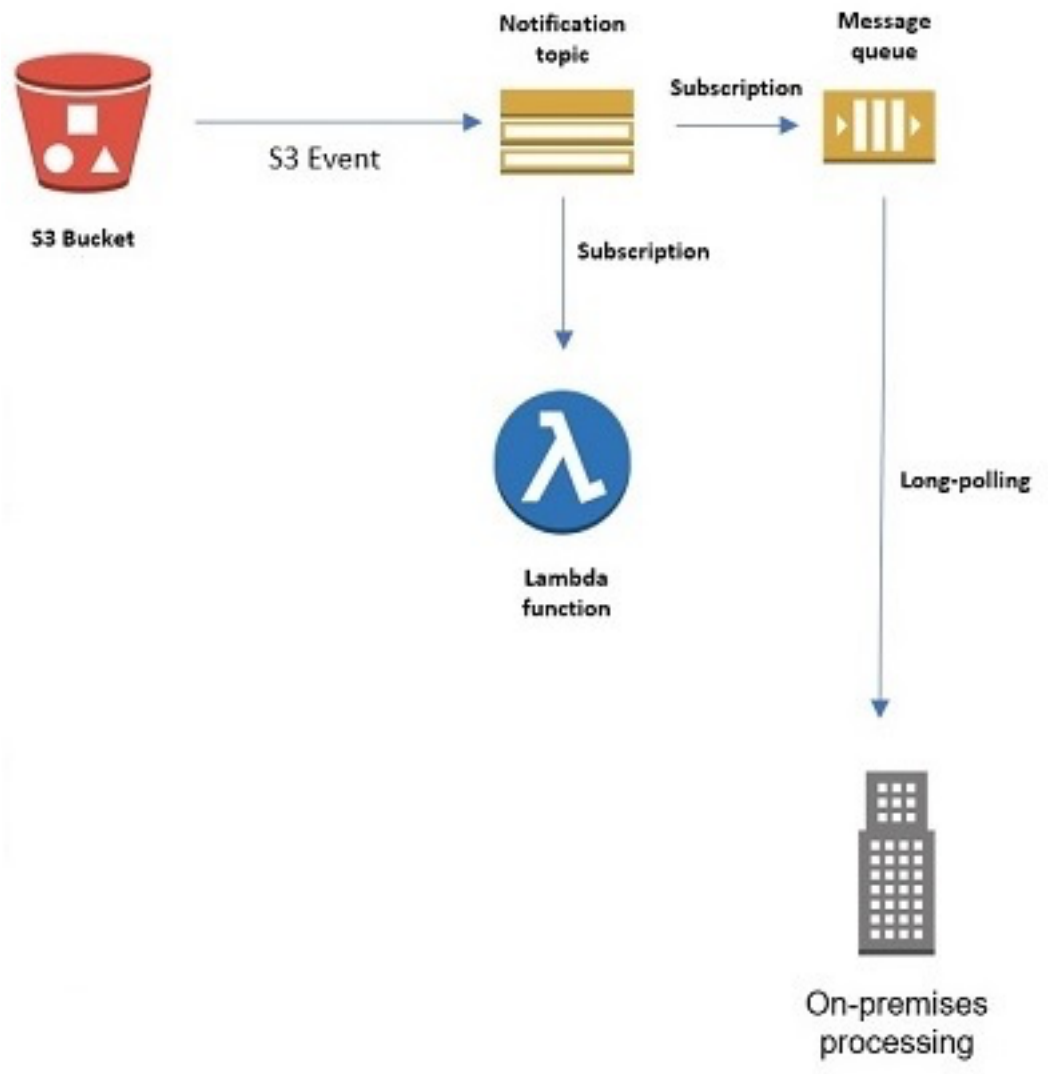
Audit Access Expires May 4, 2020
You lose all access to this course, including your progress, on May 4, 2020.
Upgrade by Jun 20, 2020 to get unlimited access to the course as long as it exists on the site. **Upgrade now**

Exercise 12

In this exercise, you will make your application more distributed by issuing an Amazon S3 bucket event notification to an Amazon SNS topic whenever a photo is uploaded to the bucket. This triggers the subscribed AWS Lambda function, which talks to Amazon Rekognition. Let's assume that you have an on-premises application doing additional processing of the uploaded photo. To achieve this in a distributed manner, you will create an Amazon SQS queue and subscribe it to the Amazon SNS topic. Whenever a photo is uploaded to ket, the Amazon SNS topic sends a notification and

the subscribed Amazon SQS queue stores the incoming request. The on-premises application will then poll the queue for processing as shown in the screenshot below.



