[version_1.1]

Note

The exercises in this course will have an associated charge in your AWS account. In this exercise, you create or use the following resources:

- AWS Identity and Access Management (IAM) policy and user (policies and users are AWS account features, offered at no additional charge)
- Amazon Elastic Compute Cloud (Amazon EC2) instance
- Virtual private cloud (VPC) with subnets and route tables

Familiarize yourself with **Amazon EC2 pricing** and the **AWS Free Tier**.

Exercise 4: Setting up a VPC

In this scenario, you create the underlying network infrastructure where the EC2 instance that hosts the employee directory will live.

In this exercise, you set up a new virtual private cloud (VPC). This new VPC will have four subnets (two public subnets and two private subnets) and two route tables (one public route table and one private route table). Then, you launch an EC2 instance inside the new VPC. Finally, at the end of the exercise, you stop the instance to prevent future costs from incurring.

Task 1: Creating the VPC

In this task, you will create a new VPC.

1. If needed, log in to the AWS Management Console as your *Admin* user.
2. In the **Services** search box, enter `VPC` and open the VPC console by choosing **VPC** from the list.
3. In the navigation pane, under **Virtual private cloud**, choose **Your VPCs**.
4. Choose **Create VPC**.
5. Configure these settings:
   - **Name tag**: `app-vpc`
   - **IPv4 CIDR block**: `10.1.0.0/16`
6. Choose **Create VPC**.
7. In the navigation pane, under **Virtual private cloud**, choose **Internet gateways**
8. Choose **Create internet gateway**.
9. For **Name tag**, paste `app-igw` and choose **Create internet gateway**.
10. In the details page for the internet gateway, choose **Actions** and then choose **Attach to VPC**.
11. For **Available VPCs**, choose `app-vpc` and then choose **Attach internet gateway**.

Task 2: Creating subnets

In this task, you will create the four subnets for your VPC. You will configure the two public subnets first, and then configure the two private subnets.

1. From the navigation pane, choose **Subnets**.
2. Choose **Create subnet**.
3. For the first public subnet, configure these settings:
   - **VPC ID**: *app-vpc*
   - **Subnet name**: `Public Subnet 1`
   - **Availability Zone**: Choose the first Availability Zone
     - Example: If you are in US West (Oregon), you would choose *us-west-2a*
   - **IPv4 CIDR block**: `10.1.1.0/24`
4. Choose **Add new subnet**.
5. For the second public subnet, configure these settings:
   - **Subnet name**: `Public Subnet 2`
   - **Availability Zone**: Choose the second Availability Zone
     - Example: If you are in US West (Oregon), you would choose *us-west-2b*
   - **IPv4 CIDR block**: `10.1.2.0/24`
6. Choose **Add new subnet** and for the first private subnet, configure these settings:
   - **Subnet name**: `Private Subnet 1`
   - **Availability Zone**: Choose the first Availability Zone
     - Example: If you are in US West (Oregon), you would choose *us-west-2a*
   - **IPv4 CIDR block**: `10.1.3.0/24`.
7. Choose **Add new subnet** and for the second private subnet, configure the following:
   - **Subnet name**: `Private Subnet 2`
   - **Availability Zone**: Choose the second Availability Zone
     - Example: If you are in US West (Oregon), you would choose *us-west-2b*
   - **IPv4 CIDR block**: `10.1.4.0/24`
8. Finally, choose **Create subnet**.
9. After the subnets are created, select the check box for **Public Subnet 1**.
10. Choose **Actions** and then choose **Edit subnet settings**.
11. For **Auto-assign IP settings**, select **Enable auto-assign public IPv4 address** and then choose **Save**.
12. Clear the check box for **Public Subnet 1** and select the check box for **Public Subnet 2**.
13. Again, choose **Actions** and then **Edit subnet settings**.
14. For **Auto-assign IP settings**, select **Enable auto-assign public IPv4 address** and save the settings.

Task 3: Creating route tables

In this task, you will create the route tables for your VPC.

First, you will create the public route table.

1. In the navigation pane, choose **Route Tables**.
2. Choose **Create route table**.
3. For the route table, configure these settings:
   - **Name**: `app-routetable-public`
   - **VPC**: *app-vpc*
4. Choose **Create route table**.
5. If needed, open the route table details pane by choosing **app-routetable-public** from the list.
6. Choose the **Routes** tab and choose **Edit routes**.
7. Choose **Add route** and configure these settings:
   - **Destination**: `0.0.0.0/0`
   - **Target**: *Internet Gateway*, then choose *app-igw* (which you set up in the VPC task)
8. Choose **Save changes**.
9. Choose the **Subnet associations** tab.
10. Scroll to **Subnets without explicit associations** and choose **Edit subnet associations**.
11. Select the two public subnets that you created (**Public Subnet 1** and **Public Subnet 2**) and choose **Save associations**.

