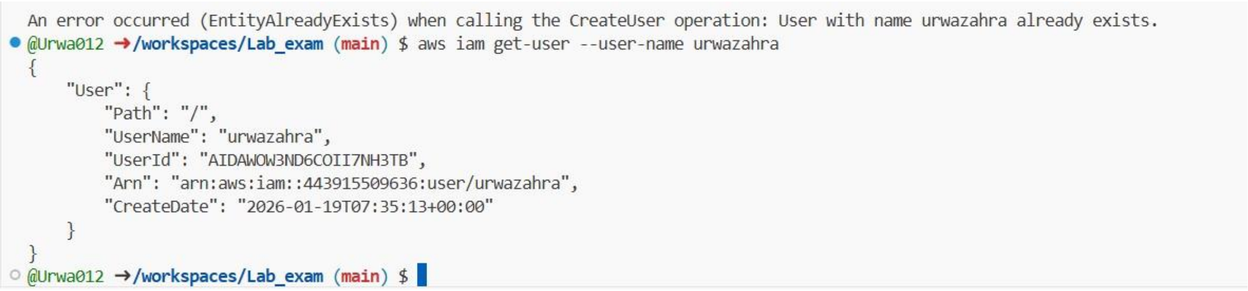
```
Default output format [json]: json
@Urwa012 →/workspaces/Lab_exam (main) $ aws iam create-group --group-name SoftwareEngineering
{
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAWOW3ND6CIY2FKRLA6",
        "Arn": "arn:aws:iam::443915509636:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:39:31+00:00"
    }
}
@Urwa012 →/workspaces/Lab_exam (main) $
```



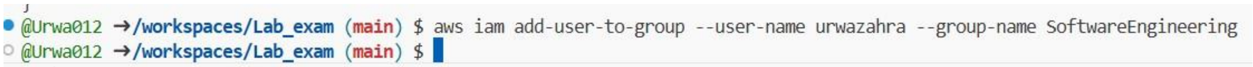
```
@Urwa012 →/workspaces/Lab_exam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAWOW3ND6CIY2FKRLA6",
        "Arn": "arn:aws:iam::443915509636:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:39:31+00:00"
    }
}
@Urwa012 →/workspaces/Lab_exam (main) $
```



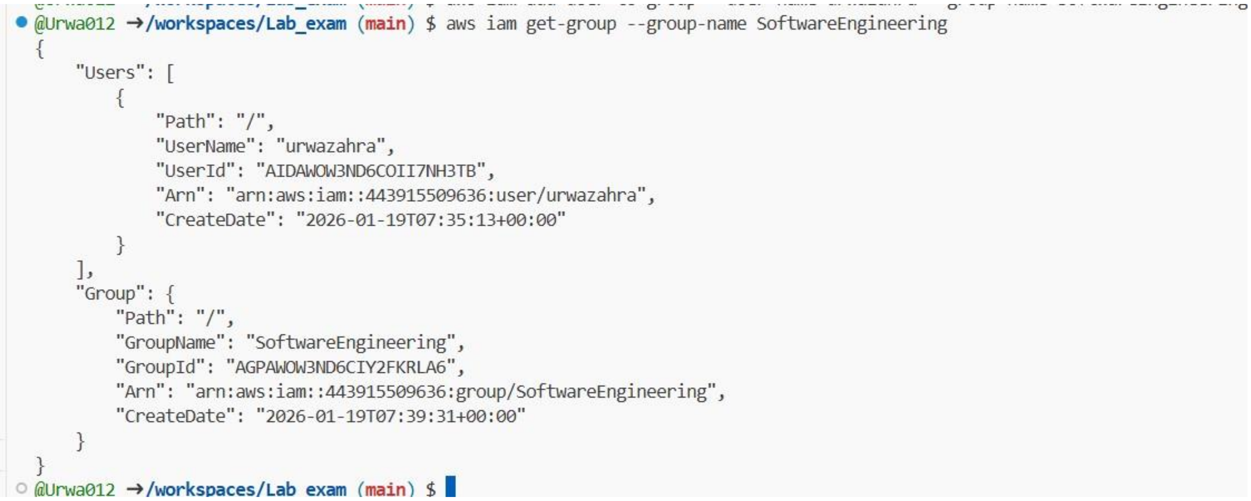
```
@Urwa012 →/workspaces/Lab_exam (main) $ aws iam create-user --user-name urwazahra

An error occurred (EntityAlreadyExists) when calling the CreateUser operation: User with name urwazahra already exists.
@Urwa012 →/workspaces/Lab_exam (main) $
```



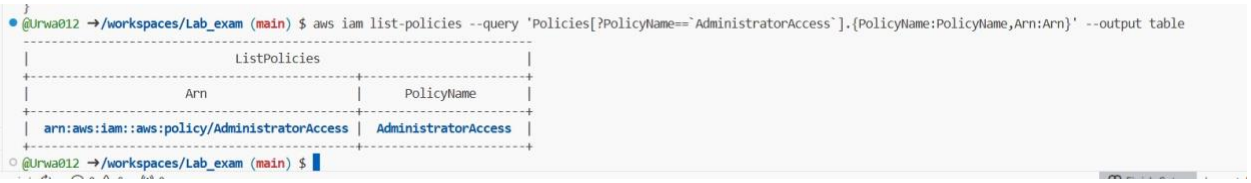
```
An error occurred (EntityAlreadyExists) when calling the CreateUser operation: User with name urwazahra already exists.
● @Urwa012 →/workspaces/Lab_exam (main) $ aws iam get-user --user-name urwazahra
{
    "User": {
        "Path": "/",
        "UserName": "urwazahra",
        "UserId": "AIDAWOW3ND6COII7NH3TB",
        "Arn": "arn:aws:iam::443915509636:user/urwazahra",
        "CreateDate": "2026-01-19T07:35:13+00:00"
    }
}
○ @Urwa012 →/workspaces/Lab_exam (main) $
```



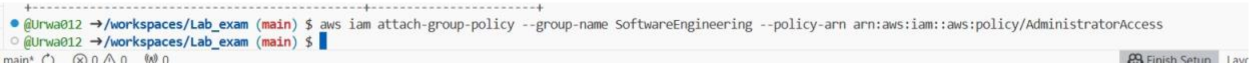
```
}
● @Urwa012 →/workspaces/Lab_exam (main) $ aws iam add-user-to-group --user-name urwazahra --group-name SoftwareEngineering
○ @Urwa012 →/workspaces/Lab_exam (main) $
```



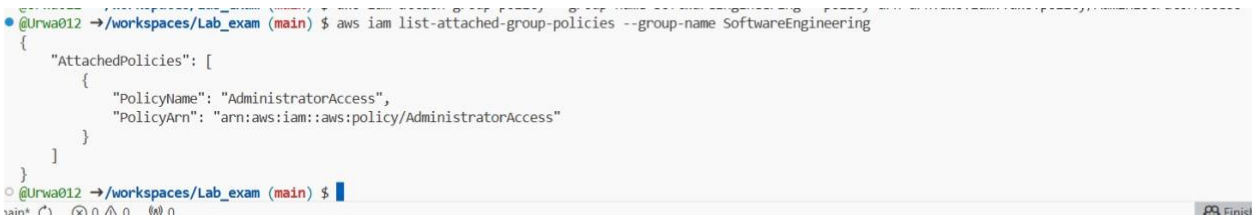
```
● @Urwa012 →/workspaces/Lab_exam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [
        {
            "Path": "/",
            "UserName": "urwazahra",
            "UserId": "AIDAWOW3ND6COII7NH3TB",
            "Arn": "arn:aws:iam::443915509636:user/urwazahra",
            "CreateDate": "2026-01-19T07:35:13+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAWOW3ND6CIY2FKRLA6",
        "Arn": "arn:aws:iam::443915509636:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:39:31+00:00"
    }
}
○ @Urwa012 →/workspaces/Lab exam (main) $
```



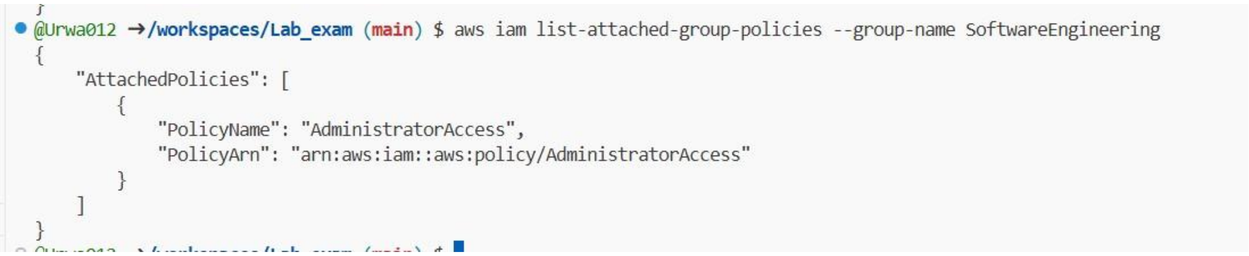
```
}
● @Urwa012 →/workspaces/Lab_exam (main) $ aws iam list-policies --query 'Policies[?PolicyName==`AdministratorAccess`].{PolicyName:PolicyName,Arn:Arn}' --output table
-------------------------------------------------------------------------
|                             ListPolicies                              |
+---------------------------------------------+-------------------------+
|                     Arn                     |       PolicyName        |
+---------------------------------------------+-------------------------+
|  arn:aws:iam::aws:policy/AdministratorAccess |  AdministratorAccess   |
+---------------------------------------------+-------------------------+
○ @Urwa012 →/workspaces/Lab_exam (main) $
```