Note: Make sure to sign-in to your AWS account with the AWS IAM user **edXProjectUser** credentials.

To begin, follow the steps below.

1. Start the RDS database instance.

- In the AWS Console, click **Services**, then click **Relational Database Service** to open the Amazon RDS dashboard.
- In the left navigation pane, click **Instances**. From the list of instances,

select **edx-photos-db**.

- At the top, click **Instance actions**, and then click **Start**.

2. Turn on the NAT instance.

To turn the NAT instance back on again, follow the steps below.

- In the AWS Management console, open the Amazon EC2 dashboard.
- In the navigation pane, click **Instances**. In the list of instances, select **edx-nat-instance**.
- Click **Actions -> Instance State -> Start**.

3. Create an Amazon SNS topic and update the topic permissions.

In this section, you will create an Amazon SNS topic and update its permissions to allow Amazon S3 to publish an event to the topic.

- In the AWS Console, click **Services**, then click **Simple Notification Service** to open the Amazon SNS dashboard.
- Click **Get started**.
- Make sure you are still in the **Oregon (us-west-2)** region.
- Click **Create topic**.
- For **Topic name**, type **uploads-topic**
- Click **Create topic**.
- On the left navigation menu, click **Topics**.
- Write down the topic ARN for the **uploads-topic** for later use.
- Select the topic and click **Actions -> Edit topic policy**.
- Click **Advanced view**.

- Copy the topic policy shown below, delete the default policy in the editor, and paste the one shown below.
  In the editor, replace **YOUR_TOPIC_ARN** with the topic ARN you made a note of earlier and replace **YOUR_BUCKET_NAME** with the Amazon S3 bucket name that stores your application photos.

```
{
        "Version": "2008-10-17",
        "Statement":
        [
                {
                        "Effect": "Allow",
                        "Principal":
                        {
                                "AWS": "*"
                        },
                        "Action": "SNS:Publish",
                        "Resource": "YOUR_TOPIC_ARN",
                        "Condition":
                        {
                                "ArnLike":
                                {
                                "aws:SourceArn":
"arn:aws:s3:::YOUR_BUCKET_NAME"
                                }
                        }
                }
        ]
}
```

- Click **Update policy**.

4. Update the Amazon S3 bucket event notification to publish an event to the Amazon SNS topic.

- In the AWS Console, click **Services**, then click **S3** to open the Amazon S3 dashboard.
- From the list of buckets, select the bucket that stores your application photos.
- Click the **Properties** tab.
- Scroll down to **Advanced settings** and click **Events**.
- Delete the event you created in the previous exercise for AWS Lambda by selecting the event and clicking **Delete**.
- Click **Add notification**.
- For **Events**, select **ObjectCreate (All)**.
- For **Send to**, select **SNS Topic**.
- For **SNS**, select **uploads-topic**.
- Click **Save**.

  Your Amazon S3 bucket will now publish an event to the Amazon SNS topic as soon as a photo is uploaded to the bucket.

5. Create an email subscription for the Amazon SNS topic.

- To test the Amazon SNS topic you created earlier and see how email subscription works, open the Amazon SNS dashboard in the AWS Console.
- In the left navigation menu, click **Topics**.
- Select **uploads-topic**.
- Click **Actions -> Subscribe to topic**.
- For **Protocol**, select **Email**.
- For **Endpoint**, type your email address.

- Click **Create subscription**.
- In the left navigation menu, click **Subscriptions**. You should see a subscription showing status as **PendingConfirmation**.
  You should receive a subscription confirmation email within a few minutes. Confirm your email address by following the instructions in the email.

6. Download and run the exercise code.

- Type the command below in your AWS Cloud9 terminal to make sure you are in the ~/environment directory of your AWS Cloud9 instance.

  ```
  cd ~/environment
  ```

- In your AWS Cloud9 environment, download the exercise code by typing the command below in the terminal.

  ```
  wget https://us-west-2-tcdev.s3.amazonaws.com/courses/AWS-100-ADG/v1.1.0/exercises/ex-sns-sqs.zip
  ```

- Unzip the exercise code .zip file by typing the command below in your AWS Cloud9 terminal.

  ```
  unzip ex-sns-sqs.zip
  ```

  The contents of the .zip file should be extracted to a folder with similar name. You can view the folder on the left tree view.

