Name : Maira Malik

Reg.no: 2023-BSE-040 (Section-B)

Subject: Cloud Computing

LAB-EXAM

Q1:

1.

```
@Maira222 →/workspaces/lab-exam (main) $  aws iam create-group --group-nam
e SoftwareEngineering
{
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPA2XJTBOLBPRT7E2KVK",
        "Arn": "arn:aws:iam::737230811842:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:31:30+00:00"
:
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPA2XJTBOLBPRT7E2KVK",
        "Arn": "arn:aws:iam::737230811842:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:31:30+00:00"
:
    }
}
(END)
{
```

2.

```
@Maira222 →/workspaces/lab-exam (main) $ aws iam create-user --user-name MairaMalik
{
    "User": {
        "Path": "/",
        "UserName": "MairaMalik",
        "UserId": "AIDA2XJTBOLBKQUIC2ZRL",
        "Arn": "arn:aws:iam::737230811842:user/MairaMalik",
        "CreateDate": "2026-01-19T07:34:31+00:00"
    }
}
```

3.

```
@Maira222 →/workspaces/lab-exam (main) $ aws iam get-user --user-name MairaMalik
{
    "User": {
        "Path": "/",
        "UserName": "MairaMalik",
        "UserId": "AIDA2XJTBOLBKQUIC2ZRL",
        "Arn": "arn:aws:iam::737230811842:user/MairaMalik",
        "CreateDate": "2026-01-19T07:34:31+00:00"
    }
}
```

4.

```
@Maira222 →/workspaces/lab-exam (main) $ aws iam add-user-to-group --group-name Softwar
eEngineering --user-name MairaMalik
```

5.

```
@Maira222 ➜/workspaces/lab-exam (main) $ aws iam get-group --group-name SoftwareEnginee
ring
{
    "Users": [
        {
            "Path": "/",
            "UserName": "MairaMalik",
            "UserId": "AIDA2XJTBOLBKQUIC2ZRL",
            "Arn": "arn:aws:iam::737230811842:user/MairaMalik",
            "CreateDate": "2026-01-19T07:34:31+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPA2XJTBOLBPRT7E2KVK",
        "Arn": "arn:aws:iam::737230811842:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:31:30+00:00"
    }
}
```

6.

```
@Maira222 ➜/workspaces/lab-exam (main) $ aws iam get-policy --policy-arn arn:aws:iam::a
ws:policy/AdministratorAccess
{
    "Policy": {
        "PolicyName": "AdministratorAccess",
        "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
        "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 2,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "Description": "Provides full access to AWS services and resources.",
        "CreateDate": "2015-02-06T18:39:46+00:00",
        "UpdateDate": "2015-02-06T18:39:46+00:00",
        "Tags": []
    }
}
```

7.

```
@Maira222 ➜/workspaces/lab-exam (main) $ aws iam attach-group-policy --group-name Softw
areEngineering --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

8.

```
@Maira222 ➜/workspaces/lab-exam (main) $ aws iam list-attached-group-policies --group-n
ame SoftwareEngineering
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
```

Q2:

1.

```
@Maira222 ➜/workspaces/lab-exam (main) $ cat ~/.aws/credentials
cat ~/.aws/config
[default]
aws_access_key_id =  AKIA2XJTBOLBHMGZUHUO
aws_secret_access_key = 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
[default]
region = us-east-1
```

2.

terraform > variables.tf

```hcl
variable "vpc_cidr_block" {
  description = "CIDR block for the VPC"
  type        = string
  default     = "10.0.0.0/16"
}

variable "subnet_cidr_block" {
  description = "CIDR block for the subnet"
  type        = string
  default     = "10.0.10.0/24"
}

variable "availability_zone" {
  description = "Availability zone for the subnet"
  type        = string
  default     = "me-central-1a"
}

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

```hcl
variable "instance_type" {
  description = "EC2 instance type"
  type        = string
  default     = "t3.micro"
}
```

3.

terraform > 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"

```

4.

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

# Manage default route table and add route
resource "aws_default_route_table" "myapp_route_table" {
  default_route_table_id = aws_vpc.myapp_vpc.default_rou

  route {
    cidr_block      = "0.0.0.0/0"
    gateway_id      = aws_internet_gateway.myapp_igw.id
  }

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

5.

terraform > outputs.tf

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

output "ec2_public_hostname" {
  description = "Public DNS hostname of the EC2 instance"
  value       = aws_instance.myapp_ec2.public_dns
}
```