    Next, you will create the private route table.

12. In the navigation pane, choose **Route Tables**.
13. Choose **Create route table** and configure these settings:
    - **Name**: `app-routetable-private`
    - **VPC**: *app-vpc*
14. Choose **Create route table**.

14. Choose **Create route table**.
15. If needed, open the details pane for **app-routetable-private** by choosing it from the list.
16. Choose the **Subnet associations** tab.
17. Scroll to **Subnets without explicit associations** and choose **Edit subnet associations**.
18. Select the two private subnets (**Private Subnet 1** and **Private Subnet 2**) and choose **Save associations**.

Task 4: Launching an EC2 instance that uses a role

Now that you have created a network, you will launch your EC2 instance by using the VPC that you created!

1. In the search box, enter `EC2` and open the Amazon EC2 console by choosing **EC2** from the list.
2. In the navigation pane, choose **Instances** and choose **Launch instances**.
3. For **Name** use `employee-directory-app`.
4. Under **Application and OS Images (Amazon Machine Image)**, choose the default **Amazon Linux 2023**.
5. Under **Instance type**, select **t2.micro**.
6. Under **Key pair (login)** choose the **app-key-pair** created in exercise-3.
7. Configure the following settings under **Network settings** and **Edit**.
   - **VPC**: *app-vpc*
   - **Subnet**: *Public Subnet 1*
   - **Auto-assign Public IP**: *Enable*
8. Under **Firewall (security groups)** choose **Create security group**. Use `web-security-group` as the **Security group name** and change **Description** to `Enable HTTP access`.
9. Under **Inbound security groups rules** choose **Remove** above the **ssh** rule.
10. Choose **Add security group rule**. For **Type** choose **HTTP**. Under **Source type** choose **Anywhere**.
11. Choose **Add security group rule**. For **Type** choose **HTTPS**. Under **Source type** choose **Anywhere**.
12. Expand **Advanced details** and under **IAM instance profile** choose **S3DynamoDBFullAccessRole**.
13. In the **User data** box, paste the following code:

```
#!/bin/bash -ex
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/DEV-AWS-MO-GCNv2/FlaskApp.zip
unzip FlaskApp.zip
cd FlaskApp/
yum -y install python3-pip
pip install -r requirements.txt
yum -y install stress
export PHOTOS_BUCKET=${SUB_PHOTOS_BUCKET}
export AWS_DEFAULT_REGION=<INSERT REGION HERE>
export DYNAMO_MODE=on
FLASK_APP=application.py /usr/local/bin/flask run --host=0.0.0.0 --port=80
```

14. Change the following line to match your Region (the Region is listed at the top right, next to your user name):

```
export AWS_DEFAULT_REGION=<INSERT REGION HERE>
```

    Example:

    This example uses the US West (Oregon) Region, or *us-west-2*.

```
export AWS_DEFAULT_REGION=us-west-2
```

    Note: You still don't need to change the `SUB_PHOTOS_BUCKET` variable in the user data script. You will update this placeholder in a later lab.

15. Choose **Launch instance**.
16. Choose **View all instances**.

    The instance should now be listed under **Instances**.

17. Wait for the **Instance state** to change to *Running* and the **Status check** to change to *2/2 checks passed*.

    Note: Often, the status checks update, but the console user interface (UI) might not update to reflect the most recent information. You can minimize waiting by refreshing the page after a few minutes.

18. Select the running **employee-directory-app** instance by selecting its check box.
19. On the **Details** tab, copy the **Public IPv4 address**.

    Note: Make sure that you only copy the address instead of choosing the **open address** link.

20. In a new browser window, paste the IP address that you copied. *Make sure to remove the 'S' after HTTP so you are using only HTTP instead.*
21. In a new browser window, paste the IP address that you copied.

You should see an **Employee Directory** placeholder. You won't be able to interact with the application yet because it's not connected to a database.

Task 5: Stopping the instance

Congratulations! You have launched an EC2 instance that hosts your employee directory application into a customized VPC.

To prevent future costs, you will now stop the instance.

Note: Do not terminate this instance because you will use it in a later exercise.

1. Return to the console, choose **Instance state**, and choose **Stop instance**.
2. In the dialog box, choose **Stop**.

The **Instance state** will eventually go into the *Stopped* state.