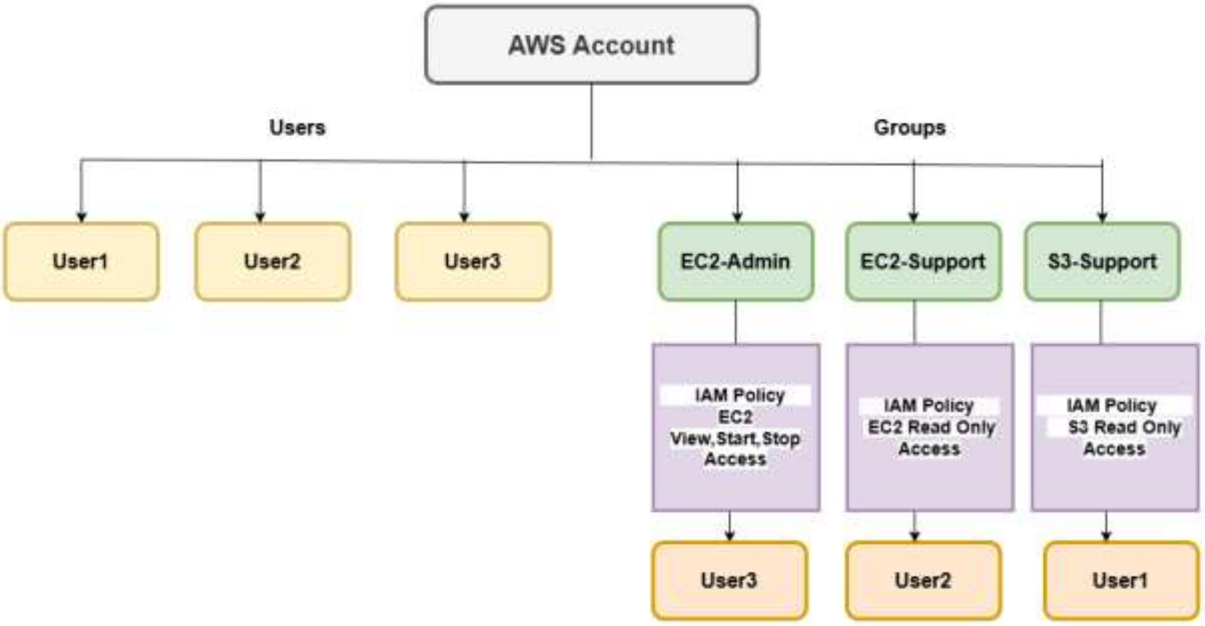
Task 3:

(a) AWS Identity and Access Management (IAM) Task
(b) Cloud Networking with Amazon VPC

AWS Identity and Access Management (IAM) is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage **users**, **security credentials** such as access keys, and **permissions** that control which AWS resources users can access.


AWS Account
Users
Groups
User1
User2
User3
EC2-Admin
EC2-Support
S3-Support
IAM Policy EC2 View,Start,Stop Access
IAM Policy EC2 Read Only Access
IAM Policy S3 Read Only Access
User3
User2
User1

Activity-1 : Creating Users:

Step 1: Sign in to the AWS Management Console

1. Go to the AWS Management Console at https://aws.amazon.com/console/.
2. Sign in with your AWS account credentials.

Step 2: Navigate to the IAM (Identity and Access Management) Service

1. In the AWS Management Console, search for **IAM** in the search bar or find it under the **Security, Identity, & Compliance** category.
2. Click on **IAM** to open the IAM dashboard.

Step 3: Create a New User

1. In the IAM dashboard, click on **Users** in the left-hand menu.

2. Click on **Create User**

Step 4: Configure the User Details

1. Enter the **User name** :User1
2. Under **Select AWS access type**, check **AWS Management Console access**.
   - For **Console password**, choose **Custom password** (You create a password for the User1 as User1@123).
3. Uncheck **Require password reset** to force the user to change their password upon first login.

. Step 5: Set Permissions

1. Click **Next: Permissions**.
2. Choose the following options to set permissions for the user:
   - **Attach existing policies directly**: Select policies that define the permissions for the user.