- To run the code, point the Run Configuration to the correct exercise folder. In the **Python3RunConfiguration** pane at the bottom, type the text shown below in the **Command** text box and then click **Run**.

  ```
  exercise-sns-sqs/FlaskApp/application.py
  ```

You should see a message like the one below:

```
Running on "http://0.0.0.0:8080/"
```

7. Test the email subscription to the Amazon SNS topic.

- To test the application, click **Preview -> Preview Running Application** on the top menu bar of the Cloud9 environment.
- Pop out the application in a new window by clicking the **Pop Out** button.
- Log in to the application and upload a photo.

  This upload publishes an Amazon S3 event to the Amazon SNS topic, which sends out an email to the subscribed email address. You should receive an email with JSON output containing information about the bucket and the photo you just uploaded.

8. Create an Amazon SQS queue and update the queue permissions.

In this section, you will create an Amazon SQS queue that stores the incoming request from the Amazon SNS topic as soon as a photo is uploaded to your bucket. Let's say you have an on-premises application doing additional processing on the uploaded photo. In a distributed system, your on-premises application will keep polling the Amazon SQS queue and process as soon as the request comes in. To create an Amazon SQS queue, follow the steps below.

- In the AWS Console, open the Amazon SQS dashboard and click **Get Started Now**.
- Make sure you are still in the **Oregon (us-west-2)** region.

- For **Queue Name**, type **uploads-queue**
- Scroll down and click **Quick-Create Queue**.
- In the **Details** tab at the bottom, make a note of the queue URL and ARN for later use.
- To update the queue policy to allow Amazon SNS publish an event to the queue, click the **Permissions** tab at the bottom.
- Click **Edit Policy Document (Advanced)**.
- In the policy editor, replace the existing policy document with the policy shown below. Make sure to replace **YOUR_QUEUE_ARN** and **YOUR_TOPIC_ARN** with the ARN values of the Amazon SQS queue and the Amazon SNS topic respectively.

```
  {
        "Version": "2012-10-17",
        "Id": "policy1",
        "Statement":
        [
                {
                        "Sid": "sid1",
                        "Effect": "Allow",
                        "Principal": "*",
                        "Action": "SQS:SendMessage",
                        "Resource": "YOUR_QUEUE_ARN",
                        "Condition":
                        {
                                "ArnEquals":
                                {

"aws:SourceArn": "YOUR_TOPIC_ARN"
                                }
                        }
                }
        ]
  }
```

- Click **Review Policy**.
- Click **Save Changes**.

9. Subscribe the Amazon SQS queue to the Amazon SNS topic.

- In the AWS Console, open the Amazon SNS dashboard.
- In the left navigation menu, click **Topics**.
- To select the topic, click the checkbox against **uploads-topic** .

- Click **Actions -> Subscribe to topic**.
- For **Protocol**, select **Amazon SQS**.
- For **Endpoint**, paste the Amazon SQS queue ARN you noted earlier.
- Click **Create subscription**.

  You should now receive a message in the queue whenever a photo is uploaded to the bucket.

10. Update the AWS Lambda function to subscribe to the Amazon SNS topic.

- In the AWS Console, open the AWS Lambda dashboard.
- In the left menu, click **Functions** and select the AWS Lambda function you created in the previous exercise.
- On the left **Add triggers** list, scroll down and select **SNS**.
- Scroll down to the **Configure triggers** section and notice that **uploads-topic** has been selected by default.
- Click **Add** at the bottom to save the trigger.
- Scroll up and click **Save** at the top-right corner to save the changes to the AWS Lambda function.

  Your AWS Lambda function will now process the photo as soon an Amazon SNS topic triggers the function.

11. Explore the AWS Lambda function code and package the function code along with the dependent libraries.

- In your AWS Cloud9 environment, open the **/exercise-sns-sqs/LambdaImageLabels/lambda_function.py** file.
- Explore the updated function code. Notice that the function now

processes the Amazon SNS event received when a photo is uploaded to the Amazon S3 bucket. Then it talks to Amazon Rekognition to process the photo labels, and finally updates the labels in the database.

- To update the new function code in your AWS Lambda function, you will need to package the code files and the MySQL connector libraries. You can use the .zip file, lambda.zip, created in the previous exercise. That file already contains the MySQL connector libraries. But you will need to update the **lambda_function.py** and the **config.py files** in the lambda.zip file. To update the lambda.zip file, type the commands below.

  ```
  cd exercise-sns-sqs/LambdaImageLabels
  zip ~/environment/lambda.zip *.py
  ```

  Note: If you were unsuccessful in creating the lambda.zip file with the MySQL connector libraries in the previous exercise, you may want to refer to the steps in the previous exercise to create the lambda.zip all over again.