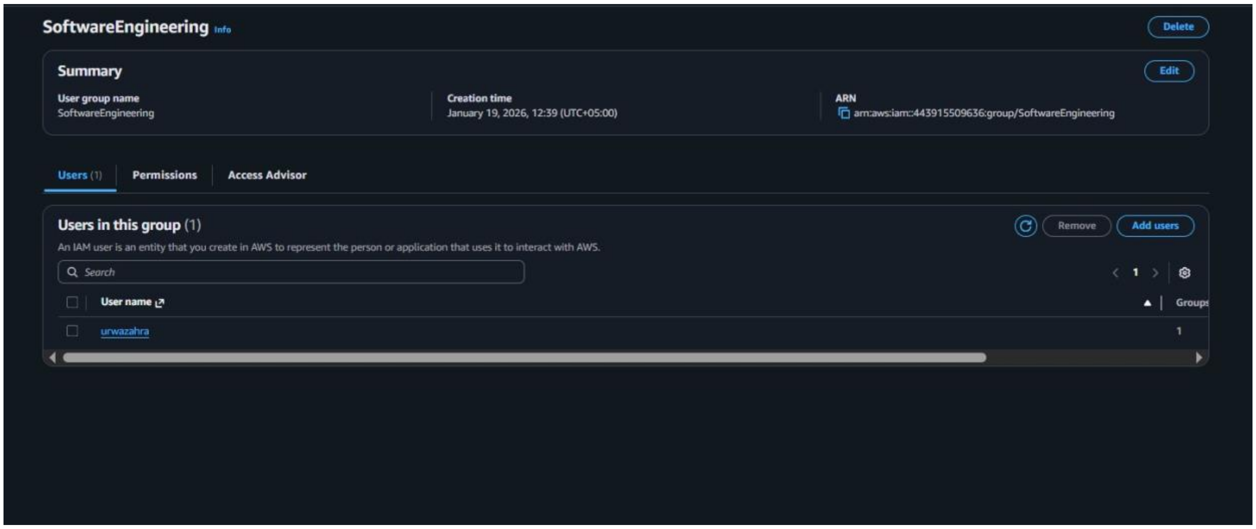
```
+----------------------------------------------+----------------------+
● @Urwa012 →/workspaces/Lab_exam (main) $ aws iam attach-group-policy --group-name SoftwareEngineering --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
○ @Urwa012 →/workspaces/Lab_exam (main) $ 
```



```
● @Urwa012 →/workspaces/Lab_exam (main) $ aws iam list-attached-group-policies --group-name SoftwareEngineering
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
○ @Urwa012 →/workspaces/Lab_exam (main) $ 
```



```
● @Urwa012 →/workspaces/Lab_exam (main) $ aws iam list-attached-group-policies --group-name SoftwareEngineering
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
```



SoftwareEngineering Info

Summary

User group name
SoftwareEngineering

Creation time
January 19, 2026, 12:39 (UTC+05:00)

ARN
arn:aws:iam::443915509636:group/SoftwareEngineering

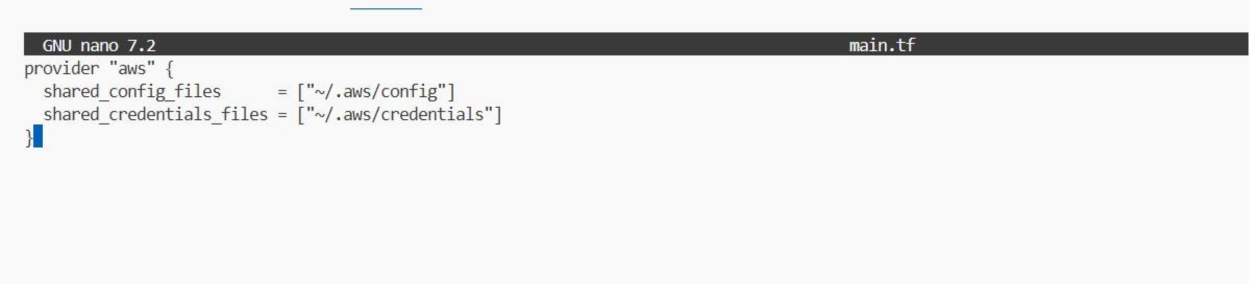
Users (1) | Permissions | Access Advisor

Users in this group (1)

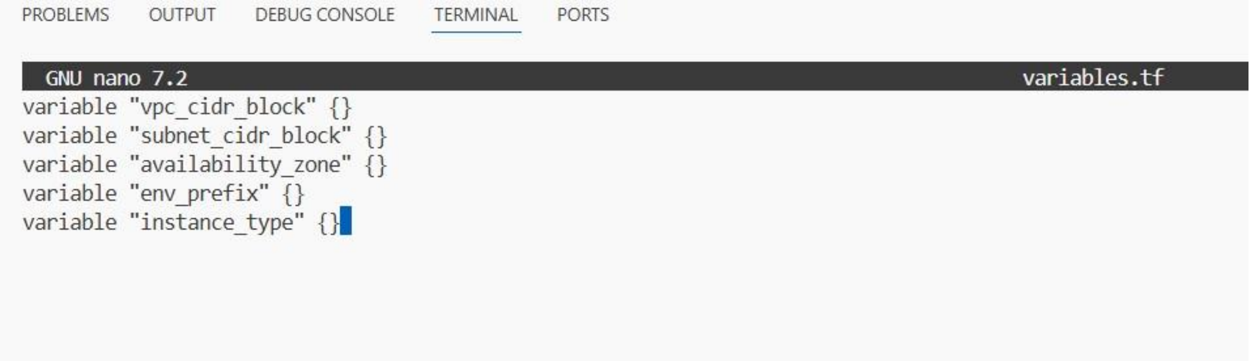
An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.

□	User name	Groups
□	urwazahra	1

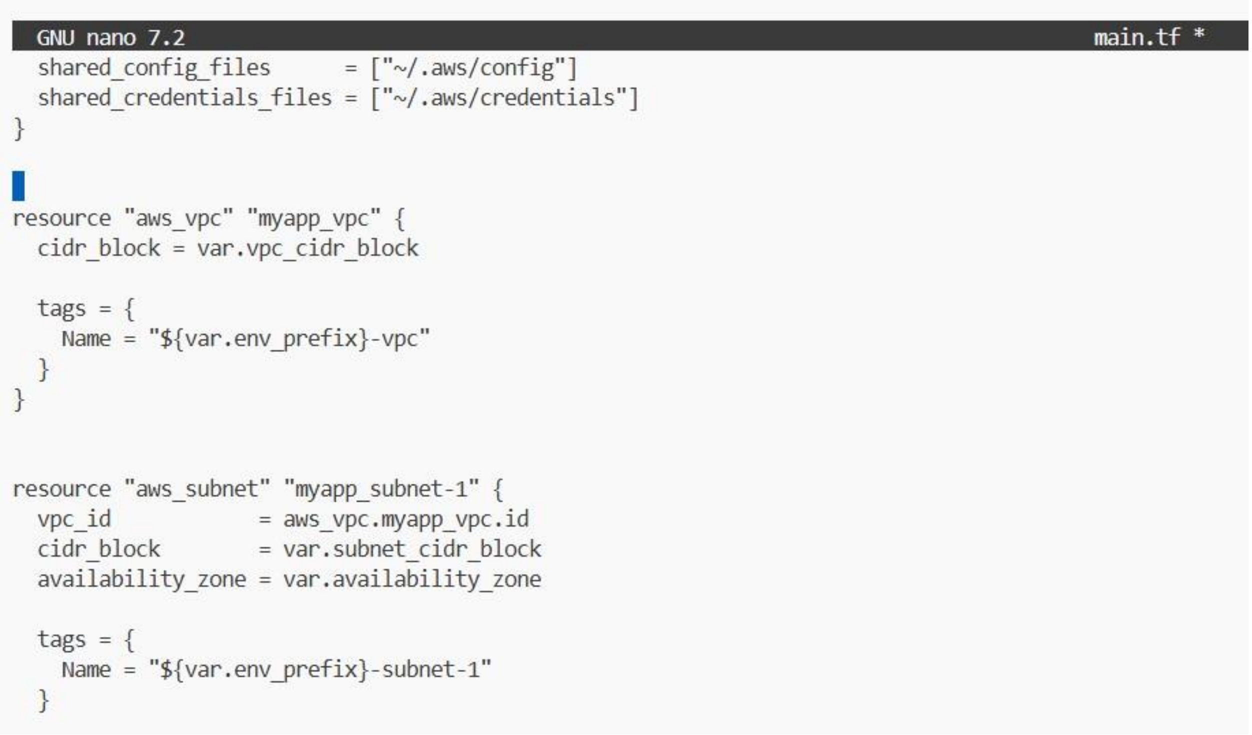
Question 2:



