**Set up an IAM role for EC2 instance**

1. Now that you are logged in as the Admin user, search for **IAM** again in the service search bar. On the left side panel, choose **Roles**. Then, choose **Create role**.
2. Choose **AWS service**. Choose **EC2**. Choose **Next: Permissions**.
3. Next to **Filter policies**, search for amazons3full and choose **AmazonS3FullAccess**.
4. Next to **Filter policies** search for amazondynamodb and choose **AmazonDynamoDBFullAccess**.
5. Choose **Next: Tags**. Choose **Next: Review**.
6. For **Role name** paste in S3DynamoDBFullAccessRole. Choose **Create role**.

### Launch EC2 instance using role

1. Search for **EC2** in the search bar at the top. Choose **EC2**.
2. Select **Instances** on the left side panel, and then choose the **Launch instance** button.
3. Choose **Select** next to the first AMI, which should be **Amazon Linux 2 AMI (HVM), SSD Volume Type**.
4. Choose the **t2.micro** (Free tier eligible) as the **Type**. Choose **Next: Configure Instance Details**.
5. Leave the **Network** as the (default). Next to **Subnet**, choose the first subnet in the drop down list.
6. Next to **Auto-assign Public IP** choose **Enable**.
7. Next to **IAM role** choose the **S3DynamoDBFullAccessRole**.
8. Scroll down to **Advanced Details**

Script::

#!/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

export AWS\_DEFAULT\_REGION=us-west-2

 Choose **Next: Add Storage**. Choose **Next: Add Tags**.

 Choose **Add Tag**. Under **Key** paste in Name. Under **Value** paste in employee-directory-app.

 Choose **Next: Configure Security Group**. For **Security group name:** paste in app-sg.

 Choose **Add Rule**. For **Type** choose **HTTP**. For **Source** change to **Anywhere**. Then, next to the **SSH** rule, choose the **X** at the right to remove it as you will not need SSH access to the instance.

Note: You may get a warning that you will no longer be able to SSH into your instance. This is fine - as you won't need that functionality for this course.

 Choose **Review and Launch**. Choose **Launch**. Choose **Create a new key pair**. Under **Key pair name** paste in app-key-pair. Choose **Download Key Pair**. Finally choose **Launch instances**.

 Scroll down, and choose **View Instances**. The instance should now show up under **Instances**. 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 and the UI does not. Feel free to refresh the page after a few minutes to minimize waiting.

 Next to **Name**, choose the checkbox to select the instance. Under the **Details** tab, copy down the **Public IPv4 address**.

Note: do not click the link to open the IPv4 address. Simply just copy the address and paste it into a new tab.

 Paste it into a new browser tab/window. You should see a **Employee Directory** placeholder. Right now you will not be able to interact with it as it's not currently connected to our DynamoDB database.

Congrats! You've successfully created an EC2 instance hosting the employee directory application. After you've finished looking around, it's time to stop and terminate your instance, so that you don't incur future costs.

 Back in the AWS Management Console, the employee-directory-app should still be selected. Now, choose **Instance state** at the top and choose **Stop instance**. Choose **Stop**. The **Instance state** will eventually go into the **Stopped** state

### Create VPC

1. Search for **VPC** in the search bar at the top. Choose **VPC**.
2. Choose **Your VPCs** in the left panel. Choose **Create VPC**. Under **Name tag** paste in app-vpc.
3. For the **IPv4 CIDR block** paste in 10.1.0.0/16. Choose **Create VPC**.
4. Choose **Internet Gateways** in the left panel. Choose **Create internet gateway**. Under **Name tag** paste in app-igw. Choose **Create internet gateway**.
5. Choose **Actions** and **Attach to VPC**. Under **Available VPCs** choose the app-vpc. Choose **Attach internet gateway**

### Create Subnets

1. Choose **Subnets** at the left. Choose **Create subnet**. Under **VPC ID**, select the app-vpc from the drop down list.
2. Under **Subnet settings** and **Subnet name**, paste in Public Subnet 1. Under **Availability Zone**, choose the 1st AZ.

Example: If you are in US West (Oregon) you would choose **us-west-2a**.

1. For the **IPv4 CIDR block** paste in 10.1.1.0/24.
2. Choose **Add new subnet**. Under **Subnet name**, paste in Public Subnet 2. Under **Availability Zone**, choose the 2nd AZ.

Example: If you are in US West (Oregon) you would choose **us-west-2b**.

1. For the **IPv4 CIDR block** paste in 10.1.2.0/24.
2. Choose **Add new subnet**. Under **Subnet name**, paste in Private Subnet 1. Under **Availability Zone,** choose the 1st AZ.