6.

```
@Maira222 →/workspaces/lab-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_ed2
5519
Your public key has been saved in /home/codespace/.ssh/id_ed25519
.pub
The key fingerprint is:
SHA256:37HJvtLfXp8Vq89ZTm0h5MBbrgc5k+ht+cXQeZRrbdo codespace@code
spaces-fd0fd2
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|          .     .|
|           o o ..|
|          . X ..+|
|        S. B.= B+|
|        ...o*+=+*|
|         ..*=..+E|
|          o.+ +=B|
|           .o=o*=|
+----[SHA256]-----+
```

7.

```
# Manage default security group
resource "aws_default_security_group" "myapp_default_sg" {
  vpc_id = aws_vpc.myapp_vpc.id

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

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

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

  # Allow all outbound traffic
  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

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

8.

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

# Create EC2 instance
```

9.

```
# Create EC2 instance
resource "aws_instance" "myapp_ec2" {
  ami           = "ami-02e22a303d3db269d"  # Amazon Linux
  instance_type = var.instance_type

  subnet_id                   = aws_subnet.myapp_subnet_1
  vpc_security_group_ids      = [aws_default_security_group
  availability_zone           = var.availability_zone
  associate_public_ip_address = true
  key_name                    = aws_key_pair.serverkey.key

  user_data = file("${path.module}/entry-script.sh")

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

10.

terraform > $ entry-script.sh

```
#!/bin/bash
set -e

# Update system packages
yum update -y

# Install Nginx
yum install -y nginx

# Generate self-signed certificate
mkdir -p /etc/nginx/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/nginx/ssl/nginx-server.key \
  -out /etc/nginx/ssl/nginx-server.crt \
  -subj "/CN=example.com"

# Configure Nginx
cat > /etc/nginx/nginx.conf <<'EOF'
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log notice;
pid /run/nginx.pid;
```

terraform > $ entry-script.sh

```
events {
    worker_connections 1024;
}

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

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

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

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

    # HTTP - redirect to HTTPS
    server {
        listen 80;
```

```
</head>
<body>
    <div class="container">
        <h1>🚀 Welcome!</h1>
        <p>This is MairaMalik's Terraform environment</p>
        <p><strong>This instance was created with Terraform</strong></p>
    </div>
</body>
</html>
EOF

# Enable and start Nginx
systemctl enable nginx
systemctl start nginx

echo "Nginx configured and running successfully!"
```

```
http {
    server {
        return 301 https://$host$request_uri;
    }

    # HTTPS
    server {
        listen 443 ssl;
        server_name _;

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

        ssl_protocols TLSv1.2 TLSv1.3;
        ssl_ciphers HIGH:!aNULL:!MD5;

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

11.

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

Outputs:

ec2_public_hostname = ""
ec2_public_ip = "40.172.221.78"
@Maira222 ➜/workspaces/lab-exam/terraform (main) $
Waiting for instance to fully initialize...
@Maira222 ➜/workspaces/lab-exam/terraform (main) $
<!DOCTYPE html>
    <title>MairaMalik's Terraform Environment</title>
        <h1>🚀 Welcome!</h1>
        <p>This is MairaMalik's Terraform environment</p>
        <p><strong>This instance was created with Terraform</strong></p>
```

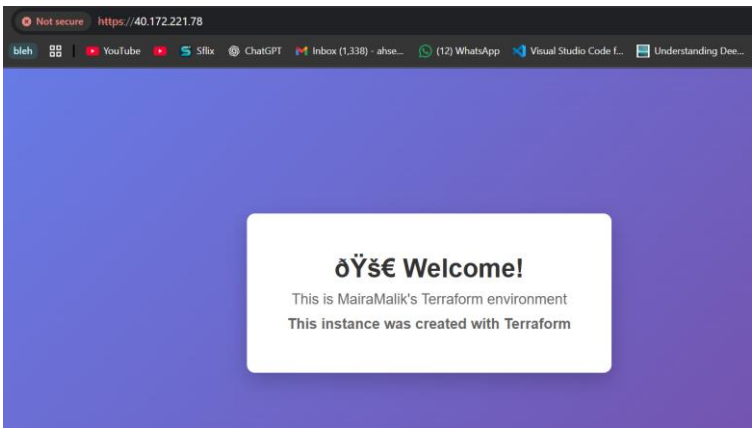
12.

