Name: Zuha Irfan

Roll No: 2023-BSE-073

Semester: V-B

Lab Exam

Q1 – AWS IAM Setup Using AWS CLI and Console Verification (10 marks)

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ aws iam create-group --group-name SoftwareEngineering
aws iam get-group --group-name SoftwareEngineering
aws iam create-user --user-name Zuha-Irfan-2023-bse-073
aws iam get-user --user-name Zuha-Irfan-2023-bse-073
aws iam add-user-to-group --user-name Zuha-Irfan-2023-bse-073 --group-name SoftwareEngineering
aws iam get-group --group-name SoftwareEngineering
aws iam list-policies --scope AWS --query "Policies[?PolicyName=='AdministratorAccess'].[PolicyName,Arn]" --output table
aws iam attach-group-policy --group-name SoftwareEngineering --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
aws iam list-attached-group-policies --group-name SoftwareEngineering
```

```
        },
        {
            "Path": "/",
            "UserName": "Zuha-Irfan-2023-bse-073",
            "UserId": "AIDAQYF75YQJQJIQQLZEK",
            "Arn": "arn:aws:iam::051942114323:user/Zuha-Irfan-2023-bse-073",
            "CreateDate": "2026-01-19T07:52:15+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "SoftwareEngineering",
        "GroupId": "AGPAQYF75YQJQ3WVPRXSR",
        "Arn": "arn:aws:iam::051942114323:group/SoftwareEngineering",
        "CreateDate": "2026-01-19T07:31:55+00:00"
    }
}

An error occurred (EntityAlreadyExists) when calling the CreateUser operation: User with name Zuha-Irfan-2023-bse-073 already exists.
{
    "User": {
        "Path": "/",
        "UserName": "Zuha-Irfan-2023-bse-073",
        "UserId": "AIDAQYF75YQJQJIQQLZEK"
```

```
        "UserName": "YOURNAME",
        "UserId": "AIDAQYF75YQJ63G3JWESW",
        "Arn": "arn:aws:iam::051942114323:user/YOURNAME",
        "CreateDate": "2026-01-19T07:31:59+00:00"
    },
    {
        "Path": "/",
        "UserName": "Zuha-Irfan-2023-bse-073",
        "UserId": "AIDAQYF75YQJQJIQQLZEK",
        "Arn": "arn:aws:iam::051942114323:user/Zuha-Irfan-2023-bse-073",
        "CreateDate": "2026-01-19T07:52:15+00:00"
    }
],
"Group": {
    "Path": "/",
    "GroupName": "SoftwareEngineering",
    "GroupId": "AGPAQYF75YQJQ3WVPRXSR",
    "Arn": "arn:aws:iam::051942114323:group/SoftwareEngineering",
    "CreateDate": "2026-01-19T07:31:55+00:00"
}
```



```
{
    "User": {
        "Path": "/",
        "UserName": "YOURNAME",
        "UserId": "AIDAQYF75YQJ63G3JWESW",
        "Arn": "arn:aws:iam::051942114323:user/YOURNAME",
        "CreateDate": "2026-01-19T07:31:59+00:00"
    }
}
{
    "User": {
        "Path": "/",
        "UserName": "YOURNAME",
        "UserId": "AIDAQYF75YQJ63G3JWESW",
        "Arn": "arn:aws:iam::051942114323:user/YOURNAME",
        "CreateDate": "2026-01-19T07:31:59+00:00"
    }
}
```

```
-------------------------------------------------------------------------
|                            ListPolicies                               |
+----------------------+----------------------------------------------+
|  AdministratorAccess |  arn:aws:iam::aws:policy/AdministratorAccess  |
+----------------------+----------------------------------------------+
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        }
    ]
}
```

Q2 – Terraform Lab: Simple AWS Environment with Nginx over HTTPS (30 marks)

Lab_Exam > terraform > variables.tf

```
1   variable "vpc_cidr_block" {
2     description = "CIDR block for VPC"
3     type        = string
4   }
5
6   variable "subnet_cidr_block" {
7     description = "CIDR block for subnet"
8     type        = string
9   }
10
11  variable "availability_zone" {
12    description = "Availability Zone for subnet and EC2"
13    type        = string
14  }
15
16  variable "env_prefix" {
17    description = "Prefix for resource names"
18    type        = string
19  }
20
21  variable "instance_type" {
22    description = "EC2 instance type"
```

Lab_Exam > terraform > main.tf

```hcl
provider "aws" {
  region = "me-central-1"  # adjust if needed
}

# 1️⃣ Create VPC
resource "aws_vpc" "myapp_vpc" {
  cidr_block = var.vpc_cidr_block
  tags = {
    Name = "${var.env_prefix}-vpc"
  }
}

