MUSFIRA FAROOQ

2023-BSE-045

CLOUD COMPUTING FINAL EXAM

Q1 – AWS IAM Setup Using AWS CLI and Console Verification

```
@Musfira-0514 →/workspaces/exam (main) $ aws iam create-group --group-name SoftwareEngineering
{
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAVSVUK5OYG535GBBHJ",
        "Arn": "arn:aws:iam::383704034224:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:35:40+00:00"
    }
}
@Musfira-0514 →/workspaces/exam (main) $
```

```
@Musfira-0514 →/workspaces/exam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAVSVUK5OYG535GBBHJ",
        "Arn": "arn:aws:iam::383704034224:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:35:40+00:00"
    }
}
@Musfira-0514 →/workspaces/exam (main) $
```

```
@Musfira-0514 ➜/workspaces/exam (main) $ aws iam create-user --user-name Musfira
{
    "User": {
        "Path": "/",
        "UserName": "Musfira",
        "UserId": "AIDAVSVUK5OYGK4IQUN53",
        "Arn": "arn:aws:iam::383704034224:user/Musfira",
        "CreateDate": "2026-01-19T07:37:15+00:00"
    }
}
@Musfira-0514 ➜/workspaces/exam (main) $
```

```
@Musfira-0514 ➜/workspaces/exam (main) $ aws iam get-user --user-name Musfira
{
    "User": {
        "Path": "/",
        "UserName": "Musfira",
        "UserId": "AIDAVSVUK5OYGK4IQUN53",
        "Arn": "arn:aws:iam::383704034224:user/Musfira",
        "CreateDate": "2026-01-19T07:37:15+00:00"
    }
}
```

```
@Musfira-0514 ➜/workspaces/exam (main) $ aws iam add-user-to-group --user-name Musfira --group-name SoftwareEngineering
@Musfira-0514 ➜/workspaces/exam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Musfira",
            "UserId": "AIDAVSVUK5OYGK4IQUN53",
            "Arn": "arn:aws:iam::383704034224:user/Musfira",
            "CreateDate": "2026-01-19T07:37:15+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAVSVUK5OYG535GBBHJ",
        "Arn": "arn:aws:iam::383704034224:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:35:40+00:00"
    }
}
@Musfira-0514 ➜/workspaces/exam (main) $
```

```
@Musfira-0514 ➜ /workspaces/exam (main) $ aws iam list-policies --scope AWS --query "Policies[?PolicyName=='AdministratorAccess']"
[
    {
        "PolicyName": "AdministratorAccess",
        "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
        "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 2,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2015-02-06T18:39:46+00:00",
        "UpdateDate": "2015-02-06T18:39:46+00:00"
    }
]
@Musfira-0514 ➜ /workspaces/exam (main) $
```

```
@Musfira-0514 ➜ /workspaces/exam (main) $ aws iam attach-group-policy \
    --group-name SoftwareEngineering \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

