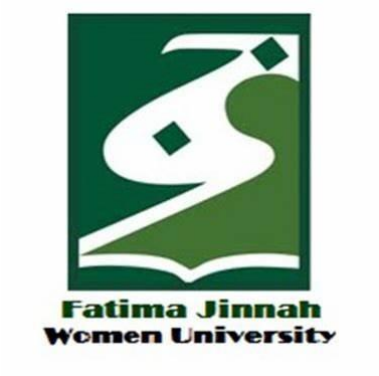
CLOUD COMPUTING LAB EXAM



SUBMITTED TO
ENGR. SHOAIB

SUBMITTED BY
HAMNA MAHMOOD

2023-BSE-025

BSE V-A

Lab Exam – Cloud Computing (IAM, Terraform, Ansible)

Q1

- q1_create_group.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam create-group --group-name SoftwareEngineer
ing
{
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAX4VWZI3PZCOGOJG2V",
        "Arn": "arn:aws:iam::542622959327:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T08:00:18+00:00"
    }
}
```

- q1_group_details.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAX4VWZI3PZCOGOJG2V",
        "Arn": "arn:aws:iam::542622959327:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T08:00:18+00:00"
    }
}
```

- q1_create_user.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam create-user --user-name HamnaMahmood25
{
    "User": {
        "Path": "/",
        "UserName": "HamnaMahmood25",
        "UserId": "AIDAX4VWZI3PRCMLD2ARH",
        "Arn": "arn:aws:iam::542622959327:user/HamnaMahmood25",
        "CreateDate": "2026-01-19T07:48:47+00:00"
    }
}
```

- q1_user_details.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam get-user --user-name HamnaMahmood25
{
    "User": {
        "Path": "/",
        "UserName": "HamnaMahmood25",
        "UserId": "AIDAX4VWZI3PRCMLD2ARH",
        "Arn": "arn:aws:iam::542622959327:user/HamnaMahmood25",
        "CreateDate": "2026-01-19T07:48:47+00:00"
    }
}
```

- q1_add_user_to_group.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam add-user-to-group --user-name HamnaMahmood
25 --group-name SoftwareEngineering
@HamnaMahmood20 ➜/workspaces/LabExam (main) $
```

- q1_group_membership.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam get-group --group-name SoftwareEngineering
{
    "Users": [
        {
            "Path": "/",
            "UserName": "HamnaMahmood25",
            "UserId": "AIDAX4VWZI3PRCMLD2ARH",
            "Arn": "arn:aws:iam::542622959327:user/HamnaMahmood25",
            "CreateDate": "2026-01-19T07:48:47+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAX4VWZI3PZCOGOJG2V",
        "Arn": "arn:aws:iam::542622959327:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T08:00:18+00:00"
    }
}
```

- q1_find_admin_policy.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam list-policies \
  --scope AWS \
  --query "Policies[?PolicyName=='AdministratorAccess'].[PolicyName,Arn]" \
  --output table
-------------------------------------------------------------------------
|                            ListPolicies                               |
+----------------------+----------------------------------------------+
|  AdministratorAccess |  arn:aws:iam::aws:policy/AdministratorAccess  |
+----------------------+----------------------------------------------+
```

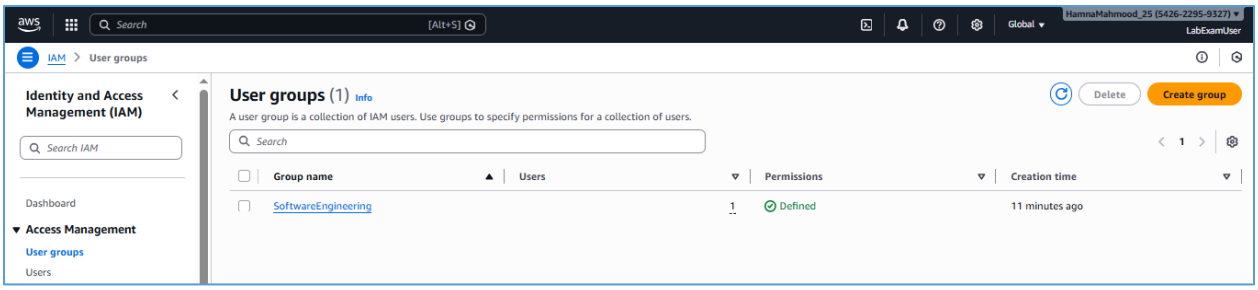
- q1_attach_admin_policy.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam attach-group-policy \
  --group-name SoftwareEngineering \
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
@HamnaMahmood20 ➜/workspaces/LabExam (main) $
```