```
@Maira222 ➜/workspaces/lab-exam (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 5.0"...
- Finding latest version of hashicorp/http...
- Installing hashicorp/http v3.5.0...
- Installed hashicorp/http v3.5.0 (signed by HashiCorp)
- Installing hashicorp/aws v5.100.0...
- Installed hashicorp/aws v5.100.0 (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.
```

13.

```
@Maira222 ➜/workspaces/lab-exam (main) $ terraform 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_hostname = null -> (known after apply)
  ~ ec2_public_ip       = "3.29.125.138" -> (known after apply)
aws_instance.myapp_ec2: Creating...
aws_instance.myapp_ec2: Still creating... [00m10s elapsed]
aws_instance.myapp_ec2: Creation complete after 14s [id=i-077ef713438dd2a06]

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

14.



Q3

1.

```
[ec2]
40.172.221.78

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

```

2.

ansible > ansible.cfg

```
[defaults]
host_key_checking = False
inventory = ./hosts
remote_user = ec2-user
private_key_file = ~/.ssh/id_ed25519

[ssh_connection]
ssh_args = -o StrictHostKeyChecking=no
remote_python_interpreter=/usr/bin/python3

```

3.

ansible > ! my-playbook.yml

```
---
- name: Configure EC2 Web Server with Apache HTTPD
  hosts: ec2
  become: true

  tasks:
    - name: Update all packages
      yum:
        name: '*'
        state: latest

    - name: Stop Nginx service if present
      service:
        name: nginx
        state: stopped
        enabled: no
```

```
- name: Configure EC2 Web Server with Apache HTTPD
  tasks:

    - name: Uninstall Nginx if present
      yum:
        name: nginx
        state: absent
      ignore_errors: yes

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

    - name: Start and enable Apache HTTPD
      service:
        name: httpd
        state: started
        enabled: yes

    - name: Get IMDSv2 token
      uri:
        url: http://169.254.169.254/latest/api/token
        method: PUT
```

```
      content: |
        </head>
        <body>
            <div class="container">
                <h1>🚀 Welcome!</h1>
                <p>This is MairaMalik's Apache Web Server</p>
                <p><strong>Managed with Ansible</strong></p>
                <p>Public IP: {{ public_ipv4.content }}</p>
            </div>
        </body>
        </html>
      dest: /var/www/html/index.html

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

4.

```
@Maira222 ➜/workspaces/lab-exam (main) $ ansible-playbook -i hosts my-playbook.ymlansibl
e-playbook -i hosts my-playbook.yml
[WARNING]: Ansible is being run in a world writable directory (/workspaces/lab-exam/ansib
le), ignoring it as an ansible.cfg source. For more information see https://docs.ansible.
com/ansible/devel/reference_appendices/config.html#cfg-in-world-writable-dir

PLAY [Configure EC2 Web Server with Apache HTTPD] ***********************************
****

TASK [Gathering Facts] ***************************************************************
****
[WARNING]: Host '40.172.221.78' is using the discovered Python interpreter at '/usr/bin/p
ython3.9', but future installation of another Python interpreter could cause a different
interpreter to be discovered. See https://docs.ansible.com/ansible-core/2.20/reference_ap
pendices/interpreter_discovery.html for more information.
ok: [40.172.221.78]

TASK [Update all packages] ***********************************************************
****
ok: [40.172.221.78]

TASK [Stop Nginx service if present] *************************************************
****
changed: [40.172.221.78]
```

```
TASK [Display public IP] *************************************************************
****
ok: [40.172.221.78] => {
    "msg": "Instance public IP: 40.172.221.78"
}

TASK [Display public hostname] *******************************************************
****
ok: [40.172.221.78] => {
    "msg": "Instance public hostname: "
}

TASK [Create Apache welcome page] ****************************************************
****
changed: [40.172.221.78]

TASK [Restart Apache HTTPD] **********************************************************
****
changed: [40.172.221.78]

PLAY RECAP ***************************************************************************
****
40.172.221.78              : ok=13   changed=6    unreachable=0    failed=0    skipped=0
   rescued=0    ignored=0
```

5.