Example: If you are in US West (Oregon) you would choose **us-west-2a**.

1. For the **IPv4 CIDR block** paste in 10.1.3.0/24.
2. Choose **Add new subnet**. Under **Subnet name**, paste in Private Subnet 2. Under **Availability Zone**, choose the 2nd AZ.

Example: If you are in US West (Oregon) you would choose **us-west-2b**.

1. For the **IPv4 CIDR block** paste in 10.1.4.0/24.
2. Finally choose **Create subnet**.
3. Select the checkbox next to **Public Subnet 1** after the subnets have been created. Choose **Actions** and **Modify auto-assign IP settings**. Under **Auto-assign IPv4**, choose **Enable auto-assign public IPv4 address**. Choose **Save**.
4. De-select **Public Subnet 1**. Select the checkbox next to **Public Subnet 2**. Choose **Actions** and **Modify auto-assign IP settings**. Under **Auto-assign IPv4**, choose **Enable auto-assign public IPv4 address**. Choose **Save**.

### Create Route Tables

1. Choose **Route Tables** at the left. Choose **Create route table**. Under **Name tag**, paste in app-routetable-public. Under **VPC**, choose the app-vpc. Choose **Create**. Choose **Close**.
2. Select the app-routetable-public from the list. Choose the **Routes** tab. Choose **Edit routes**.
3. Choose **Add route**. For **Destination**, paste in 0.0.0.0/0. For **Target** choose **Internet Gateway**. Choose the app-igw you set up in the VPC section. Select **Save routes**. Choose **Close**.
4. Choose the **Subnet Associations** tab. Choose **Edit subnet associations**. Select the 2 Public subnets (Public Subnet 1 & Public Subnet 2) you created in the Subnet section. Choose **Save**.
5. Choose **Create route table**. Under **Name tag** paste in app-routetable-private. Under **VPC** chose the app-vpc. Choose **Create**. Choose **Close**.
6. Deselect the app-routetable-public. Select the app-routetable-private from the list. Choose the **Subnet Associations** tab. Choose **Edit subnet associations**. Select the 2 Private subnets (Private Subnet 1 & Private Subnet 2) you created in the Subnet section. Choose **Save**.

### Launch EC2 instances using role

Now that you've created a network, it's time to launch your EC2 instance using the VPC you created!

1. Search for **EC2** in the search bar at the top. Choose **EC2**.
2. Under **Launch instance** choose the **Launch instance** button.
3. Choose **Select** next to the first AMI which should be **Amazon Linux 2 AMI (HVM), SSD Volume Type**.
4. Choose the **t2.micro** (Free tier eligible) as the **Type**. Choose **Next: Configure Instance Details**.
5. Next to **Network** choose the app-vpc from the list. Next to **Subnet** choose Public Subnet 1 from the list.
6. Next to **Auto-assign Public IP** choose **Enable**.
7. Next to **IAM role** choose the **S3DynamoDBFullAccessRole**.
8. Scroll down to **Advanced Details**
9.  Choose **Next: Add Storage**. Choose **Next: Add Tags**.
10.  Choose **Add Tag**. Under **Key** paste in Name. Under **Value** paste in employee-directory-app.
11.  Choose **Next: Configure Security Group**. For **Security group name:** paste in web-security-group.
12.  For the **Description** paste in Enable HTTP access.
13.  Choose **Add Rule**. For **Type**, choose **HTTP**. For **Source**, change to **Anywhere**. Choose **Add Rule**. For **Type**, choose **HTTPS**. For **Source**, change to **Anywhere**. Then, next to the **SSH** rule, choose the **X** at the right to remove it as you will not need SSH access to the instance.
14.  Choose **Review and Launch**. Choose **Launch**. Leave **Choose an existing key pair** selected. Under **Select a key pair** app-key-pair should be selected. Select the checkbox next to the acknowledgement. Choose **Launch Instances**.
15.  Choose **View Instances**. The instance should now show up under **Instances**. Wait for the **Instance state** to change to **Running** and the **Status check** to change to **2/2 checks passed**.
16. Note: Often, the status checks update and the UI does not. Feel free to refresh the page after a few minutes to minimize waiting.
17.  Next to **Name**, choose the checkbox to select the running employee-directory-app instance. Under the **Details** tab, copy down the **Public IPv4 address**.
18. Note: do not click the link to open the IPv4 address. Simply just copy the address and paste it into a new tab.
19.  Paste it into a new browser tab/window. You should see a **Employee Directory** placeholder. Right now, you will not be able to interact with it as it's not currently connected to a database

### Stop instance

### Create an S3 bucket

1. Search for **S3** in the search bar at the top. Choose **S3**.
2. Choose **Create bucket**.
3. For the **Bucket name** name it employee-photo-bucket- then use your initials and a unique number.
4.  Make sure the **Region** is the region where you have created the other services. Again, this can be found at the top right.
5.  Choose **Create bucket**

### Upload a photo

1. Choose your newly created bucket by clicking on the name of your bucket.
2. Choose **Upload**.
3. Choose **Add files**. Choose a photo of your choice on your computer.
4. Choose **Upload**.
5. At the top, you should see **Upload succeeded** in green. Choose **Exit**.