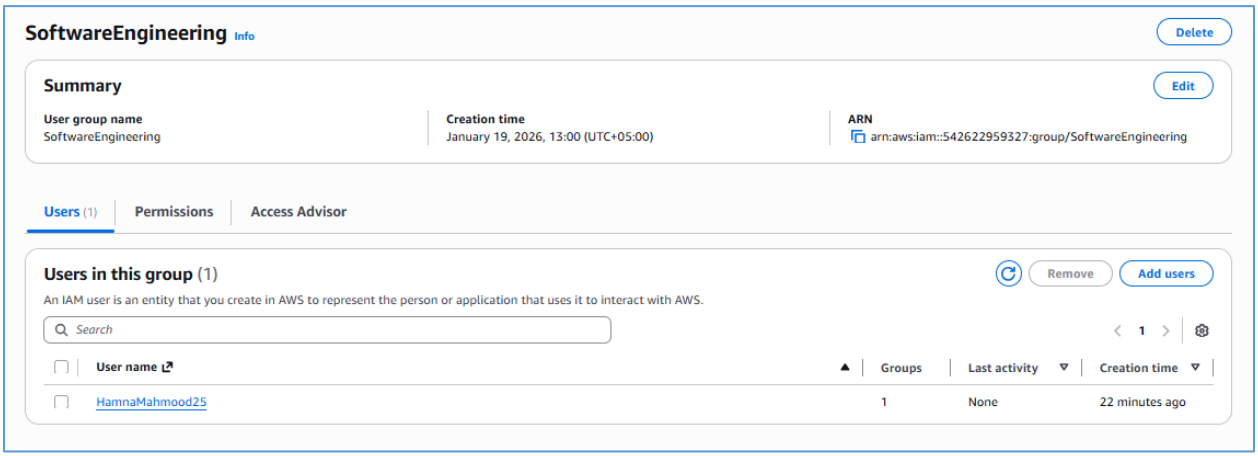
- q1_list_group_policies.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ aws iam list-attached-group-policies \
  --group-name SoftwareEngineering
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
```

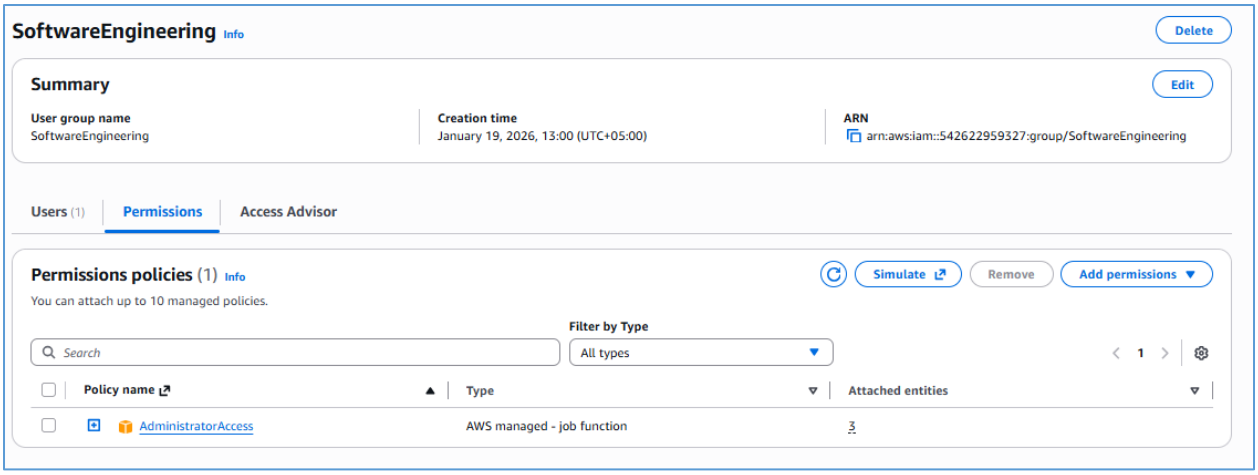
- q1_console_group.png



- q1_console_user_in_group.png



- q1_console_group_policy.png



- **Q2**
  q2_provider.png

