

## 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
- Elastic Load Balancing (ELB) load balancer
- AWS Auto Scaling
- Amazon Simple Notification Service (Amazon SNS) topic

Familiarize yourself with [Amazon EC2 pricing](#), [Elastic Load Balancing pricing](#), [AWS Auto Scaling pricing](#), [Amazon SNS pricing](#), and the [AWS Free Tier](#).

# Exercise 7: Load Balancing and Auto Scaling

For this scenario, you are tasked with setting up an ELB load balancer and an Auto Scaling group so that your application can scale horizontally.

In this exercise, you first launch another EC2 instance. You then create an Application Load Balancer and a launch template. Next, you set up an Auto Scaling group that uses the load balancer and launch template that you created. Finally, you test and stress the application, and watch your application scale in real time.

## Task 1: Launching an EC2 instance

In this task, you will launch an EC2 instance that hosts the application.

1. If needed, log in to the AWS Management Console as your *Admin* user.
2. Search for and open **EC2**.
3. In the navigation pane, choose **Instances**.
4. Select the check box for the **employee-directory-app-exercise6** instance, which should be in the *Stopped* state.
5. Choose **Actions** and then choose **Image and templates**, **Launch more like this**.
6. For **Name** and at the end of the **Value**, append `-exercise7`.

Example:

7. For **Key pair name**, select **app-key-pair**.
8. Under **Network settings** and **Auto-assign Public IP**, choose **Enable**.
9. Choose **Launch instance**.
10. Choose **View all instances**.

The instance should now be in the **Instances** list.

11. Wait for the **Instance state** to change to *Running* and the **Status check** to change to *2/2 checks passed*.
12. Select the check box for **employee-directory-app-exercise7**.
13. On the **Details** tab, copy the **Public IPv4 address** and paste it into a new browser window.
14. 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.*

## Task 2: Creating the Application Load Balancer

In this task, you will create the Application Load Balancer.

1. Return to the Amazon EC2 console.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Choose **Create Load Balancer**.
4. On the **Application Load Balancer** card, choose **Create**.
5. Configure the following load balancer settings.
  - **Load balancer name**: app-alb
  - **VPC**: *app-vpc*
  - **Mappings**: Select both Availability Zones
    - Example: If you are in US West (Oregon), you would select both *us-west-2a* and *us-west-2b*
  - **First Availability Zone Subnet**: *Public Subnet 1*
  - **Second Availability Zone Subnet**: *Public Subnet 2*
6. In the **Security groups** section, remove the default security group (by choosing the **X**) and choose **Create new security group**.

A new window opens for creating a security group.

7. Configure the following security group settings:
  - **Security group name**: load-balancer-sg
  - **Description**: HTTP access
  - **VPC**: If needed, paste the VPC ID for *app-vpc* and choose it when it appears under the box

- **Note:** You can find the *app-vpc* ID by opening the VPC console in a new window
  - **Inbound rules:** *Add Rule*
    - **Type:** *HTTP*
    - **Source:** *Anywhere-IPv4*
8. Choose **Create security group**.
  9. Close the security group browser window or return to the **Load balancers** window.
  10. For **Security groups**, add the new **load-balancer-sg** group. **Note:** To see the new security group, you might need to refresh the **Security groups** list.
  11. In **Listeners and routing**, choose **Create target group**.

A new window opens for creating a target group.
  12. For **Specify group details**, configure the following settings.
    - **Choose a target type:** Keep *Instances* selected
    - **Target group name:** *app-target-group*
    - **Health checks:** Expand *Advanced health check settings* and configure the following:
      - **Healthy threshold:** 2
      - **Unhealthy threshold:** 5
      - **Timeout:** 30
      - **Interval:** 40
  13. Choose **Next**.
  14. For **Register targets**, select **employee-directory-app-exercise7** and choose **Include as pending below**.
  15. Choose **Create target group**.
  16. Close the target groups window or return to the **Load balancers** window.
  17. Under **Listeners and routing**, refresh the available listener and choose **app-target-group**.
  18. Finally, choose **Create load balancer**.
  19. Choose **View load balancer**.
  20. Make sure that **app-alb** is selected and wait for the load balancer **State** to become *Active*.
  21. On the **Description** tab, copy **DNS name** and paste it into a text editor of your choice.
  22. In the text editor, at the beginning of the URL, add `http://`.

Example:

`http://app-elb-123456789012.us-west-2.elb.amazonaws.com`

23. Copy the DNS name (with `http://` added) and paste it into a new browser window.

You should see the employee directory application.

## Task 3: Creating the launch template

Now that you can access your application from a singular DNS name, you can scale the application horizontally. To scale horizontally, you need a launch template. In this task, you will create a launch template.