```
@Musfira-0514 ➜ /workspaces/exam (main) $ aws iam list-attached-group-policies --group-name SoftwareEngineering
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
@Musfira-0514 ➜ /workspaces/exam (main) $
```

User groups (1) Info

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

Delete | Create group

Search

1

Group name	Users	Permissions	Creation time
SoftwareEngineering	1	Defined	20 minutes ago

Users (3) Info

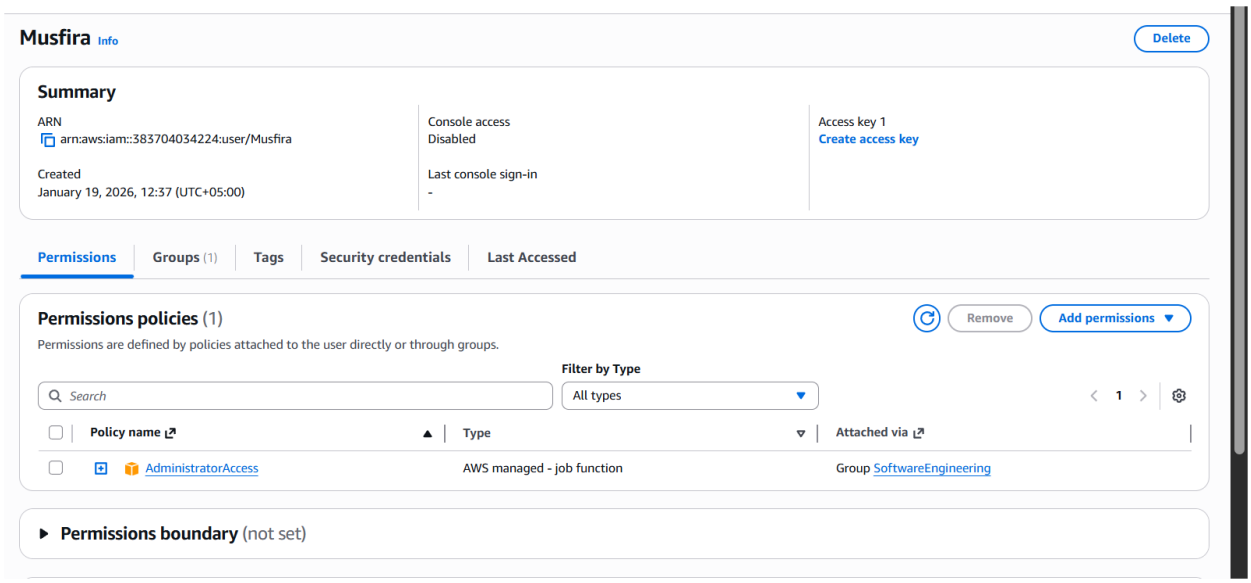
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

Delete | Create user

Search

1

User name	Path	Groups	Last activity	MFA	Password age	Console last sign-in	Access key ID
ansibe-user	/	0	Yesterday	-	12 days	-	Active - AKIAVSVUK5O...
exam	/	0	9 minutes ago	-	19 hours	-	Active - AKIAVSVUK5O...
Musfira	/	1	-	-	-	-	-



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

```
@Musfira-0514 →/workspaces/exam (main) $ touch main.tf
@Musfira-0514 →/workspaces/exam (main) $ nano main.tf
@Musfira-0514 →/workspaces/exam (main) $ cat main.tf
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
@Musfira-0514 →/workspaces/exam (main) $
```

```
  GNU nano 7.2
provider "aws" {
  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"
  }
}
```

```
@Musfira-0514 →/workspaces/exam (main) $ cat main.tf

resource "aws_subnet" "myapp_subnet" {
  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"
  }
}
```

```
@Musfira-0514 →/workspaces/exam (main) $ cat main.tf

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

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

```
@Musfira-0514 →/workspaces/exam (main) $ cat main.tf
resource "aws_default_route_table" "default_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"
  }
}
```

```
@Musfira-0514 ➜/workspaces/exam (main) $ cat main.tf
resource "aws_default_security_group" "default_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

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

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

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

```
@Musfira-0514 →/workspaces/exam (main) $ cat main.tf
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

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

```
 @Musfira-0514 →/workspaces/exam (main) $ cat main.tf

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

```
@Musfira-0514 →/workspaces/exam (main) $ cat main.tf
}

resource "aws_instance" "myapp_ec2" {
  ami                         = "ami-0d593311db5abb72b"
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet.id
  availability_zone           = var.availability_zone
  vpc_security_group_ids      = [aws_default_security_group.default_sg.id]
  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"
  }
}
@Musfira-0514 →/workspaces/exam (main) $
```

```
@Musfira-0514 →/workspaces/exam (main) $ cat 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/selfsigned.key \
  -out /etc/nginx/ssl/selfsigned.crt \
  -subj "/C=PK/ST=Islamabad/L=Islamabad/O=Terraform/CN=localhost"