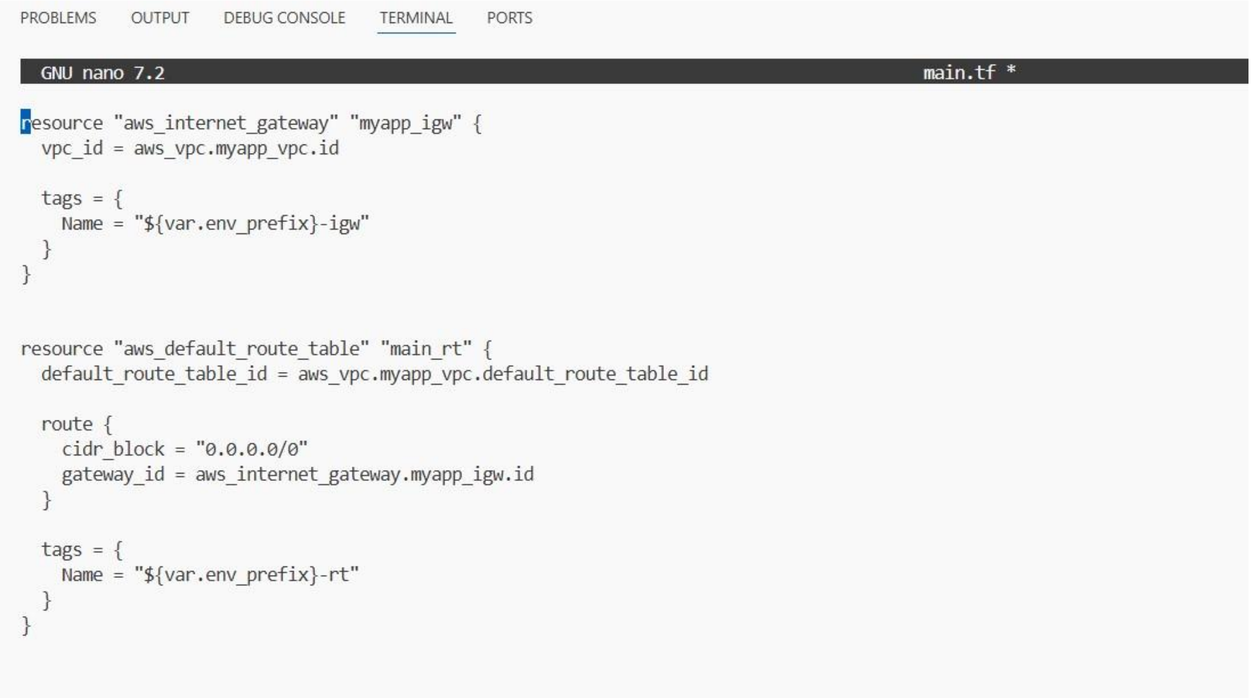
```
  GNU nano 7.2                                                   main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
```



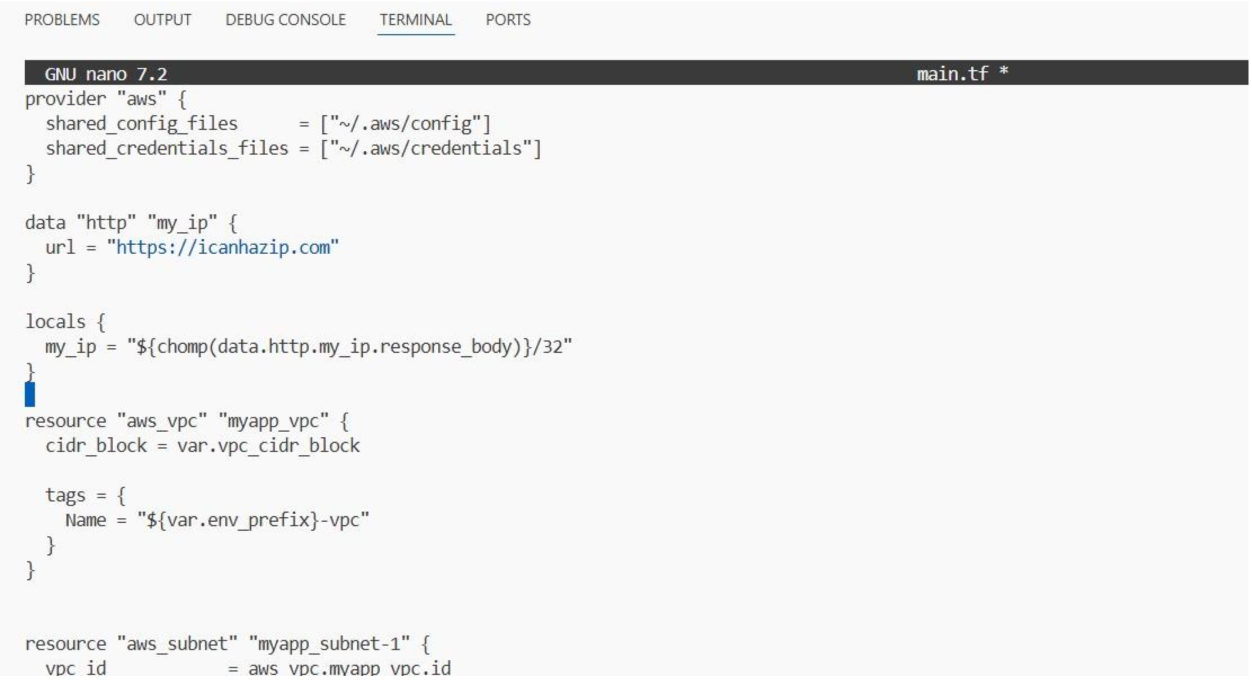
```
  GNU nano 7.2                                              variables.tf
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
```



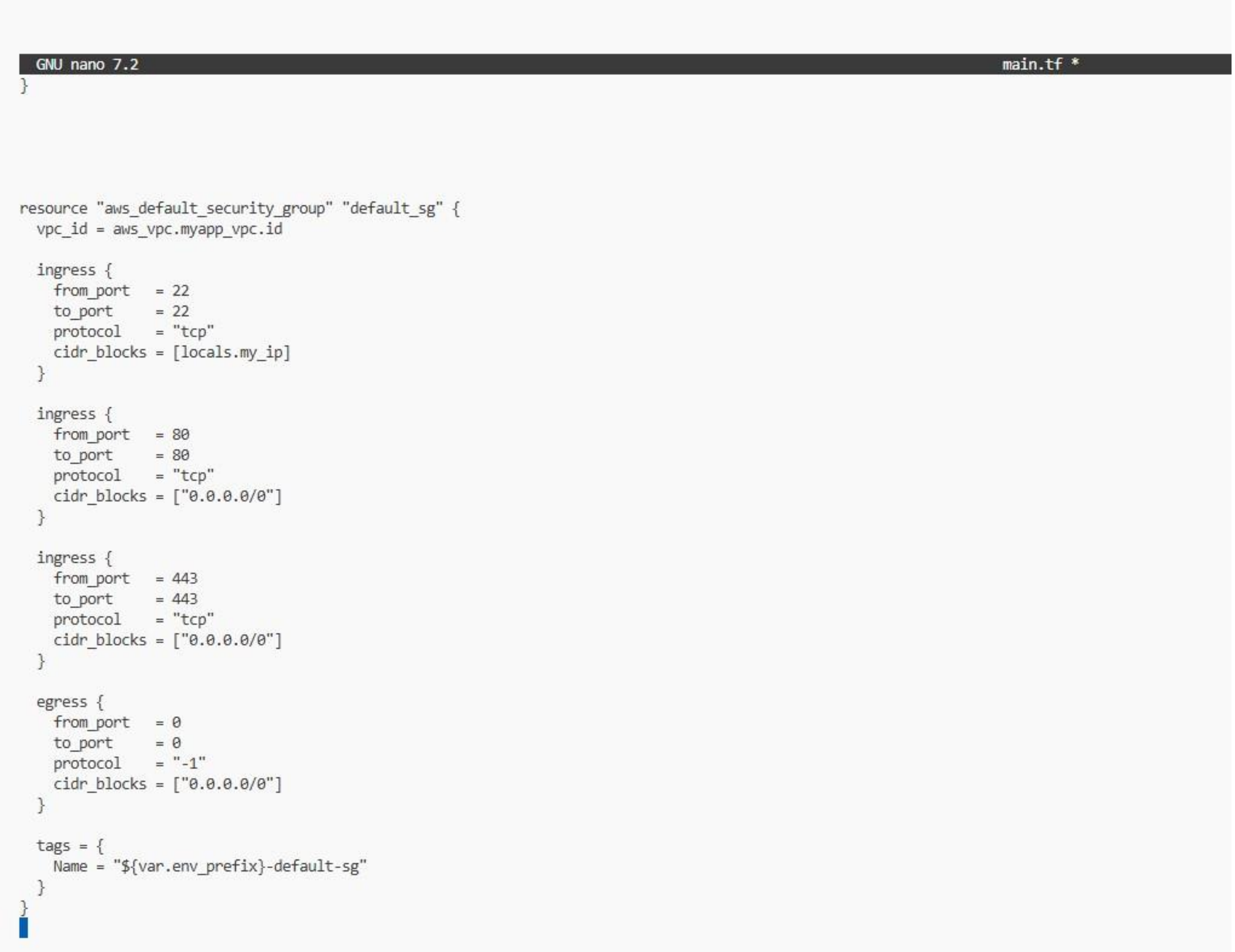
```
  GNU nano 7.2                                                main.tf *
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}


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"
  }
```