Step 6: Review and Create the User

1. Click **Next: Tags** to add optional tags for the user.
2. Click **Next: Review** to review the user's details and permissions.
3. Click **Create user** to finalize the process.

Click on **download .csv file**

Step 7: Write the above steps to create User2

Step 8: Write the above steps to create User3

Activity 2: Create User Groups

(a) Create “EC2-Admin” User Group

Step 1: Navigate to the IAM (Identity and Access Management) Service

1. In the AWS Management Console, search for **IAM** in the search bar or find it under the **Security, Identity, & Compliance** category.
2. Click on **IAM** to open the IAM dashboard.

Step 2: Create a New User Group

1. In the IAM dashboard, click on **User groups** in the left-hand menu.
2. Click on **Create group**.

Step 3: Configure the Group Details

1. Enter **EC2-Admin** as the **Group name**.
2. Click **Create group** to create the group without attaching any policies at this step.

Step 4: Attach an Inline Policy to the Group

1. In the **User groups** list, click on the **EC2-Admin** group name.
2. Click on the **Permissions** tab.
3. Click **Add permissions** and then select **Create inline policy**.

Step 5: Define the Inline Policy

1. In the **Create policy** editor, switch to the **JSON** tab.
2. Paste the following policy JSON to allow view, start, and stop access to EC2 instances:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeImages",
                "ec2:DescribeVolumes",
                "ec2:DescribeTags",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*"
        }
    ]
}
```

Step 7: Name and Attach the Policy

1. Enter a name for the policy, such as **EC2-ViewStartStopAccess**.
2. Click **Create policy** to attach it to the group.

Step 8: Add Users to the Group

1. In the **EC2-Admin** group page, click on the **Users** tab.
2. Click **Add users**.

3. Select the **User3** to add to this group.
4. Click **Add users** to finalize the process.

(b) Create “EC2-Support” UserGroups

Step 1: Navigate to the IAM Service

1. In the AWS Management Console, search for **IAM** in the search bar or find it under the **Security, Identity, & Compliance** category.
2. Click on **IAM** to open the IAM dashboard.

Step 2: Create a New User Group

1. In the IAM dashboard, click on **Groups** in the left-hand menu.
2. Click on **Create New Group**.

Step 3: Configure the Group Details

1. Enter **EC2-Support** as the **Group Name**.
2. Click **Next Step** to proceed.

Step 4: Attach a Policy to the Group

1. On the **Attach Policy** page, use the search bar to find the **AmazonEC2ReadOnlyAccess** policy.
2. Select the checkbox next to **AmazonEC2ReadOnlyAccess**.
3. Click **Next Step** to continue.

Step 5: Review and Create the Group

1. Review the group's name and attached policies.
2. Click **Create Group** to finalize the process.

Step 6: Add Users to the Group (Optional)

1. To add users, go to the **Groups** section, select **EC2-Support**, click on the **Group Actions** dropdown, and choose **Add Users to Group**.
2. Select the **User2** and click **Add Users**.

(c) Create “S3-Support” UserGroup

Step 1: Navigate to the IAM (Identity and Access Management) Service

1. In the AWS Management Console, search for **IAM** in the search bar or find it under the **Security, Identity, & Compliance** category.
2. Click on **IAM** to open the IAM dashboard.

Step 2: Create a New User Group

1. In the IAM dashboard, click on **User groups** in the left-hand menu.
2. Click on **Create group**.

Step 3: Configure the User Group Details