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

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

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

cat <<EOF > /usr/share/nginx/html/index.html
<html>
  <body>
```

```
cat <<EOF > /usr/share/nginx/html/index.html
<html>
  <body>
    <h1>This is Musfira Farooq's Terraform environment.</h1>
  </body>
</html>
EOF

systemctl enable nginx
systemctl restart nginx
@Musfira-0514 ➜ /workspaces/exam (main) $
```

```
@Musfira-0514 ➜ /workspaces/exam (main) $ touch outputs.tf
@Musfira-0514 ➜ /workspaces/exam (main) $ nano outputs.tf
@Musfira-0514 ➜ /workspaces/exam (main) $ cat outputs.tf
output "aws_instance_public_ip" {
  value = aws_instance.myapp-server.public_ip
}
@Musfira-0514 ➜ /workspaces/exam (main) $
```

```
}
@Musfira-0514 ➜ /workspaces/exam (main) $ touch locals.tf
@Musfira-0514 ➜ /workspaces/exam (main) $ nano locals.tf
@Musfira-0514 ➜ /workspaces/exam (main) $ cat locals.tf
locals {
  my_ip = "${chomp(data.http.my_ip.response_body)}/32"
}

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

@Musfira-0514 ➜ /workspaces/exam (main) $
```

```
@Musfira-0514 ➜ /workspaces/exam (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/http...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v6.28.0...
- Installed hashicorp/aws v6.28.0 (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.
@Musfira-0514 ➜ /workspaces/exam (main) $
```

```
@Musfira-0514 ➜ /workspaces/exam (main) $ ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519 -N
Generating public/private ed25519 key pair.
Created directory '/home/codespace/.ssh'.
Your identification has been saved in /home/codespace/.ssh/id_ed25519
Your public key has been saved in /home/codespace/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Dx8LWcWErJkiqLMocMlEG41JgrVtfagTiZ1XOK9Wvjw codespace@codespaces-62eff3
The key's randomart image is:
+--[ED25519 256]--+
|ooo+   ... +o    |
|..==.+oo  o..    |
| .o+B +o.+.      |
|  +..+..*o       |
| + .o. =S .      |
|+ +  .o .= o     |
|o+   . . .+      |
|+       E        |
|.        .       |
+----[SHA256]-----+
@Musfira-0514 ➜ /workspaces/exam (main) $
```