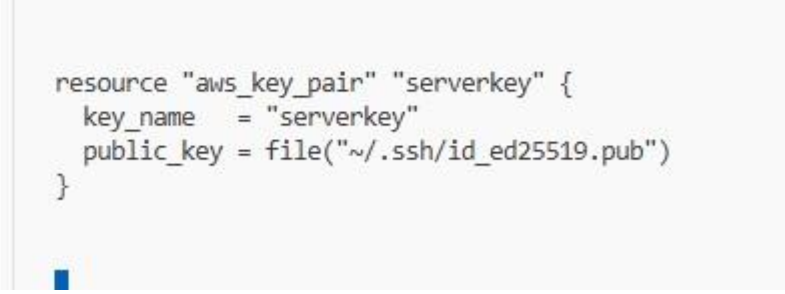
```
resource "aws_internet_gateway" "myapp_igw" {
  vpc_id = aws_vpc.myapp_vpc.id

  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"
  }
}
```



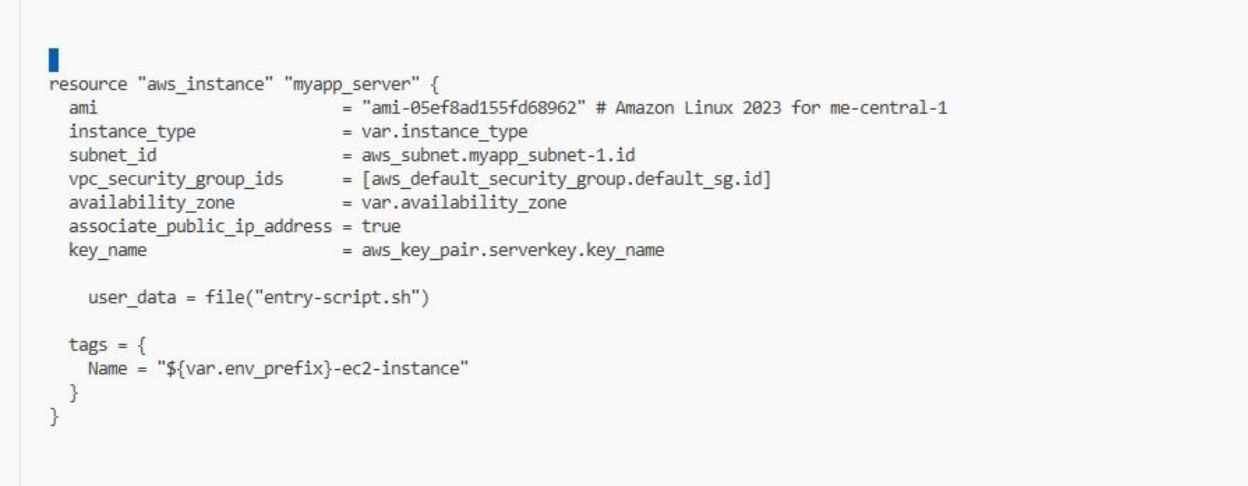
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}

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

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

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
```



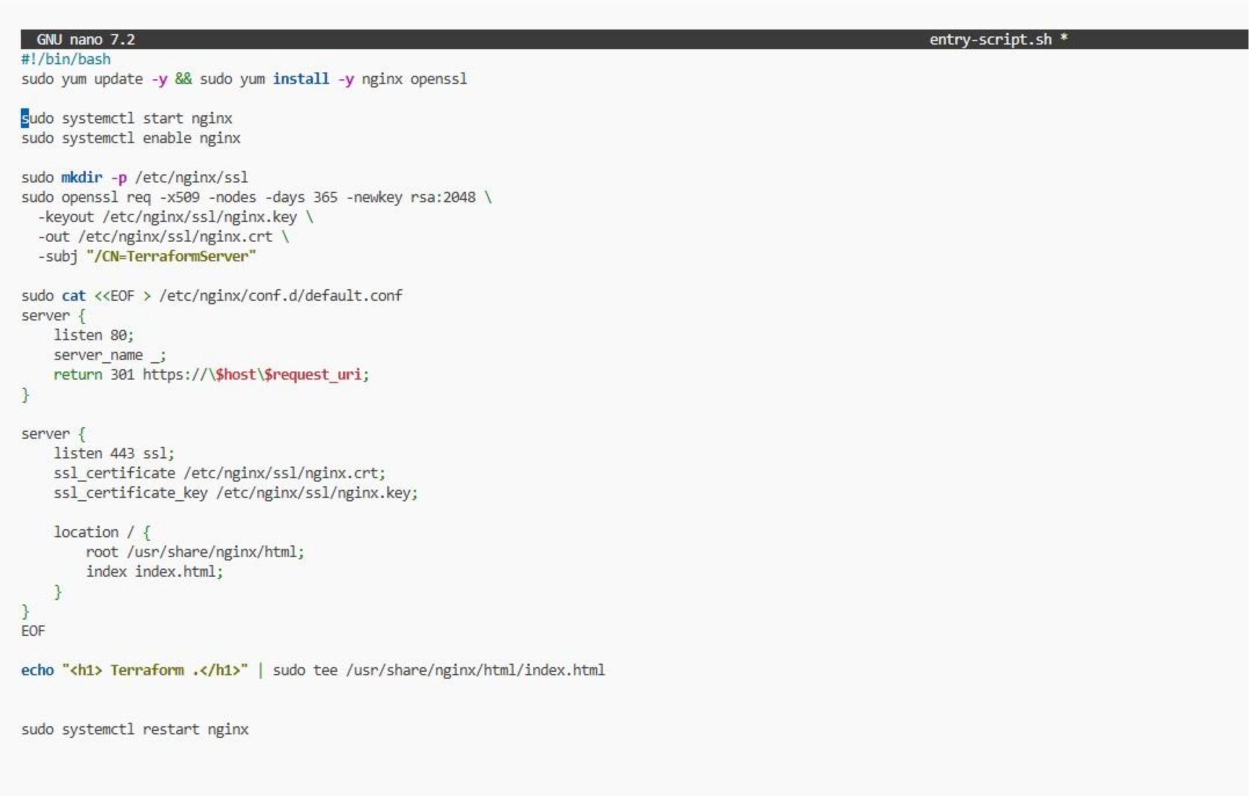
```
}



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"
  }
}
```



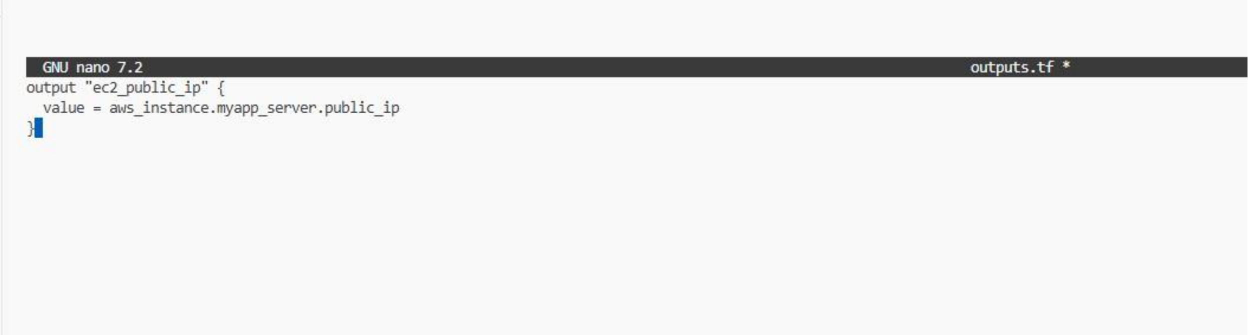
```
resource "aws_key_pair" "serverkey" {
  key_name   = "serverkey"
  public_key = file("~/.ssh/id_ed25519.pub")
}
```



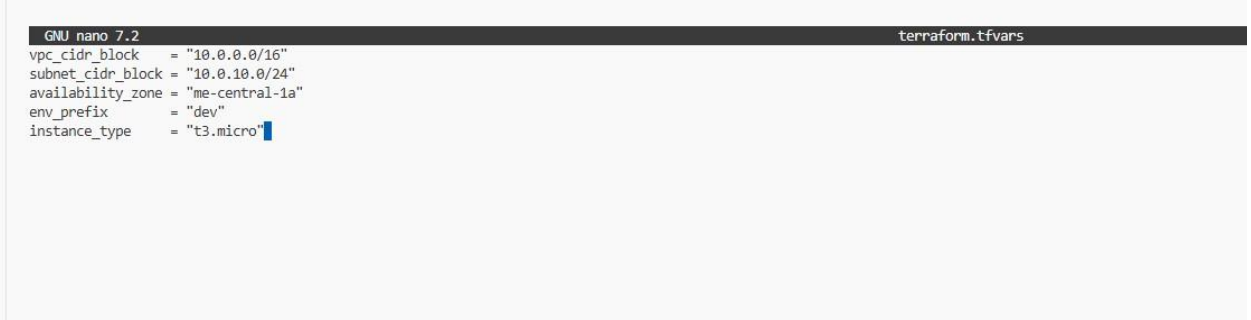
```
resource "aws_instance" "myapp_server" {
  ami                         = "ami-05ef8ad155fd68962" # Amazon Linux 2023 for me-central-1
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet-1.id
  vpc_security_group_ids      = [aws_default_security_group.default_sg.id]
  availability_zone           = var.availability_zone
  associate_public_ip_address = true
  key_name                    = aws_key_pair.serverkey.key_name