1. In the **Group name** field, enter **S3-Support**.
2. Click **Next**.

Step 4: Attach the S3 Read-only Access Policy

1. On the **Attach policies** page, search for **AmazonS3ReadOnlyAccess**.
2. Check the box next to the **AmazonS3ReadOnlyAccess** policy to grant the group read-only access to Amazon S3.
3. Click **Next**.

Step 5: Review and Create the Group

1. Review the group name and attached policy on the **Review** page.
2. Click **Create group** to finalize the process.

Step 6: Add Users to the Group (Optional)

1. In the **User groups** page, click on the **S3-Support** group you just created.
2. Click on the **Users** tab.
3. Click **Add users**.
4. Select the **User1 t**o add to this group, Click **Add users**.

Activity 3: Create EC2 Instance “My Server” with Linux OS Image

1. **Navigate to Compute Engine**
    - In the Google Cloud Console, click on the **Navigation Menu** (top left).
    - Select **Compute Engine** > **VM instances**.
2. **Create a New VM Instance**
    - Click on the **Create Instance** button.
    - **Name** your instance “My-Server” (e.g., my-vm-instance).
    - **Region and Zone**: Select a region close to your user base or requirements. The zone is a specific data center within a region.
    - **Machine Configuration:**

    - Choose a **machine family** (e.g., General-purpose).
    - Select a **machine type** (e.g., e2-medium with 2 vCPUs and 4 GB RAM).
  - **Boot Disk:**
    - The default is a ubuntu Linux image, but you can choose other operating systems.
    - Set the disk size (default is 10 GB).
  - **Firewall**: You can allow HTTP and **HTTPS** traffic if you plan to run a web server.
  - **Identity and API access**: Choose default service account or a specific service account for the VM.
  - Click **Create** to launch your virtual machine.

Activity 4: Create S3 bucket and add some files to bucket

Create an S3 Bucket and Upload a File

1. **Sign in to AWS Management Console:**
   - Go to the AWS Management Console.
   - Sign in with your AWS account.
2. **Navigate to S3 Dashboard:**
   - In the AWS Management Console, search for **S3** in the services menu and click on it.
3. **Create a Bucket:**
   - Click the **Create Bucket** button.
   - Enter a **Bucket Name** (must be globally unique).
   - Choose a **Region** (e.g., us-east-1).
   - Configure settings (e.g., versioning, encryption, tags).
   - Click **Create Bucket**.
4. **Upload a File:**
   - Select the bucket you just created.
   - Click the **Upload** button.
   - Add files from your computer and click **Upload**.

Activity 5: Sign-In and Test Users

1. In the navigation pane on the left, choose **Dashboard**.

- A **Sign-in URL for IAM users in this account** link is displayed on the right. It will look similar to: *https://123456789012.signin.aws.amazon.com/console*
- This link can be used to sign-in to the AWS Account you are currently using.
- Copy the **Sign-in URL for IAM users in this account** to a text editor.

2. Open a private (Incognito) window.

- Choose the ellipsis at the top-right of the screen
- Select **New Incognito Window**

3. Paste the **IAM users sign-in** link into the address bar of your private browser session and press **Enter**.

- Sign-in with:
  - **IAM user name:** User1
  - **Password:**User1@123

4. In the search box to the right of **Services**, search for and choose **S3** to open the S3 console.

- Choose the name of the bucket that exists in the account and browse the contents.
- Since your user1 is part of the **S3-Support** Group in IAM, they have permission to view a list of Amazon S3 buckets and the contents.