```
@Musfira-0514 ➜ /workspaces/exam (main) $ ls ~/.ssh
id_ed25519  id_ed25519.pub
@Musfira-0514 ➜ /workspaces/exam (main) $
```

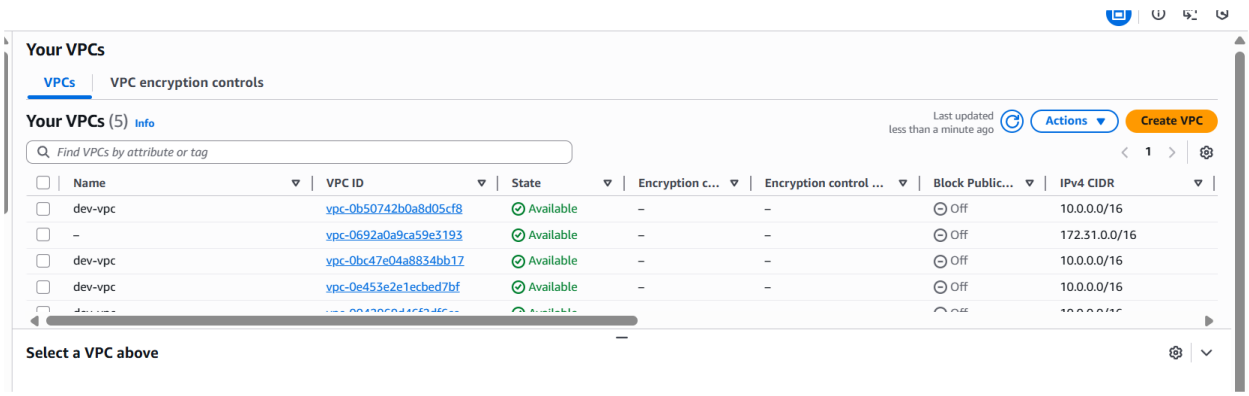
```
@Musfira-0514 →/workspaces/exam (main) $ terraform validate
Success! The configuration is valid.
```

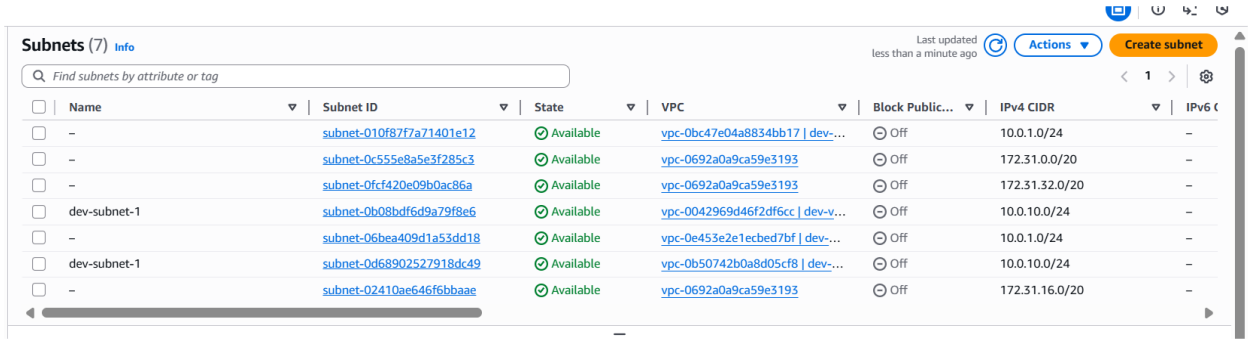
```
@Musfira-0514 →/workspaces/exam (main) $ terraform plan
data.http.my_ip: Reading...
data.http.my_ip: Read complete after 0s [id=https://icanhazip.com]

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.default_rt will be created
  + resource "aws_default_route_table" "default_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                  = [
          + {
              + cidr_block                = "0.0.0.0/0"
              + gateway_id                = (known after apply)
                # (10 unchanged attributes hidden)
            },
        ]
      + tags                   = {
          + "Name" = "dev-rt"
        }
      + tags_all               = {
          + "Name" = "dev-rt"
        }
      + vpc_id                 = (known after apply)
    }

  # aws_default_security_group.default_sg will be created
```

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

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

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

Note: You didn't use the -out option to save this plan, so Terraform
@Musfira-0514 →/workspaces/exam (main) $
```

```
@Musfira-0514 ➜/workspaces/exam (main) $ terraform 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:
  + aws_instance_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_ec2: Creating...
aws_instance.myapp_ec2: Still creating... [00m10s elapsed]
aws_instance.myapp_ec2: Creation complete after 14s [id=i-0ea4908d256b16c09]

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

Outputs:

aws_instance_public_ip = "158.252.83.160"
@Musfira-0514 ➜/workspaces/exam (main) $
```

```
@Musfira-0514 ➜/workspaces/exam (main) $ terraform output
aws_instance_public_ip = "158.252.83.160"
@Musfira-0514 ➜/workspaces/exam (main) $
```