1. Back in the console, if needed, search for and open **EC2**.
2. In the navigation pane, under **Instances**, choose **Launch Templates**.
3. Choose **Create launch template** and configure the following settings.
  - **Launch template name**: app-launch-template
  - **Template version description**: A web server for the employee directory application
  - **Auto Scaling guidance**: *Provide guidance to help me set up a template that I can use with EC2 Auto Scaling*
  - **Application and OS Images (Amazon Machine Image) - required**: *Currently in use*
  - **Instance type**: *t2.micro*
  - **Key pair name**: *app-key-pair*
  - **Security groups**: *web-security-group*
4. Expand the **Advanced details** section.
5. For **IAM instance profile**, choose **S3DynamoDBFullAccessRole**.
6. Scroll to **User data** and 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 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=<INSERT REGION HERE>
export DYNAMO_MODE=on
FLASK_APP=application.py /usr/local/bin/flask run --host=0.0.0.0 --port=80
```

7. In the user data code, replace the `PHOTOS_BUCKET` placeholder value with the name of your bucket.

Example:

```
export PHOTOS_BUCKET=employee-photo-bucket-a1-907
```

8. Replace the `AWS_DEFAULT_REGION` placeholder value with your Region (the Region is listed at the top right, next to your user name).

Example:

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

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

9. Choose **Create launch template**.
10. Choose **View Launch templates**.

## Task 4: Creating the Auto Scaling group

In this task, you will create the Auto Scaling group.

1. In the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
2. Choose **Create Auto Scaling group**.
3. For **Choose launch template or configuration**, configure these settings:
  - **Auto Scaling group name:** *app-asg*
  - **Launch template:** *app-launch-template*
4. Choose **Next**.
5. For **Choose instance launch options**, configure these settings:
  - **VPC:** *app-vpc*
  - **Availability Zones and subnets:** Choose the Availability Zones with *Public Subnet 1* and *Public Subnet 2*
6. Choose **Next**.
7. For **Configure advanced options**, use these settings:
  - **Load balancing:** *Attach to an existing load balancer*
  - **Attach to an existing load balancer:** *Choose from your load balancer target groups*
  - **Existing load balancer target groups:** *app-target-group*
  - **Health checks:** *ELB*
8. Choose **Next**.
9. For **Configure group size and scaling policies**, use these settings:
  - **Desired capacity:** *2*
  - **Minimum capacity:** *2*
  - **Maximum capacity:** *4*
  - **Scaling policies:** *Target tracking scaling policy*
  - **Target value:** *60*
  - **Instances need:** *300*
10. Choose **Next**.
11. For **Add notifications**, choose **Add notification** and configure these settings:

- **SNS Topic:** *Create a topic*
- **Send a notification to:** `app-sns-topic`
- **With these recipients:** Enter your email address

12. Choose **Next** and then choose **Next** again.

13. Choose **Create Auto Scaling group**.

You should receive an **AWS Notification - Subscription Confirmation** email.

14. Open this email message and choose **Confirm subscription**.

A web browser window should open with a *Subscription confirmed!* message.

## Task 5: Testing the application

In this task, you will stress-test the application and confirm that it scales.

1. Return to the Amazon EC2 console.
2. In the navigation pane, under **Load Balancing**, choose **Target Groups**.
3. Make sure that **app-target-group** is selected and choose the **Targets** tab.

You should see two additional instances launching.

4. Wait until the **Status** for both instances is *healthy*.
5. In the navigation pane, choose **Load Balancers** and make sure that **app-alb** is selected.
6. Again, copy the **DNS name** and paste it into a text editor of your choice.
7. In the text editor, at the beginning of the URL, add `http://` and copy the modified URL.

Example:

```
http://app-elb-123456789012.us-west-2.elb.amazonaws.com
```

8. In a new browser window, paste the URL.
9. At the end of the URL, append `/info`.

Example:

```
http://app-alb-123456789012.us-west-2.elb.amazonaws.com/info
```

You should see an **Instance Info** page, which shows which **instance\_id** and **availability\_zone** you are being routed to.

10. Refresh the page a few times. Each time, note that the values for **instance\_id** or **availability\_zone** can be different from the previous ones.

Now, you need to test auto scaling by stressing the CPU of the instance.

11. For **Stress cpu**, choose **10 min**.

The top of the browser window should show a message that says *Stressing CPU*.

12. Wait for 10 minutes and after the 10 minutes are over, return to the Amazon EC2 console window.

13. In the navigation pane, under **Load Balancing**, choose **Target Groups**.

14. Select **app-target-group** and choose the **Targets** tab.

You should see additional instances were launched because of the stress test. You should also see a notification email.

## Task 6: Deleting the course resources

In this task, you will delete all the resources that you created in your AWS account so that you don't incur additional costs.

1. Delete the Auto Scaling group.
  - In the Amazon EC2 navigation pane, choose **Auto Scaling groups**.
  - Select **app-asg** and choose **Delete**.
  - In the box, enter `delete` and choose **Delete**.
2. Delete the Application Load Balancer.
  - In the Amazon EC2 navigation pane, choose **Load Balancers**.
  - Select **app-elb**, choose **Actions**, and then choose **Delete**.
  - Confirm the deletion by choosing **Yes, Delete**.
3. Delete the target group.
  - In the Amazon EC2 navigation pane, choose **Target Groups**.
  - Select **app-target-group**, choose **Actions** and then choose **Delete**.
  - Confirm the deletion by choosing **Yes, delete**.
4. Terminate all the EC2 instances that you created during this course.
  - In the Amazon EC2 navigation pane, choose **Instances**.
  - Select all the EC2 instances that you created for this course (all instances that start with *employee-directory-app*)
  - Choose **Instance State** and then choose **Terminate instance**.
  - Confirm the deletion by choosing **Terminate**.
5. Delete the DynamoDB table.
  - Return to the DynamoDB console.
  - In the navigation pane, choose **Tables**.
  - Select the **Employees** table.
  - Choose **Delete**.
  - Confirm the deletion by entering `delete` and choosing **Delete table**.
6. Delete the S3 bucket.
  - Return to the Amazon S3 console.
  - Choose the radio button for **employee-photo-bucket-**.
  - Choose **Empty**.
  - In the box, enter `permanently delete` and choose **Empty**.
  - In the message at the top of the window, choose **delete bucket configuration**.
  - In the box, paste the bucket name and choose **Delete bucket**.
7. Delete the route tables.
  - In the console, return to the VPC dashboard.
  - In the navigation pane, choose **Route Tables**.
  - Select **app-routetable-public** and choose the **Subnet Associations** tab.

- Choose **Edit subnet associations**.
  - Clear the following check boxes:
    - **Public Subnet 1**
    - **Public Subnet 2**
  - Choose **Save associations**.
  - Select **app-routetable-public** again, choose **Actions**, and choose **Delete route table**.
  - Confirm the deletion by entering `delete` , and then choose **Delete**.
  - Repeat the previous steps to delete **app-routetable-private**.
8. Delete the internet gateway.
- In the VPC dashboard navigation pane, choose **Internet Gateways**.
  - Select **app-igw**, choose **Action**, and choose **Detach from VPC**.
  - In the dialog box, choose **Detach internet gateway**.
  - Select **app-igw** again, choose **Actions**, and choose **Delete internet gateway**.
  - Confirm the deletion by entering `delete` , and choose **Delete internet gateway**.
9. Delete the subnets.
- In the VPC dashboard navigation pane, choose **Subnets**.
  - Select the following subnets:
    - **Public Subnet 1**
    - **Public Subnet 2**
    - **Private Subnet 1**
    - **Private Subnet 2**
  - Choose **Actions** and then choose **Delete subnet**.
  - Confirm the deletion by entering `delete` , and choose **Delete**.
10. Delete the VPC.
- In the VPC dashboard navigation pane, choose **Your VPCs**.
  - Select the **app-vpc**, choose **Actions**, and choose **Delete VPC**.
  - Confirm the deletion by entering `delete` , and then choose **Delete**.
11. If needed, delete the security groups.
- Return to the Amazon EC2 console.
  - In the navigation pane, choose **Security Groups**.
  - Select the following security groups:
    - **app-sg**
    - **load-balancer-sg**
  - Choose **Actions** and then choose **Delete security groups**.
  - Confirm the deletion by choosing **Delete**.
12. Delete the IAM role that you created.
- Return to the IAM console.
  - In the navigation pane, choose **Roles** and search for `S3DynamoDB` .
  - Select **S3DynamoDBFullAccessRole** and choose **Delete**.
  - In the **To confirm deletion** box, paste `S3DynamoDBFullAccessRole` , and then choose **Delete**.
  - You can also delete the IAM Admin user that you set up.
13. Delete the SNS topic.
- Open the Amazon SNS console.
  - In the navigation pane, choose **Topics**.
  - Choose the radio button for **app-sns-topic** and choose **Delete**.
  - Confirm the deletion by entering `delete me` , and then choose **Delete**.



Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections, feedback, or other questions? Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>. All trademarks are the property of their owners.