Now, test whether they have access to Amazon EC2.

5. In the search box to the right of **Services**, search for and choose **EC2** to open the EC2 console.

- In the left navigation pane, choose **Instances**.
- You cannot see any instances. Instead, you see a message that states *You are not authorized to perform this operation*. This is because this user has not been granted any permissions to access Amazon EC2.

6. At the top of the screen, choose **User1**

- Choose **Sign Out**

7. Now sign-in as **User2**, who has been hired as your Amazon EC2 support person.

- Paste the **IAM users sign-in** link into your private browser tab's address bar and press **Enter**.
- Sign-in with:
  - **IAM user name: User2**
  - **Password:**User2@123

8. In the search box to the right of **Services**, search for and choose **EC2** to open the EC2 console.

- In the navigation pane on the left, choose **Instances**.
- You are now able to see an Amazon EC2 instance "**MyServer**" because you have Read only permissions.
- However, you will not be able to make any changes to Amazon EC2 resources.

9. Select the instance named ”**MyServer”**

- In the **Instance state** menu above, select **Stop instance**.
- In the **Stop Instance** window, select **Stop**.
- You will receive an error stating *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to view information, without making changes.
- Choose the X to close the *Failed to stop the instance* message.

10. Next, check if User-2 can access Amazon S3.

- In the search box to the right of **Services**, search for and choose **S3** to open the S3 console.
- You will see the message **“You don't have permissions to list buckets”** because User2 does not have permission to access Amazon S3.
- At the top of the screen, choose User-2
- Choose **Sign Out**

11. You will now sign-in as **User3**, who has been hired as your Amazon EC2 administrator.

- Sign-in with:
  - **IAM user name:** User3
  - **Password:** User3@123

12. In the search box to the right of **Services**, search for and choose **EC2** to open the EC2 console.

- In the navigation pane on the left, choose **Instances**.
- As an EC2 Administrator, you should now have permissions to **Stop** the Amazon EC2 instance.

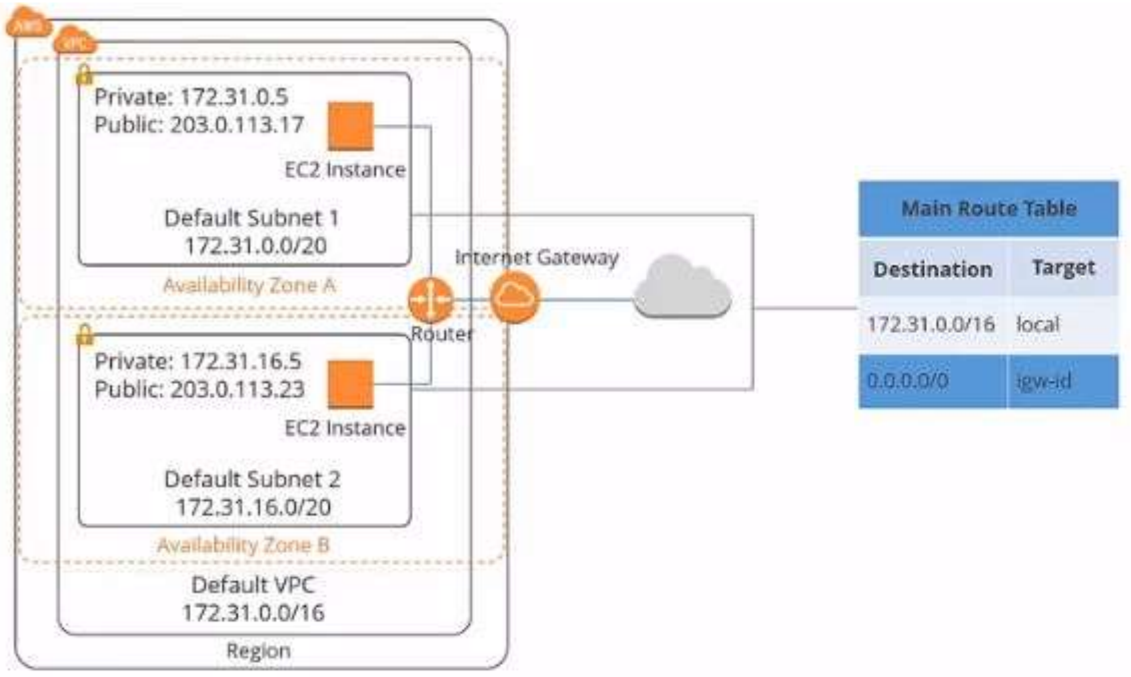
13. Select the instance named **“MyServer**”

- In the **Instance state** menu, choose **Stop instance**.
- In the **Stop instance** window, choose **Stop**.
- The instance will enter the *stopping* state and will shutdown.

b) Cloud Networking with Amazon VPC

Objective:

Learn to create and configure a Virtual Private Cloud (VPC) in AWS, including subnets, route tables, Internet gateways, and security groups, to enable and manage network communication in the cloud.



Activity-1: Creating VPC

Step 1: Create a VPC

1. **Sign in to AWS Management Console**:
   - Go to the [AWS Management Console](https://aws.amazon.com/console/).
   - Log in with your AWS credentials.
2. **Navigate to the VPC Dashboard**:
   - In the **Services** menu, search for and select **VPC**.
3. **Create a New VPC**:
   - In the VPC Dashboard, click **Create VPC**.
   - Provide a **Name tag** (e.g., `LabVPC`).
   - Choose an **IPv4 CIDR block** (e.g., `10.0.0.0/16`).
   - Leave other options as default and click **Create VPC**.

Step 2: Create Subnets

1. **Create Subnets**:
   - Navigate to **Subnets** and click **Create Subnet**.
   - Select the VPC you just created.
   - Create two subnets:
     - **Public Subnet**: Assign a CIDR block (e.g., `10.0.1.0/24`).