# 2️⃣ Create Subnet
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"
  }
}

```

Lab_Exam > terraform > outputs.tf

```hcl
output "ec2_public_ip" {
  value = aws_instance.web_server.public_ip
}

output "ec2_public_dns" {
  value = aws_instance.web_server.public_dns
}

```

Lab_Exam > terraform > $ entry-script.sh

```
#!/bin/bash
# Update system
sudo dnf update -y

# Install Nginx
sudo dnf install -y nginx

# Generate self-signed TLS certificate
sudo mkdir -p /etc/nginx/ssl
sudo 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=Punjab/L=Lahore/O=DevOps/CN=example.com"

# Configure Nginx for HTTPS
sudo tee /etc/nginx/conf.d/default.conf > /dev/null <<EOF
server {
    listen 80;
    server_name _;
    return 301 https://\$host\$request_uri;
}
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ ssh -i MyED25519Key.pem ec2-user@3.28.45.44
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Dec 25 06:28:13 2025 from 4.240.18.226
[ec2-user@ip-10-0-1-127 ~]$ sudo dnf update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-127 ~]$ sudo dnf install -y nginx
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ cd /workspaces/CC-ZuhaIrfan-073
mkdir -p Lab_Exam/terraform
cd Lab_Exam/terraform
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ cat <<EOF > variables.tf
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}
EOF
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ cat <<EOF > main.tf
provider "aws" {}

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

resource "aws_subnet" "dev_subnet" {
  vpc_id            = aws_vpc.dev_vpc.id
  cidr_block        = var.subnet_cidr_block
  availability_zone = var.availability_zone
  map_public_ip_on_launch = true
  tags = {
    Name = "\${var.env_prefix}-subnet-1"
  }
}

resource "aws_internet_gateway" "dev_igw" {
  vpc_id = aws_vpc.dev_vpc.id
  tags = {
    Name = "\${var.env_prefix}-igw"
  }
}
```

```
resource "aws_default_route_table" "dev_rt" {
  default_route_table_id = aws_vpc.dev_vpc.default_route_table_id

EOF Name = "\${var.env_prefix}-rt"way.dev_igw.id
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ cat <<EOF > outputs.tf
output "vpc_id" {
  value = aws_vpc.dev_vpc.id
}
EOF
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- 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.
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ terraform plan \
-var="vpc_cidr_block=10.0.0.0/16" \
-var="subnet_cidr_block=10.0.10.0/24" \
-var="availability_zone=me-central-1a" \
-var="env_prefix=dev" \ …
      + region                             = "me-central-1"
      + tags                               = {
          + "Name" = "dev-vpc"
        }
      + tags_all                           = {
          + "Name" = "dev-vpc"
        }
    }

Plan: 4 to add, 0 to change, 0 to destroy.

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

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ terraform apply -auto-approve \
-var="vpc_cidr_block=10.0.0.0/16" \
-var="subnet_cidr_block=10.0.10.0/24" \
-var="availability_zone=me-central-1a" \
-var="env_prefix=dev" \ …
        }
    }

Plan: 4 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + vpc_id = (known after apply)
aws_vpc.dev_vpc: Creating...
aws_vpc.dev_vpc: Creation complete after 3s [id=vpc-08cf7aadf0829a857]
aws_internet_gateway.dev_igw: Creating...
aws_subnet.dev_subnet: Creating...
aws_internet_gateway.dev_igw: Creation complete after 0s [id=igw-00a2e5f13a1e8493a]
aws_default_route_table.dev_rt: Creating...
aws_default_route_table.dev_rt: Creation complete after 2s [id=rtb-0df620e90afe5323e]
aws_subnet.dev_subnet: Still creating... [00m10s elapsed]
aws_subnet.dev_subnet: Creation complete after 11s [id=subnet-0a8506e9347db0a7a]
```

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

Outputs:

aws_instance_public_ip = "3.28.45.44"
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073 (main) $ ssh -i MyED25519Key.pem ec2-user@3.28.45.44
```

```
[ec2-user@ip-10-0-1-127 ~]$ curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

3.28.45.44

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Q3 – Ansible Playbook for EC2 Web Server Using Q2 Instance (10 marks)

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ cd /workspaces/CC-ZuhaIrfan-073
mkdir -p Lab_Exam/ansible
cd Lab_Exam/ansible
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible (main) $ cat <<EOF > hosts
[ec2]
EC2_PUBLIC_IP

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

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible (main) $ cat <<EOF > ansible.cfg
[defaults]
host_key_checking = False
inventory = ./hosts
interpreter_python = /usr/bin/python3
EOF
```

```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.28.0

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.
```

```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible (main) $ cat <<EOF > my-playbook.yml
- name: Configure EC2 with Apache
  hosts: ec2
  become: true
  ...
    - name: Install httpd
      yum:
        name: httpd
        state: present

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

    - name: Get IMDSv2 token
```

```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ cd /workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible
echo "[webservers]" > hosts
echo "$(terraform output -raw ec2_public_ip) ansible_user=ec2-user ansible_ssh_private_key_file=/home/codespace/.ssh/id_ed25519" >> hosts
ansible-playbook -i hosts my-playbook.yml
[WARNING]: Ansible is being run in a world writable directory (/workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible), 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
[WARNING]:  * Failed to parse /workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible/hosts with yaml plugin: We were unable to read either as JSON nor YAML, these are the errors we got
from each: JSON: Expecting value: line 1 column 2 (char 1)  Syntax Error while loading YAML.   . unacceptable character #x001b: control characters are not allowed   in
"<unicode string>", position 13
[WARNING]:  * Failed to parse /workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible/hosts with ini plugin: not enough values to unpack (expected 3, got 2)
[WARNING]: Unable to parse /workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
[WARNING]: Could not match supplied host pattern, ignoring: ec2

PLAY [Configure EC2 with Apache] ***********************************************************************************************************************************
skipping: no hosts matched

PLAY RECAP *********************************************************************************************************************************************************
```