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

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



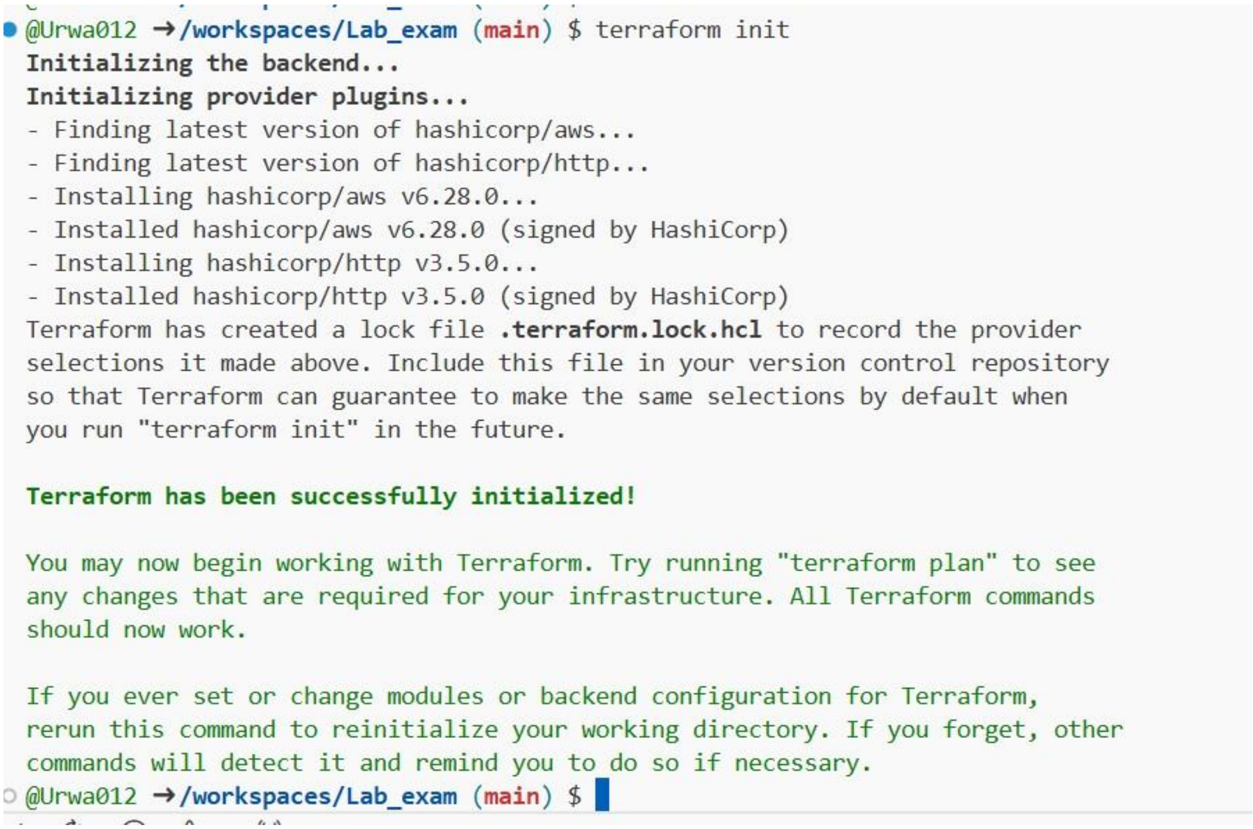
```
#!/bin/bash
sudo yum update -y && sudo yum install -y nginx openssl

sudo systemctl start nginx
sudo systemctl enable nginx

sudo mkdir -p /etc/nginx/ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/nginx/ssl/nginx.key \
  -out /etc/nginx/ssl/nginx.crt \
  -subj "/CN=TerraformServer"

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

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

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

echo "<h1> Terraform .</h1>" | sudo tee /usr/share/nginx/html/index.html


sudo systemctl restart nginx
```



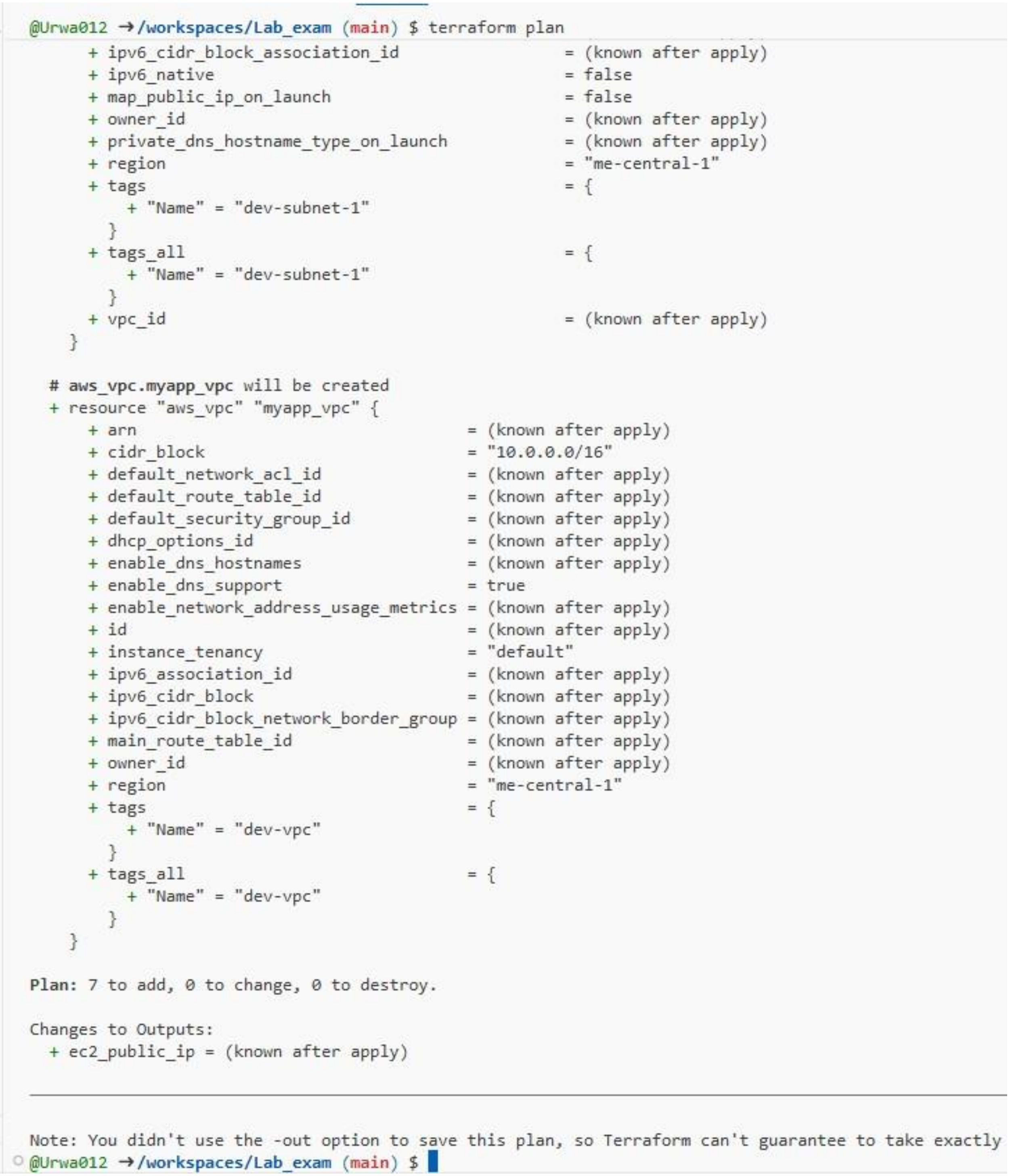
```
  GNU nano 7.2                                                                                       outputs.tf *
output "ec2_public_ip" {
  value = aws_instance.myapp_server.public_ip
}
```