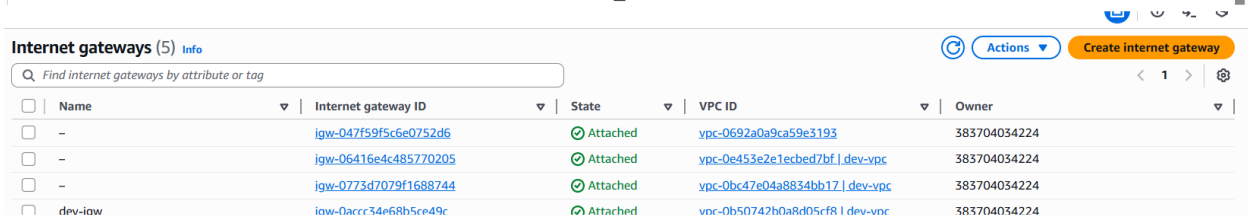


Subnets (7) Info

Last updated less than a minute ago | Actions | Create subnet

Find subnets by attribute or tag

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 C
–	subnet-010f87f7a71401e12	Available	vpc-0bc47e04a8834bb17 | dev-...	Off	10.0.1.0/24	–
–	subnet-0c555e8a5e3f285c3	Available	vpc-0692a0a9ca59e3193	Off	172.31.0.0/20	–
–	subnet-0fcf420e09b0ac86a	Available	vpc-0692a0a9ca59e3193	Off	172.31.32.0/20	–
dev-subnet-1	subnet-0b08bdf6d9a79f8e6	Available	vpc-0042969d46f2df6cc | dev-v...	Off	10.0.10.0/24	–
–	subnet-06bea409d1a53dd18	Available	vpc-0e453e2e1ecbed7bf | dev-...	Off	10.0.1.0/24	–
dev-subnet-1	subnet-0d68902527918dc49	Available	vpc-0b50742b0a8d05cf8 | dev-...	Off	10.0.10.0/24	–
–	subnet-02410ae646f6bbaae	Available	vpc-0692a0a9ca59e3193	Off	172.31.16.0/20	–

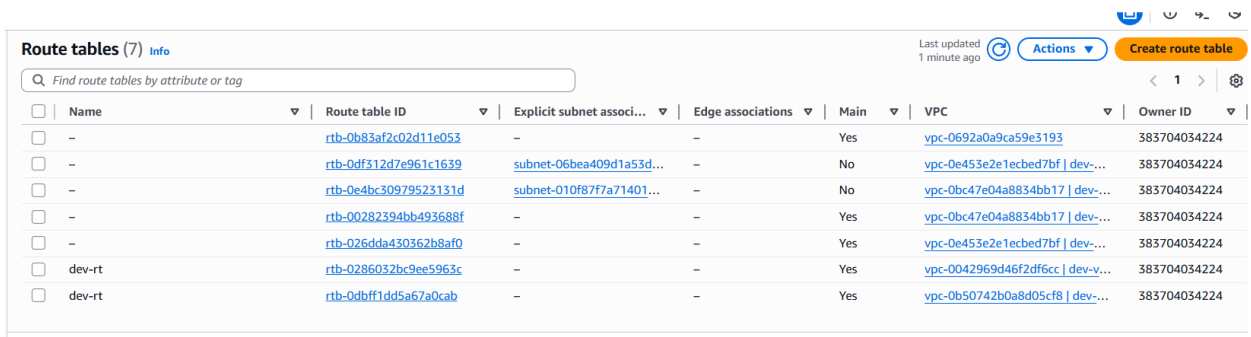


Internet gateways (5) Info

Actions | Create internet gateway

Find internet gateways by attribute or tag

Name	Internet gateway ID	State	VPC ID	Owner
–	igw-047f59f5c6e0752d6	Attached	vpc-0692a0a9ca59e3193	383704034224
–	igw-06416e4c485770205	Attached	vpc-0e453e2e1ecbed7bf | dev-vpc	383704034224
–	igw-0773d7079f1688744	Attached	vpc-0bc47e04a8834bb17 | dev-vpc	383704034224
dev-igw	igw-0accc34e68b5ce49c	Attached	vpc-0b50742b0a8d05cf8 | dev-vpc	383704034224



