[version_1.0]

Note

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

- Application Load Balancer
- Amazon EC2 Auto Scaling group

Familiarize yourself with **Elastic Load Balancing pricing**, **AWS Auto Scaling pricing**, **Amazon EC2 pricing** and the **AWS Free Tier**.

Exercise 4: Implementing Elasticity

Scenario: In this challenge, you will launch a pre-written employee directory application on Amazon Elastic Compute Cloud (Amazon EC2). The application needs to be scalable and highly available. You are tasked with setting up an Amazon EC2 Auto Scaling group and an Application Load Balancer. You will then test your configuration by accessing the application. You will also test the availability of the application by loading a user data script to simulate stress on the instance.

Technical knowledge prerequisites

Remember that the following tasks are *requirements*. They are not steps. You must know how to create and connect the requested resources.

Similar to the exam, component names will be given for you to use. To meet the requirements on the exam, it is important that you use the resource names that are provided to you, where applicable.

Prerequisites

Before you can start this exercise, you will first need to set up a virtual private cloud (VPC) with the necessary requirements for a highly available environment.

Task 1: Create the load balancer

Before you can create the load balancer, you must have already created the following resources: - A **target group** with *instances* as the target type and `HTTP/80` as the protocol in your VPC - A **security group** that is set up to allow `HTTP/80`

1. The load balancer uses these requirements.
    - **Load balancer types**: *Application Load Balancer*
    - **VPC**: Your VPC
    - **Subnets**:
        - Your *PublicSubnet1*
        - Your *PublicSubnet2*

- **Security group**: Your security group
- **Region**: *us-east-1*
- **Listener**: `HTTP/80`
- **Forward to**: Your target group

Task 2: Creating the launch template

1. The launch template uses these requirements.

- **AMI**: *Amazon Linux 2 AMI (HVM), SDD Volume Type*
- **Instance type**: *t2.micro*
- **Region**: `us-east-1`
- **Key pair**: Don't include
- **Security group**: Your security group

- **User data**:

```
#!/bin/bash -ex
wget https://aws-tc-largeobjects.s3-us-west-2.amazonaws.com/DEV-AWS-MO-GCNv2/F
unzip FlaskApp.zip
cd FlaskApp/
yum -y install python3 mysql
pip3 install -r requirements.txt
amazon-linux-extras install epel
yum -y install stress
export PHOTOS_BUCKET=${SUB_PHOTOS_BUCKET}
export AWS_DEFAULT_REGION=us-east-1
export DYNAMO_MODE=on
FLASK_APP=application.py /usr/local/bin/flask run --host=0.0.0.0 --port=80
```

Task 3: Creating the Auto Scaling group

1. These are the requirements for the AutoScaling group:

    - **Launch template**: Template created in Task 2
    - **VPC**: Your VPC
    - **Subnets**:
        - Your *PublicSubnet1*
        - Your *PublicSubnet2*
    - **Load balancing**: Attach to existing load balancer created in Task 1
    - **Desired**: `1`
    - **Minimum**: `1`
    - **Maximum**: `2`
    - **Scaling policies**: *Target tracking scaling policy*
    - **Metric type**: *Average CPU utilization*
    - **Target value**: `80`

Task 4: Simulating stress and scaling the application

1. Test the application by visiting the public DNS name, and then appending `/info` to the end of the application URL.

2. Stress the CPU for 10 minutes.

Example: `http://ec2-192-0-2-0.us-east-2.compute.amazonaws.com/info`

Note: This application will not allow you to add employees because the database is not set up. For now, use this application to only explore how Amazon EC2 Auto Scaling works.

Task 5: Verifying the scaling action

1. Verify that scaling has occurred by viewing your instances.

Cleaning up

In this task, you delete the AWS resources that you created for this exercise.

1. Open the **Amazon EC2** console.
    - In the navigation pane, choose **Auto Scaling Groups** and delete your Auto Scaling group.
    - In the navigation pane, choose **Load Balancers** and delete your Application Load Balancer.
    - Delete your target group.
    - In the navigation pane, choose **Instances** and delete your launch template.