     - **Private Subnet**: Assign a CIDR block (e.g., `10.0.2.0/24`).
   - Ensure each subnet is in different availability zones for high availability.
   - Click **Create Subnet**.

Step 3: Set Up an Internet Gateway

1. **Create an Internet Gateway**:
   - Navigate to **Internet Gateways** and click **Create Internet Gateway**.
   - Provide a name (e.g., `LabInternetGateway`) and click **Create**.
2. **Attach the Internet Gateway to the VPC**:
   - Select the Internet Gateway you just created.
   - Click **Actions** > **Attach to VPC**.
   - Select your VPC and click **Attach Internet Gateway**.

Step 4: Configure Route Tables

1. **Create a Route Table for Public Subnet**:
   - Navigate to **Route Tables** and click **Create Route Table**.
   - Provide a name (e.g., `PublicRouteTable`) and select your VPC.
   - Click **Create Route Table**.
2. **Add a Route for Internet Access**:
   - Select the route table you just created.
   - Click **Routes** > **Edit Routes**.
   - Click **Add Route** and set:
     - Destination: `0.0.0.0/0`.
     - Target: The Internet Gateway created earlier.
   - Click **Save Routes**.
3. **Associate the Route Table with the Public Subnet**:
   - Click **Subnet Associations** > **Edit Subnet Associations**.
   - Select the public subnet and click **Save**.

Step 5: Configure Security Groups

1. **Create a Security Group for the Public Subnet**:
   - Navigate to **Security Groups** and click **Create Security Group**.

    - Provide a name (e.g., `PublicSG`) and description.
    - Select your VPC and click **Create Security Group**.
2. **Add Inbound Rules**:
    - Select the security group you just created.
    - Click **Inbound Rules** > **Edit Inbound Rules**.
    - Add rules to allow:
        - **SSH (port 22)** from your IP.
        - **HTTP (port 80)** from anywhere (`0.0.0.0/0`).
    - Click **Save Rules**.

Step 6: Launch an Instance in the Public Subnet

1. **Launch an EC2 Instance**:
    - Navigate to **EC2 Dashboard** > **Instances** > **Launch Instances**.
    - Choose an Amazon Machine Image (AMI) (e.g., Amazon Linux 2).
    - Select an instance type (e.g., t2.micro).
    - In the **Configure Instance** step:
        - Select the VPC and public subnet.
        - Ensure **Auto-assign Public IP** is enabled.
    - Proceed to configure storage, tags, and select the security group (`PublicSG`).
    - Review and launch the instance.
2. **Connect to the Instance**:
    - Once the instance is running, connect to it using SSH from your local machine.

Step 7: Test Internet Connectivity

1. **Test Internet Access from the EC2 Instance**:
    - From the SSH session, run commands like `ping google.com` or `curl http://example.com` to test internet access.

Activity-2: Creating Webserver using EC2 instance:

Step 1: Install the Apache Web Server

```
$sudo yum install httpd –y
```

Step 2: Start the Apache Service

```
$sudo systemctl start httpd
```

Verify the service is running:

```
$sudo systemctl status httpd
```

Step 3: Host a Simple Webpage

- Create an HTML file in the default Apache document root directory (/var/www/html):

```
$echo "Hello, AWS Networking!" | sudo tee /var/www/html/index.html
```

- Verify the file has been created

```
$cat /var/www/html/index.html
```

Step 4: Test the Web Server

- Open a browser on your local machine.
- Enter the Public IP of your EC2 instance in the address bar (e.g., http://<EC2-Public-IP>).
- If everything is set up correctly, you will see the message:

  Hello, AWS Networking!

Step 5: Stop the Apache service if no longer needed:

```
$sudo systemctl stop httpd
```