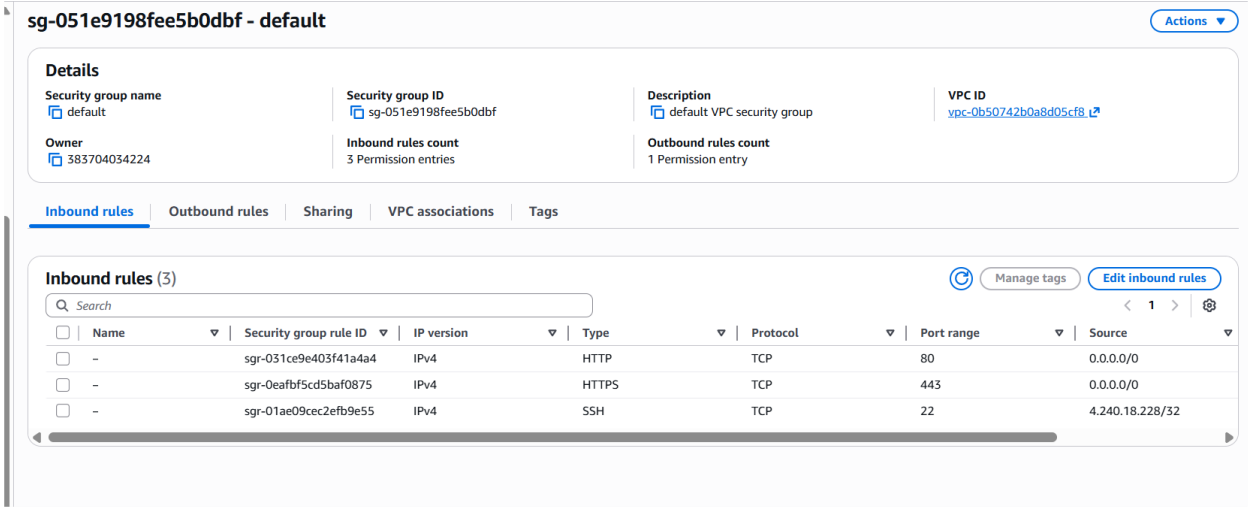
Route tables (7) Info

Last updated 1 minute ago | Actions | Create route table

Find route tables by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
–	rtb-0b83af2c02d11e053	–	–	Yes	vpc-0692a0a9ca59e3193	383704034224
–	rtb-0df312d7e961c1639	subnet-06bea409d1a53d...	–	No	vpc-0e453e2e1ecbed7bf | dev-...	383704034224
–	rtb-0e4bc309795231 31d	subnet-010f87f7a71401...	–	No	vpc-0bc47e04a8834bb17 | dev-...	383704034224
–	rtb-00282394bb493688f	–	–	Yes	vpc-0bc47e04a8834bb17 | dev-...	383704034224
–	rtb-026dda430362b8af0	–	–	Yes	vpc-0e453e2e1ecbed7bf | dev-...	383704034224
dev-rt	rtb-0286032bc9ee5963c	–	–	Yes	vpc-0042969d46f2df6cc | dev-v...	383704034224
dev-rt	rtb-0dbff1dd5a67a0cab	–	–	Yes	vpc-0b50742b0a8d05cf8 | dev-...	383704034224

Select a route table



sg-051e9198fee5b0dbf - default

Actions

Details

Security group name default	**Security group ID** sg-051e9198fee5b0dbf	**Description** default VPC security group	**VPC ID** vpc-0b50742b0a8d05cf8
Owner 383704034224	**Inbound rules count** 3 Permission entries	**Outbound rules count** 1 Permission entry	