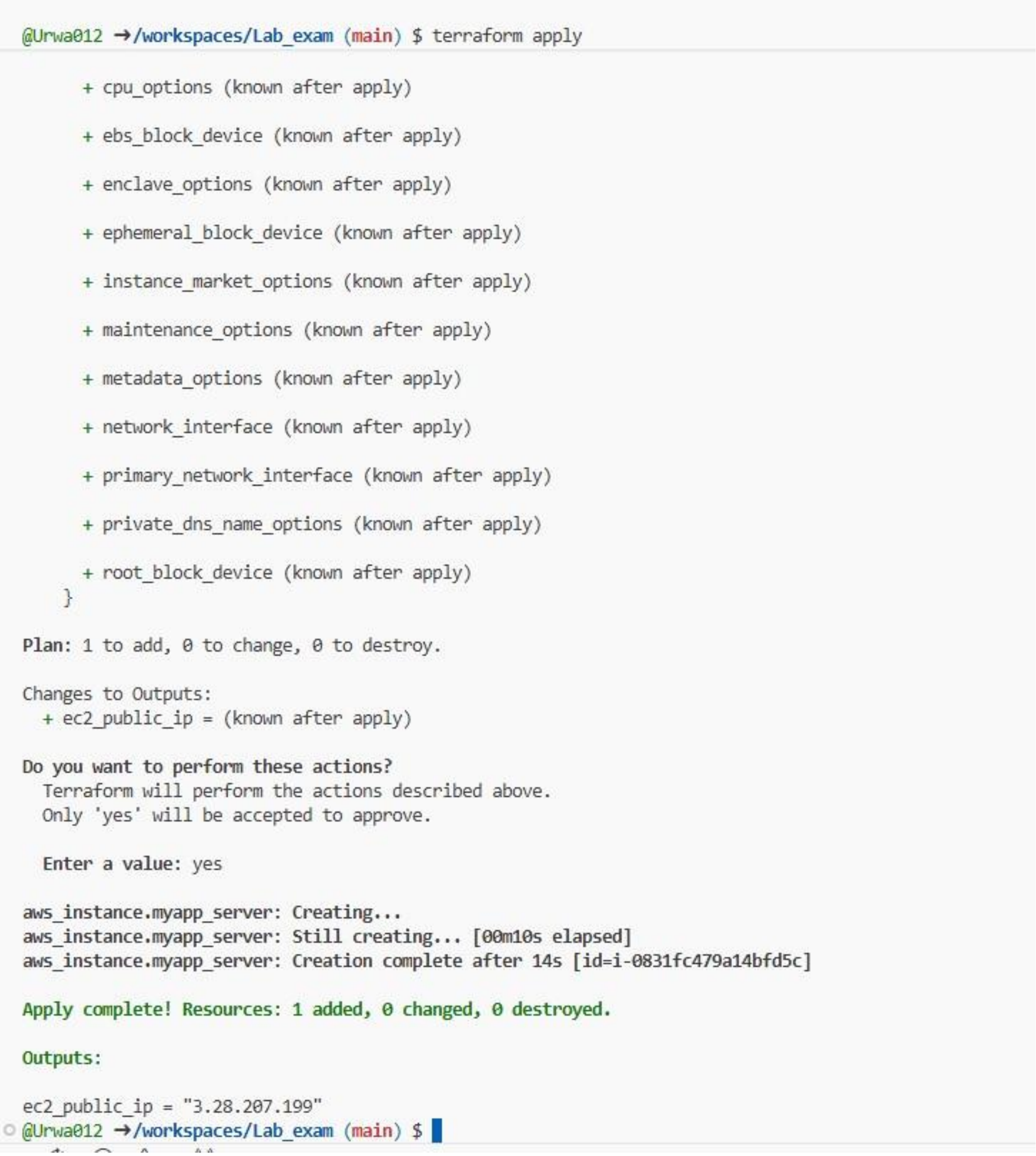
```
  GNU nano 7.2                                                                              terraform.tfvars
vpc_cidr_block     = "10.0.0.0/16"
subnet_cidr_block = "10.0.10.0/24"
availability_zone = "me-central-1a"
env_prefix         = "dev"
instance_type      = "t3.micro"
```



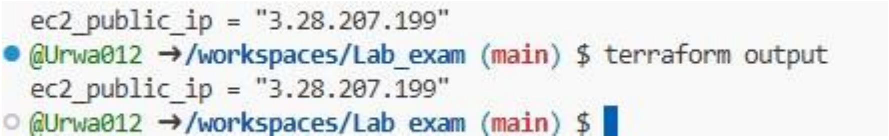
```
● @Urwa012 →/workspaces/Lab_exam (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/http...
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (signed by HashiCorp)
- 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.
○ @Urwa012 →/workspaces/Lab_exam (main) $
```



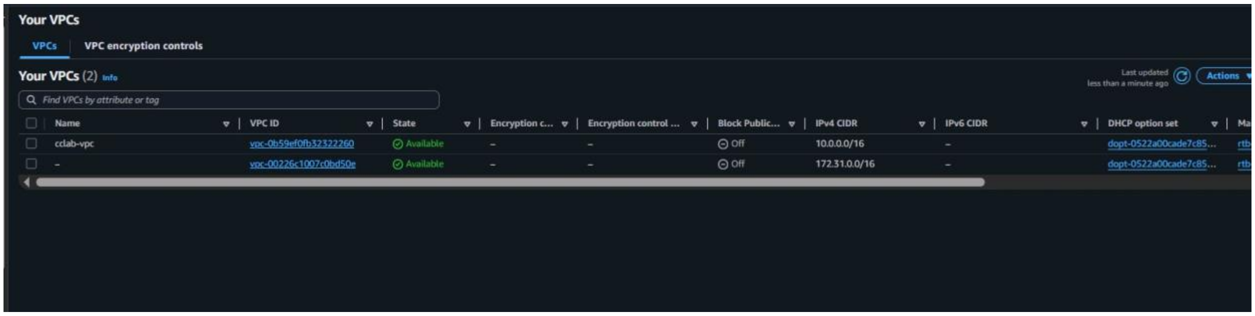
```
@Urwa012 →/workspaces/Lab_exam (main) $ terraform plan
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "me-central-1"
      + tags                                           = {
          + "Name" = "dev-subnet-1"
        }
      + tags_all                                       = {
          + "Name" = "dev-subnet-1"
        }
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.myapp_vpc will be created
  + resource "aws_vpc" "myapp_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "me-central-1"
      + tags                                 = {
          + "Name" = "dev-vpc"
        }
      + tags_all                             = {
          + "Name" = "dev-vpc"
        }
    }

Plan: 7 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + ec2_public_ip = (known after apply)

─────────────────────────────────────────────────────────────────────────────────────────────────────

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly
@Urwa012 →/workspaces/Lab_exam (main) $ 
```



```
@Urwa012 →/workspaces/Lab_exam (main) $ terraform apply

      + cpu_options (known after apply)

      + ebs_block_device (known after apply)

      + enclave_options (known after apply)

      + ephemeral_block_device (known after apply)

      + instance_market_options (known after apply)

      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + primary_network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

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

Changes to Outputs:
  + ec2_public_ip = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.myapp_server: Creating...
aws_instance.myapp_server: Still creating... [00m10s elapsed]
aws_instance.myapp_server: Creation complete after 14s [id=i-0831fc479a14bfd5c]

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

Outputs:

ec2_public_ip = "3.28.207.199"
○ @Urwa012 →/workspaces/Lab_exam (main) $
```

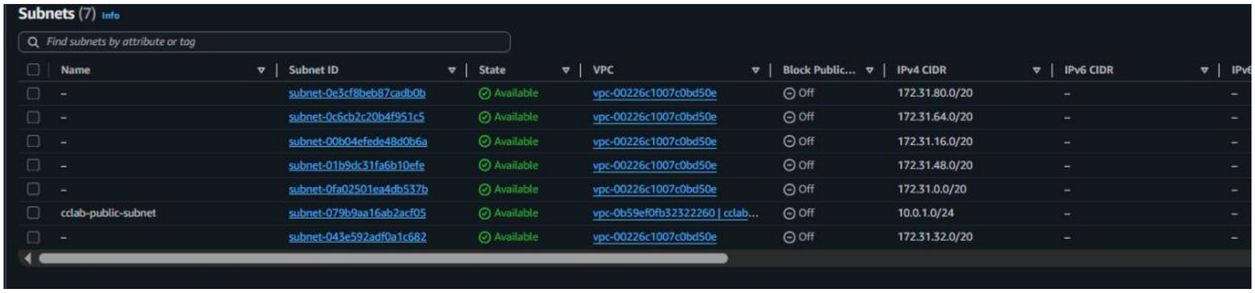


```
  ec2_public_ip = "3.28.207.199"
● @Urwa012 →/workspaces/Lab_exam (main) $ terraform output
  ec2_public_ip = "3.28.207.199"
○ @Urwa012 →/workspaces/Lab_exam (main) $
```



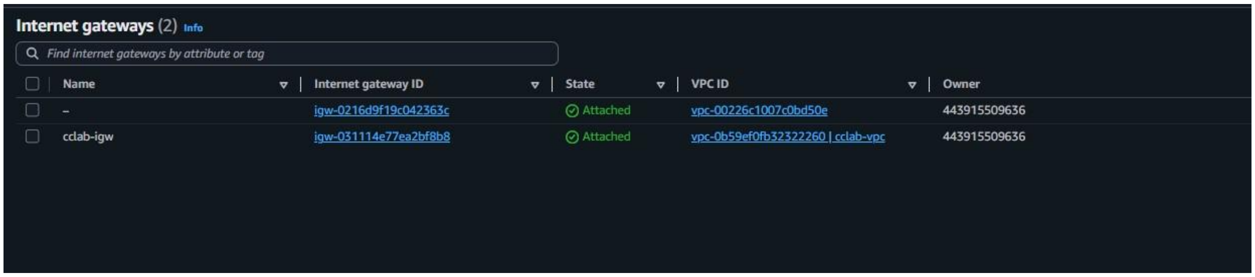
Your VPCs (2) Info

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set	Ma
cclab-vpc	vpc-0b59ef0fb32322260	Available	–	–	Off	10.0.0.0/16	–	dopt-0522a00cade7c85...	rtb
–	vpc-00226c1007c0bd50e	Available	–	–	Off	172.31.0.0/16	–	dopt-0522a00cade7c85...	rtb