### Modify the S3 bucket policy

1. Choose the **Permissions** tab. Scroll down to **Bucket policy**.
2. Choose **Edit**. Paste in the following policy:

{

   "Version": "2012-10-17",

   "Statement": [

      {

           "Sid": "AllowS3ReadAccess",

           "Effect": "Allow",

           "Principal": {

               "AWS": "arn:aws:iam::**<INSERT-ACCOUNT-NUMBER>**:role/S3DynamoDBFullAccessRole"

          },

           "Action": "s3:\*",

           "Resource": [

               "arn:aws:s3**:::<INSERT-BUCKET-NAME>**",

               "arn:aws:s3:::**<INSERT-BUCKET-NAME>**/\*"

          ]

      }

  ]

}

### Modify the application to use the S3 bucket

1. Search for **EC2** in the search bar at the top. Choose **EC2**.
2. Choose **Instances** under **Instances** at the left side panel.
3. Select the employee-directory-app instance. Which should be in the **Stopped** state.
4. Choose **Actions**. **Image and templates** and **Launch more like this**.
5. At the top, choose **3. Configure instance**.
6. Next to **Auto-assign Public IP**, choose **Enable**.
7. Scroll down to **Advanced Details**. And next to **User data**. You will update the **PHOTOS\_BUCKET** and **AWS\_DEFAULT\_REGION** variable.

#!/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**=<INSERT-BUCKET-NAME-HERE>**

export AWS\_DEFAULT\_REGION=**<INSERT-REGION-NAME-HERE>**

export DYNAMO\_MODE=on

FLASK\_APP=application.py /usr/local/bin/flask run --host=0.0.0.0 --port=80

 Choose **Review and Launch**. Choose **Launch**.

 Leave the app-key-pair selected under **Select a key pair**. Select the acknowledgement.

 Choose **Launch Instances**.

 Choose **View Instances**. The instance should now show up under **Instances**. Wait for the **Instance state** to change to **Running** and the **Status check** to change to green.

 Next to **Name**, choose the checkbox to select the employee-directory-app-s3 instance. Under the **Details** tab copy down the **Public IPv4 address**.

Note: Do not use the link to open the IPv4 address. Instead, copy only the address and paste it into a new browser.

 Paste it into a new browser tab/window. You should see a **Employee Directory** placeholder. Right now you will not be able to interact with it as it's not currently connected to the database

### Create DynamoDB table

To connect the app to a database, you first need to create one! To do this, you'll use DynamoDB.

1. Search for **DynamoDB** in the search bar at the top. Choose **DynamoDB**.
2. At the left choose **Tables**. Choose **Create table**.
3. For the **Table name** paste in Employees. For the **Primary key** paste in id.
4. Choose **Create**.

### Test the application

1. Search for **EC2** in the search bar at the top. Choose **EC2**.
2. Under **Instances (running)** select the employee-directory-app-lab6 instance.
3. Under the **Details** tab copy down the **Public IPv4 address**.
4. Choose **Add**. Create a new employee entry by typing in a name, location and job title, as well as selecting attributes. Then, choose **Choose File**. Feel free to upload a picture of your choice. Choose **Save**.

**Note:** Feel free to create a few entries, as well as edit and delete entries.

You should now be seeing the list of employees you added on the employee directory application! Great job.

### Stage 4 - View the item in the database

Let's see how these employees are stored in DynamoDB.

1. Back in the AWS Mangement Console, search for **DynamoDB** in the search bar at the top. Choose **DynamoDB**.
2. Choose **Tables**. Choose the **Employees** table.
3. Choose the **Items** tab. You can now see the entries in the database made from the EC2 application.