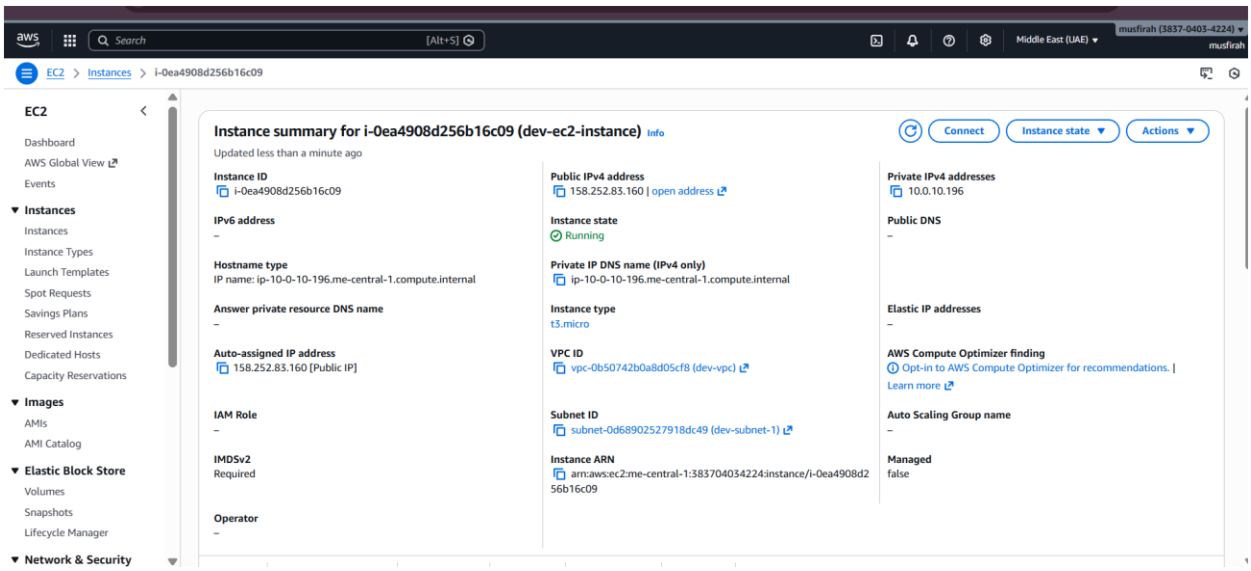
Inbound rules | Outbound rules | Sharing | VPC associations | Tags

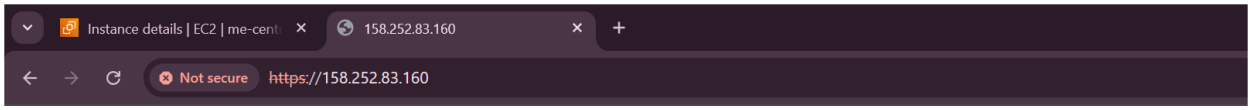
Inbound rules (3)

Manage tags | Edit inbound rules

Search

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
–	sgr-031ce9e403f41a4a4	IPv4	HTTP	TCP	80	0.0.0.0/0
–	sgr-0eafbf5cd5baf0875	IPv4	HTTPS	TCP	443	0.0.0.0/0
–	sgr-01ae09cec2efb9e55	IPv4	SSH	TCP	22	4.240.18.228/32





This is Musfira Farooq's Terraform environment.