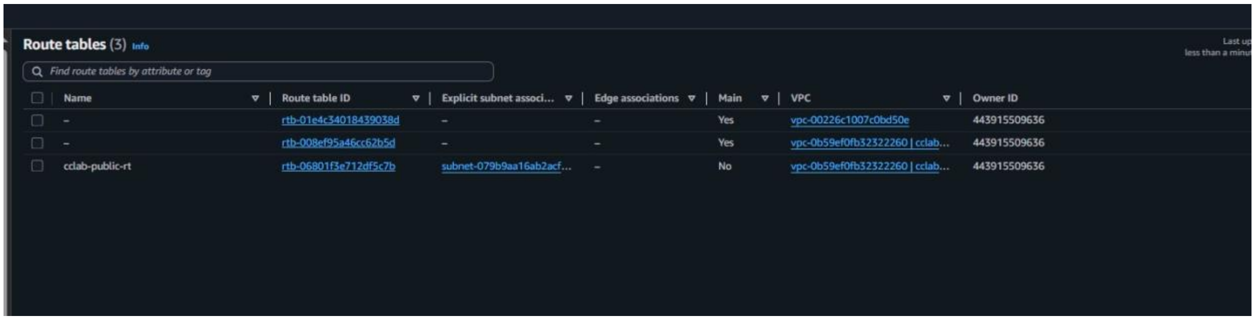
Subnets (7) Info

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR	IPv6
–	subnet-0e3cf8beb87cadb0b	Available	vpc-00226c1007c0bd50e	Off	172.31.80.0/20	–	–
–	subnet-0c6cb2c20b4f951c5	Available	vpc-00226c1007c0bd50e	Off	172.31.64.0/20	–	–
–	subnet-00b04efede48d0b6a	Available	vpc-00226c1007c0bd50e	Off	172.31.16.0/20	–	–
–	subnet-01b9dc31fa6b10efe	Available	vpc-00226c1007c0bd50e	Off	172.31.48.0/20	–	–
–	subnet-0fa02501ea4db537b	Available	vpc-00226c1007c0bd50e	Off	172.31.0.0/20	–	–
cclab-public-subnet	subnet-079b9aa16ab2acf05	Available	vpc-0b59ef0fb32322260 | cclab...	Off	10.0.1.0/24	–	–
–	subnet-043e592adf0a1c682	Available	vpc-00226c1007c0bd50e	Off	172.31.32.0/20	–	–



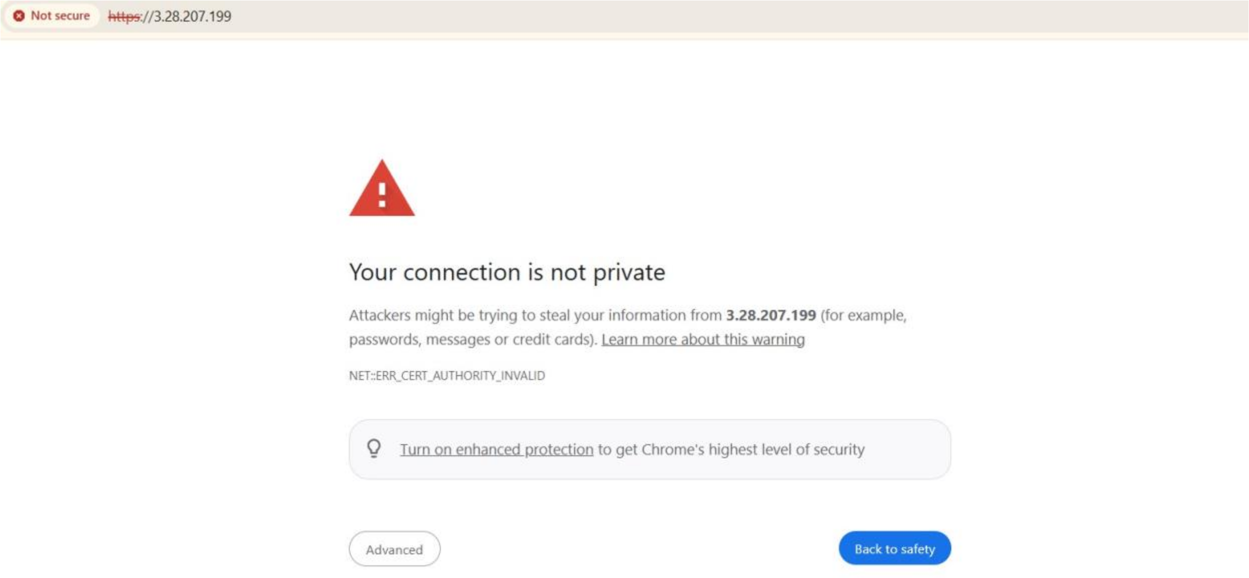
Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-0216d9f19c042363c	Attached	vpc-00226c1007c0bd50e	443915509636
cclab-igw	igw-031114e77ea2bf8b8	Attached	vpc-0b59ef0fb32322260 | cclab-vpc	443915509636



Route tables (3) Info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
–	rtb-01e4c34018439038d	–	–	Yes	vpc-00226c1007c0bd50e	443915509636
–	rtb-008ef95a46cc62b5d	–	–	Yes	vpc-0b59ef0fb32322260 | cclab...	443915509636
cclab-public-rt	rtb-06801f3e712df5c7b	subnet-079b9aa16ab2acf...	–	No	vpc-0b59ef0fb32322260 | cclab...	443915509636

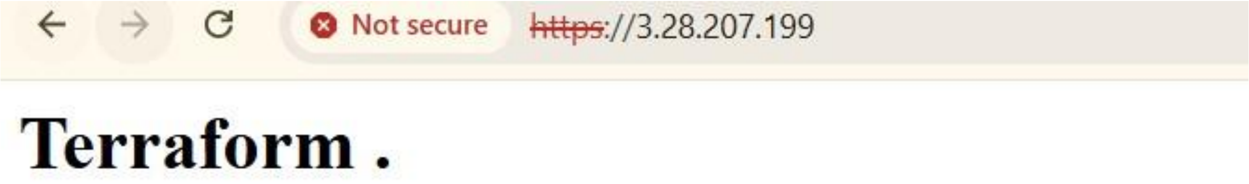


Your connection is not private

Attackers might be trying to steal your information from **3.28.207.199** (for example, passwords, messages or credit cards). Learn more about this warning

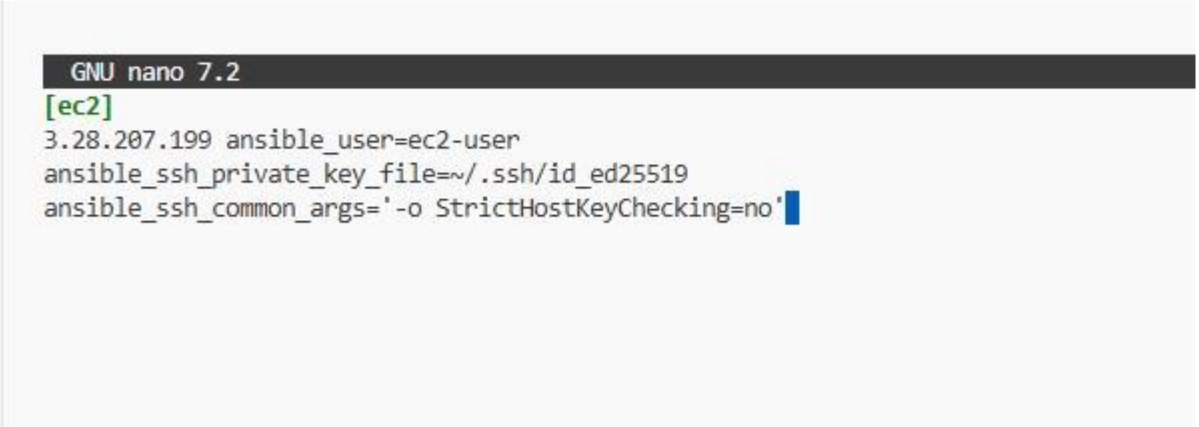
NET::ERR_CERT_AUTHORITY_INVALID

Turn on enhanced protection to get Chrome's highest level of security

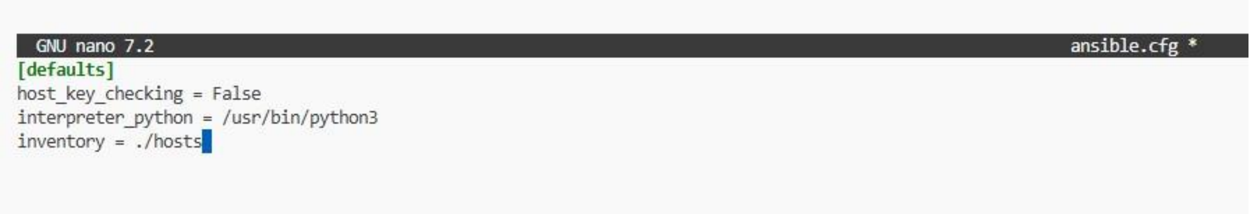


Terraform .