Lab_Exam > terraform > variables.tf

```
variable "vpc_cidr_block" {}
variable "subnet_cidr_block" {}
variable "availability_zone" {}
variable "env_prefix" {}
variable "instance_type" {}

```

Lab_Exam > terraform > main.tf

```hcl
provider "aws" {
  region = "me-central-1"
}

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

resource "aws_subnet" "dev_subnet" {
  vpc_id                  = aws_vpc.dev_vpc.id
  cidr_block              = var.subnet_cidr_block
  availability_zone       = var.availability_zone
  map_public_ip_on_launch  = true
  tags = {
    Name = "${var.env_prefix}-subnet-1"
  }
}

resource "aws_internet_gateway" "dev_igw" {
  vpc_id = aws_vpc.dev_vpc.id
  tags = {
    Name = "${var.env_prefix}-igw"
  }
}

resource "aws_security_group" "allow_ssh_http" {
  name        = "${var.env_prefix}-sg"
  description = "Allow SSH and HTTP"
  vpc_id      = aws_vpc.dev_vpc.id
```

Lab_Exam > terraform > outputs.tf

```
output "ec2_public_ip" {
  value = aws_instance.my_ec2.public_ip
}

```

Lab_Exam > terraform > $ entry-script.sh

```
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo systemctl enable httpd
sudo systemctl start httpd
echo "<h1>Welcome to Zuha Irfan's Webserver</h1>" | sudo tee /var/www/html/index.html

```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible (main) $ ssh-keygen -t ed25519 -f /home/codespace/.ssh/id_ed25519 -N ""
Generating public/private ed25519 key pair.
/home/codespace/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
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:Ob7x3wYKGinnRabxJdqNWWlhLGUlpqkHQyhZ3KQKlxs codespace@codespaces-2a945b
The key's randomart image is:
+--[ED25519 256]--+
|   +.+o  o=..    |
|  o.oo. .=+.     |
|. E.. o oo o     |
| o +  .++.=      |
|  o   .XSO       |
|    . *oB.. .    |
|     + +o. . .   |
|      o  +.  ..  |
|       . .....  |
```

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ terraform init
terraform apply -auto-approve \
  -var="vpc_cidr_block=10.1.0.0/16" \
  -var="subnet_cidr_block=10.1.10.0/24" \
  -var="availability_zone=me-central-1a" \ …
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.28.0

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.
```

Cleanup (ungraded)

```
@Zuha-Irfan →/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ terraform init
terraform apply -auto-approve \
  -var="vpc_cidr_block=10.1.0.0/16" \
  -var="subnet_cidr_block=10.1.10.0/24" \
  -var="availability_zone=me-central-1a" \ …
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 02m30s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 02m40s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 02m50s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 03m00s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 03m10s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 03m20s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 03m30s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 03m40s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 03m50s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 04m00s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 04m10s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 04m20s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 04m30s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 04m40s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 04m50s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 05m00s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 05m10s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 05m20s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 05m30s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 05m40s elapsed]
aws_vpc.dev_vpc: Still destroying... [id=vpc-08cf7aadf0829a857, 05m50s elapsed]
```

```
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ rm -f terraform.tfstate terraform.tfstate.backup .terraform.lock.hcl
rm -rf .terraform
@Zuha-Irfan ➜/workspaces/CC-ZuhaIrfan-073/Lab_Exam/terraform (main) $ cd /workspaces/CC-ZuhaIrfan-073/Lab_Exam/ansible
rm -f hosts my-playbook.yml ansible.cfg
```

Sorry sir my laptrops stops working so I can’t proceed further