Q3 – Ansible Playbook for EC2 Web Server Using Q2 Instance

```
@Musfira-0514 →/workspaces/exam (main) $ mkdir ansible
@Musfira-0514 →/workspaces/exam (main) $ cd ansible
@Musfira-0514 →/workspaces/exam/ansible (main) $ touch hosts
@Musfira-0514 →/workspaces/exam/ansible (main) $ nano hosts
@Musfira-0514 →/workspaces/exam/ansible (main) $ cat hosts
[ec2]
158.252.83.160
[ec2:vars]
ansible_user=ec2-user
ansible_ssh_private_key_file=~/.ssh/id_ed25519
```

```
@Musfira-0514 ➜ /workspaces/exam/ansible (main) $ touch ansible.cfg
@Musfira-0514 ➜ /workspaces/exam/ansible (main) $ nano ansible.cfg
@Musfira-0514 ➜ /workspaces/exam/ansible (main) $ cat ansible.cfg

[defaults]
host_key_checking = False
interpreter_python = /usr/bin/python3
inventory = ./hostscking=no
@Musfira-0514 ➜ /workspaces/exam/ansible (main) $
```

```
@Musfira-0514 →/workspaces/exam/ansible (main) $ touch my-playbook.yml
@Musfira-0514 →/workspaces/exam/ansible (main) $ nano my-playbook.yml
@Musfira-0514 →/workspaces/exam/ansible (main) $ cat my-playbook.yml
---
- name: Configure EC2 web server using Ansible
  hosts: ec2
  become: true

  tasks:

    - name: Update all system packages
      yum:
        name: "*"
        state: latest

    - name: Stop nginx if installed
      service:
        name: nginx
        state: stopped
        enabled: false
      ignore_errors: true

    - name: Remove nginx if present
      yum:
        name: nginx
        state: absent
      ignore_errors: true

    - name: Install Apache HTTPD
      yum:
        name: httpd
        state: present

    - name: Start and enable httpd service
```

```
@Musfira-0514 ➜ /workspaces/exam/ansible (main) $ cat my-playbook.yml
        name: nginx
        state: absent
      ignore_errors: true

    - name: Install Apache HTTPD
      yum:
        name: httpd
        state: present

    - name: Start and enable httpd service
      service:
        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: imds_token
```

```
    register: imds_token

  - name: Get public IPv4 address
    uri:
      url: http://169.254.169.254/latest/meta-data/public-ipv4
      headers:
        X-aws-ec2-metadata-token: "{{ imds_token.content }}"
      return_content: true
    register: public_ip

  - name: Get public hostname
    uri:
      url: http://169.254.169.254/latest/meta-data/public-hostname
      headers:
        X-aws-ec2-metadata-token: "{{ imds_token.content }}"
      return_content: true
    register: public_hostname

  - name: Print public IP address
    debug:
      msg: "EC2 Public IP is {{ public_ip.content }}"

  - name: Restart Apache HTTPD
    service:
      name: httpd
      state: restarted
@Musfira-0514 →/workspaces/exam/ansible (main) $
```

```
@Musfira-0514 →/workspaces/exam/ansible (main) $ ansible-playbook -i hosts my-playbook.yml
TASK [Gathering Facts] *********************************************************************************************************************************************
ok: [158.252.83.160]

TASK [Update all system packages] **********************************************************************************************************************************
ok: [158.252.83.160]

TASK [Stop nginx if installed] *************************************************************************************************************************************
changed: [158.252.83.160]

TASK [Remove nginx if present] *************************************************************************************************************************************
changed: [158.252.83.160]

TASK [Install Apache HTTPD] ****************************************************************************************************************************************
changed: [158.252.83.160]

TASK [Start and enable httpd service] ******************************************************************************************************************************
changed: [158.252.83.160]

TASK [Get IMDSv2 token] ********************************************************************************************************************************************
ok: [158.252.83.160]

TASK [Get public IPv4 address] *************************************************************************************************************************************
ok: [158.252.83.160]

TASK [Get public hostname] *****************************************************************************************************************************************
ok: [158.252.83.160]

TASK [Print public IP address] *************************************************************************************************************************************
ok: [158.252.83.160] => {
    "msg": "EC2 Public IP is 158.252.83.160"
}

TASK [Restart Apache HTTPD] ****************************************************************************************************************************************
changed: [158.252.83.160]

PLAY RECAP *********************************************************************************************************************************************************
158.252.83.160             : ok=11   changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

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

```
@Musfira-0514 ➜/workspaces/exam/ansible (main) $ curl http://158.252.83.160
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>It works! Apache httpd</title>
</head>
<body>
<p>It works!</p>
</body>
</html>
@Musfira-0514 ➜/workspaces/exam/ansible (main) $
```