```
provider "aws"{
region="me-central-1"
version="~>2.0"}
~
```

- q2_variables.png

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

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

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

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

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

- q2_vpc_subnet.png

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 1
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" "myapp_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"
  }
}
```

- 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

```
resource "aws_default_security_group" "myapp_default_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

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

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

  ingress {
    description = "HTTPS from anywhere"
    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" "serverkey" {
  key_name   = "serverkey"
  public_key = file("~/.ssh/id_ed25519.pub")
}
```

- q2_ec2_resource.png

```
resource "aws_instance" "myapp_ec2" {
  ami                         = "ami-0c02fb55956c7d316" # Amazon Linux 2023 (example)
  instance_type               = var.instance_type
  subnet_id                   = aws_subnet.myapp_subnet_1.id
  availability_zone           = var.availability_zone
  vpc_security_group_ids      = [aws_default_security_group.myapp_default_sg.id]
  key_name                    = aws_key_pair.serverkey.key_name
  associate_public_ip_address = true
  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.key \
  -out /etc/nginx/ssl/nginx.crt \
  -subj "/C=US/ST=State/L=City/O=Org/OU=IT/CN=localhost"

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

server {
    listen 443 ssl;
    server_name _;

    ssl_certificate /etc/nginx/ssl/nginx.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx.key;

    location / {
```

```
@HamnaMahmood20 →/workspaces/LabExam (main) $ chmod +x entry-script.sh
```

- q2_output_block.png

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 1

output "ec2_public_ip" {
  description = "Public IP address of the EC2 instance"
  value       = aws_instance.myapp_ec2.public_ip
}
```

- q2_tfvars_or_vars.png

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS 1

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

```
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

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ terraform plan
      + 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)
      + 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)
```

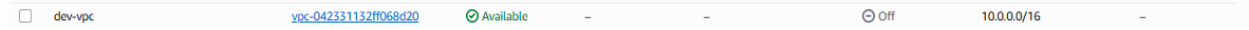
- q2_terraform_apply.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ terraform apply

      + metadata_options (known after apply)

      + 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_ec2: Creating...
aws_instance.myapp_ec2: Still creating... [00m10s elapsed]
aws_instance.myapp_ec2: Creation complete after 14s [id=i-0b49d3e79a0665285]

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

Outputs:

ec2_public_ip = "51.112.43.101"
@HamnaMahmood20 ➜/workspaces/LabExam (main) $
```

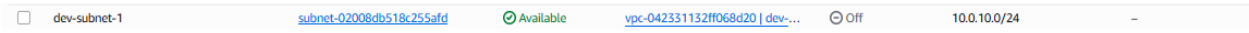
- q2_terraform_output.png

```
@HamnaMahmood20 ➜/workspaces/LabExam (main) $ terraform output
ec2_public_ip = "51.112.43.101"
@HamnaMahmood20 ➜/workspaces/LabExam (main) $
```

- q2_console_vpc.png



- q2_console_subnet.png



- q2_console_igw.png
- q2_console_route_table.png
- q2_console_sg.png
- q2_console_ec2.png
- q2_https_browser.png

Q3

- q3_hosts.png
- q3_ansible_cfg.png
- q3_playbook.png
- q3_play_run.png
- q3_http_browser.png (recommended)