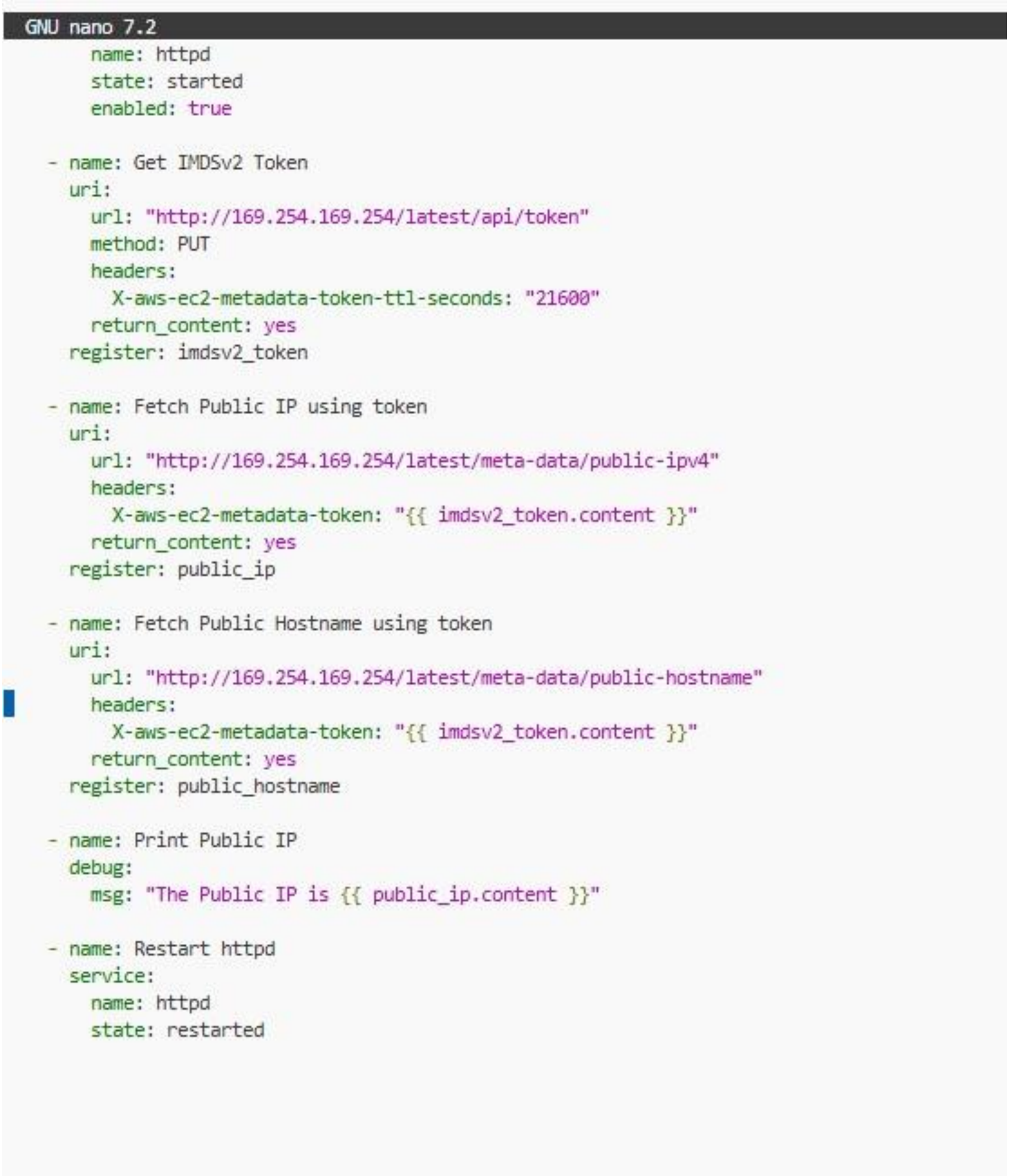
Question 3:



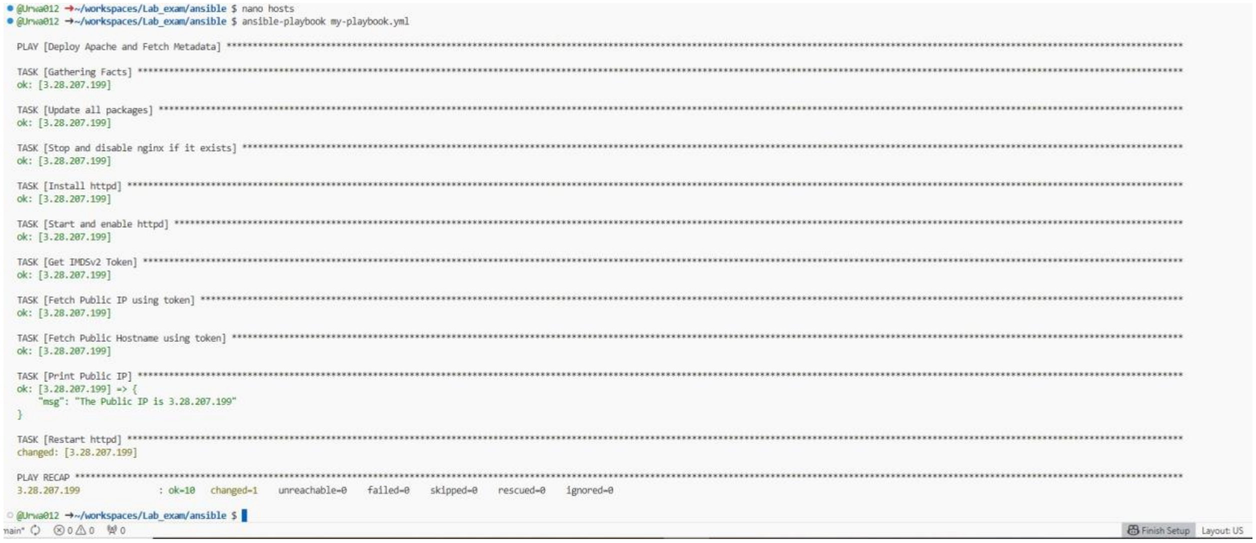
```
[ec2]
3.28.207.199 ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
ansible_ssh_common_args='-o StrictHostKeyChecking=no'
```



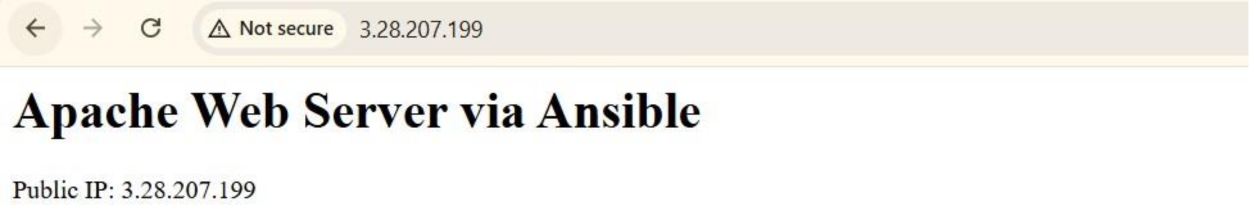
```
[defaults]
host_key_checking = False
interpreter_python = /usr/bin/python3
inventory = ./hosts
```



```
GNU nano 7.2
        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: yes
      register: imdsv2_token

    - name: Fetch Public IP using token
      uri:
        url: "http://169.254.169.254/latest/meta-data/public-ipv4"
        headers:
          X-aws-ec2-metadata-token: "{{ imdsv2_token.content }}"
        return_content: yes
      register: public_ip

    - name: Fetch Public Hostname using token
      uri:
        url: "http://169.254.169.254/latest/meta-data/public-hostname"
        headers:
          X-aws-ec2-metadata-token: "{{ imdsv2_token.content }}"
        return_content: yes
      register: public_hostname

    - name: Print Public IP
      debug:
        msg: "The Public IP is {{ public_ip.content }}"

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

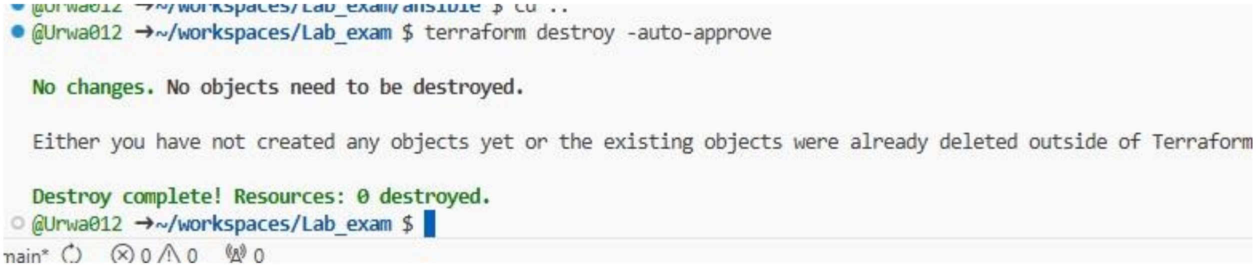


```
● @Urwa012 →~/workspaces/Lab_exam/ansible $ nano hosts
● @Urwa012 →~/workspaces/Lab_exam/ansible $ ansible-playbook my-playbook.yml

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

  TASK [Gathering Facts] ****************************************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Update all packages] ************************************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Stop and disable nginx if it exists] ********************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Install httpd] ******************************************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Start and enable httpd] *********************************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Get IMDSv2 Token] ***************************************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Fetch Public IP using token] ****************************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Fetch Public Hostname using token] **********************************************************************************************************************************************************
  ok: [3.28.207.199]

  TASK [Print Public IP] ****************************************************************************************************************************************************************************
  ok: [3.28.207.199] => {
      "msg": "The Public IP is 3.28.207.199"
  }

  TASK [Restart httpd] ******************************************************************************************************************************************************************************
  changed: [3.28.207.199]

  PLAY RECAP ****************************************************************************************************************************************************************************************
  3.28.207.199               : ok=10   changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

○ @Urwa012 →~/workspaces/Lab_exam/ansible $ 
```



Apache Web Server via Ansible

Public IP: 3.28.207.199



```
● @Urwa012 →~/workspaces/Lab_exam/ansible $ cd ..
● @Urwa012 →~/workspaces/Lab_exam $ terraform destroy -auto-approve

  No changes. No objects need to be destroyed.

  Either you have not created any objects yet or the existing objects were already deleted outside of Terraform

  Destroy complete! Resources: 0 destroyed.
○ @Urwa012 →~/workspaces/Lab_exam $ 
```