### Launch an EC2 Instance

1. Search for **EC2** in the search bar at the top. Choose **EC2**.
2. Choose **Instances** under **Instances** at the left side panel.
3. Select the employee-directory-app-lab6 instance. Which should be in the **Stopped** state.
4. Choose **Actions**. **Image and templates** and **Launch more like this**.
5. At the top, choose **3. Configure instance**.
6. Next to **Auto-assign Public IP**, choose **Enable**.
7. At the top, choose **5. Add Tags**. Remove lab6 at the end of the **value** and append lab7 instead.

employee-directory-app-

1. Leave the app-key-pair selected under **Select a key pair**. Select the acknowledgement.
2. Choose **Launch Instances**.
3. Choose **View Instances**. The instance should now show up under **Instances**. Wait for the **Instance state** to change to **Running** and the **Status check** to change to **2/2 checks passed**.
4. Next to **Name**, choose the checkbox to select the employee-directory-app-lab6 instance. Under the **Details** tab copy down the **Public IPv4 address**.
5. Paste it into a new browser tab/window. You should see a **Employee Directory** placeholder. Right now you will not be able to interact with it as it's not currently connected to the database.

### Stage 1 - Create Load Balancer

1. In the EC2 console, on the left side panel under **Load Balancing**, choose **Load Balancers**.
2. Choose **Create Load Balancer**. Choose **Create** under the **Application Load Balancer**.
3. For the **Name** paste in app-elb. Under **Availability Zones** for the **VPC**, choose the app-vpc. Select both **Availability Zones**.

Example: If you are in US West (Oregon) you would choose both **us-west-2a** and **us-west-2b**.

1. Choose **Select a subnet** and choose Public Subnet 1 for the first AZ and Public Subnet 2 for the second AZ.
2. Choose **Next: Configure Security Groups**.
3. Again choose **Next: Configure Security Groups**. Next to **Assign a security group** choose **Create a new security group**. Next to **Security group name** paste in load-balancer-sg.
4. Remove the existing rule by clicking on the **X** at the right.
5. Choose **Add Rule**. For the **Type** choose **HTTP**. For **Source** choose **Anywhere**.
6. Choose **Next: Configure Routing**.
7. Under **Target group** next to **Name** paste in app-target-group.
8. Under **Health checks** expand **Advanced health check settings**.
9. Change the **Healthy threshold** to 2. Change the **Unhealthy threshold** to 5.
10. Change the **Timeout** to 30. Change the **Interval** to 40. Choose **Next: Register Targets**.
11. Under **Instances** select the checkbox next to the employee-directory-app-lab7 instance and choose **Add to registered**.
12. Choose **Next: Review**. Choose **Create**. Once it's successfully created choose **Close**.

**Note**: Wait for the **State** to change from provisioning to **active**. Refresh the page while you wait.

1. Select the app-elb and copy the **DNS** name and paste it into a notepad. Edit the DNS name to include an http:// in front of the URL, to look like the following:

http://app-elb-000000000.us-west-2.elb.amazonaws.com

1. Copy the DNS name with http:// and paste it into a new browser tab or window. You should see the employee directory application.

### Stage 2 - Create launch template

Now that you can access your application from a singular DNS name, it's time to horizontally scale! To do this, the first thing you need is a launch template.

1. Back in the AWS Management Console, search for **EC2** in the service search bar. Then, choose EC2. On the left side panel, choose **Launch Templates** under the **Instances** header.
2. Choose **Create launch template**. Under **Launch template name** paste in app-launch-template.
3. Under **Template version description** paste in A web server for the employee directory application.
4. Under **Auto Scaling guidance** check the box for **Provide guidance to help me set up a template that I can use with EC2 Auto Scaling**.
5. Under **AMI** choose the **Amazon Linux 2 AMI (HVM), SSD Volume Type**.
6. Under **Instance type** choose the **t2.micro**.
7. Under **Key pair name** choose the app-key-pair.
8. Under **Network settings** and **Security group** choose the web-security-group.
9. Expand **Advanced details** and under **IAM instance profile**. Choose S3DynamoDBFullAccessRole.
10. Scroll down to **User data** and paste in the following information

#!/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

Change the following line to match your region:

**Note:** You can find this at the top right next to your user name.

export AWS\_DEFAULT\_REGION=<INSERT REGION HERE>

export AWS\_DEFAULT\_REGION=us-west-2

export PHOTOS\_BUCKET=<INSERT\_BUCKET\_NAME>

### Create ASG

1. Choose **View Launch templates**. On the left side panel, under the **Auto Scaling** header, choose **Auto Scaling Groups**.
2. Choose **Create Auto Scaling group**. Under **Auto Scaling group name** paste in app-asg.
3. Under **Launch template** choose the app-launch-template. Choose **Next**.
4. Under **Network** and **VPC** choose the app-vpc. Choose Public Subnet 1 and Public Subnet 2. Choose **Next**.
5. Under **Load balancing** choose **Attach to an existing load balancer**. Under **Attach to an existing load balancer** choose **Choose from you load balancer target groups**.
6. Under **Existing load balancer target groups** choose **Select target groups**. Choose the app-target-group.
7. Under **Health checks** choose **ELB**. Choose **Next**.
8. Under **Group size** change the **Desired capacity** to 2.
9. Change the **Minimum capacity** to 2.
10. Change the **Maximum capacity** to 4.
11. Under **Scaling policies** choose **Target tracking scaling policy**.
12. Change the **Target value** to 60. Change the **Instances need** to 300. Choose **Next**.
13. Under **Add notifications** choose **Add notification**.
14. Under **SNS Topic** choose **Create a topic**. Under **Send a notification to:** paste in app-sns-topic. Under **With these recipients** paste in your email address. Choose **Next**.
15. Choose **Next**. Choose **Create Auto Scaling group**.
16. You should have received an **AWS Notification - Subscription Confirmation**. In the email, choose **Confirm subscription**.

### Test the application

1. Under **Load Balancing** on the left side panel, choose **Target Groups**. Select the name of the app-target-group. Choose the **Targets** tab. You should see two additional instances launching. Wait until the **Status** shows as healthy.
2. Under **Load Balancing** on the left side panel, select **Load Balancers** and select the app-elb. Again, copy the **DNS** name and append http:// to the beginnng of it.
3. Paste it into a new browser tab or window. Append /info to the URL
4.  You should be presented with **Instance Info** which will show you which **instance\_id** and **availability\_zone** you are being routed to. Refresh the page a few times and you will notice you are being routed to a different **instance\_id** and **availability\_zone**.
5.  Now, you will need to stress the CPU of the instance in order to test Auto Scaling.
6.  Next to **Stress cpu:** choose **10 min**. At the top you will notice it now says **Stressing CPU**.
7. **Note**: Go grab a beverage as you wait :)
8.  After the 10 minutes are up, go back in the EC2 Console. Under **Load Balancing** on the left side panel, choose **Target Groups**. Choose the app-target-group. Choose the **Targets** tab.

### Delete Course Resources

1. To ensure that you don't incur additional costs, please ensure that you delete all resources in the AWS account you created.
2. In the EC2 console, on the left side panel, choose **Auto Scaling groups**. Select the app-asg and select delete. Type **delete** in the text box to confirm your deletion.
3. In the EC2 console, on the left side panel, choose **Load Balancers**. Select the app-elb and select **Actions** and then **delete**. Choose **Yes, Delete**.
4. In the EC2 console, on the left side panel, choose Target Groups. Select the app-target-group and select **Actions** and then **delete**. Choose **Yes, Delete**
5. In the EC2 console, on the left side panel, choose Instances. Terminate all EC2 instances you created in this course. Select the checkbox next to the instances you want to delete (all instances starting with employee-directory-app) and then select **Instance State**. Then, choose **Terminate Instance**.
6. In the DynamoDB console, on the left side panel, click on **Tables**. Then, click the circle next to the Employees table. Choose **Delete Table**. Type in the word delete and select **Delete**.
7. In the S3 console, find the employee-photo-bucket. Select the circle next to the bucket you created. Then, choose **Empty**. Type in permanently delete in the text field. Select **Empty**. Choose **Exit**. Now, choose **Delete**. Type in the name of your bucket in the text field and choose **Delete bucket**.
8. In the VPC console, on the left side panel, click on Route Tables. Select the checkbox next to **app-route-table-public** and click on the **Subnet Associations** tab. Select **Edit subnet associations**. Deselect the boxes next to both Public Subnets and click **save**. Select **Actions** and then **Delete**. Then, choose Delete Route Table again. Follow the same steps for **app-route-table-private**.
9. In the VPC console, on the left side panel, click on **Internet Gateways.** Select the app-igw and choose **Detach from VPC** and choose **Detach Internet Gateway**. Then, click **Actions** and select **Delete Internet Gateway**. Type in delete in the text field and select **Delete Internet Gateway**.
10. In the VPC console, on the left side panel, select **Subnets**. Select the boxes next to Public Subnet 1, Public Subnet 2, Private Subnet 1, and Private Subnet 2. Select **Actions** and then **delete subnet**. Type delete in the text field. Then, choose **Delete**.
11. In the VPC console, on the left side panel, select **Your VPCs**. Select the checkbox next to app-vpc. Select **Actions** and then **Delete VPC**. Type delete in the text field and then choose **Delete**.