- Change your working directory to the ~/environment folder by typing the command below.

  ```
  cd ~/environment
  ```

- Using the AWS CLI command below, update the AWS Lambda function you created in the previous exercise with the new function code.

  ```
  aws lambda update-function-code --function-name
  labels-lambda --zip-file fileb://lambda.zip
  ```

Upon executing the AWS CLI command, you should see a JSON output with the information of the AWS Lambda function. This means that the function has been successfully updated with the Python code and libraries.

12. Test the application.

- To run the code, you will need to point the Run Configuration to the correct exercise folder. In the **Python3RunConfiguration** pane at the bottom, type the text shown below in the **Command** text box and click **Run**.

  ```
  exercise-sns-sqs/FlaskApp/application.py
  ```

  You should see a message like the one below:

  ```
  Running on http://0.0.0.0:8080/
  ```

- To test the application, click **Preview -> Preview Running Application** on the top menu bar of the Cloud9 environment.
- Pop out the application in a new window by clicking the **Pop Out** button.
- Log in to the application and upload a photo.
- As soon as the photo is uploaded, you should see a label below the photo that reads, **Processing labels asynchronously**. The photo is now being processed by the AWS Lambda function triggered by an Amazon SNS event. You should also receive an email notification when the photo is uploaded.
- Click **my photos** on the top right corner. You should now see the labels populated in the table against the photo you just uploaded.

13. Test the Amazon SQS queue.

- In your AWS Cloud9 environment, open the **/exercise-sns-sqs/SqsLongPoll/sqs_long_poll.py** file.
- Explore the code. Notice that a Boto 3 client of the Amazon SQS SDK is created and the client long-polls the queue for an incoming message. The client then processes the message and extracts information such as the message body, the Amazon S3 object key, and the Amazon S3 object size.
- To test the code, type the command below in your AWS Cloud9 instance terminal. Make sure that you replace **YOUR_QUEUE_URL** in the command below with the Amazon SQS queue URL you noted earlier.

  ```
  cd ~/environment
  python3 exercise-sns-sqs/SqsLongPoll/sqs_long_poll.py
  YOUR_QUEUE_URL
  ```

  You should see an output like the example below. The code prints out the bucket name and the object key of the photo you uploaded.

  ```
  Polling the photos queue. Ctrl-C to exit.
  We have a new upload: bucket: myphotos key:
  photos/9bff7cde907a2b5b.png, size: 6455 bytes
  ```

- To see more similar messages, upload few more photos. Notice that the output in the instance terminal is updated with the object keys of the photos you uploaded.

  Your queue is now receiving the messages published by the Amazon SNS topic, thereby enabling your on-premises application to do the necessary processing.

14. Stop the NAT instance.

To keep your AWS account bill to a minimum, stop the NAT instance by following the steps below.

- In the AWS Console, open the Amazon EC2 dashboard.
- In the navigation pane, click **Instances**. In the list of instances, select **NAT server**.
- Click **Actions**, **Instance State**, and then **Stop**.

15. Stop the Amazon RDS database instance.

To keep your AWS account bill to a minimum, stop the Amazon RDS instance. Follow the steps below to stop the Amazon RDS database instance.

- In the AWS Console, open the Amazon RDS dashboard.
- In the left navigation pane, click **Instances**. From the list of instances, select **edx-photos-db**.
- At the top, click **Instance actions**, and then click **Stop**. You will see a prompt. Click **Yes, stop now**.

Optional Challenge

The script in `exercise-sns-sqs/SqsLongPoll/sqs_long_poll.py` is simulating what would be running on premises. We call `receive_message`, then we assume processing completes, and then we call `delete_message`. What if processing failed and exited before `delete_message`? For more information, see Visibility Timeout.

Can you answer the following questions?

- How long would it take for the failed message to reappear on the queue?

- What is the longest a message can remain on the queue?
- What if a bad message kept causing processing to fail, and kept reappearing on the queue? That could really hold up processing. Is there an automated way in Amazon SQS to move messages that can't be processed/consumed successfully? (Hint: Yes, there is. ☺)

Congratulations! You have successfully completed the course project.

If you want to delete the AWS resources created for this project, follow the steps below in order. **All steps are performed in the AWS Console.**

- Open the Amazon RDS dashboard and delete the **edx-photos-db** Amazon RDS database instance.
- Open the AWS Lambda dashboard and delete the **labels-lambda** function. Wait about 30 minutes. It will take time to fully delete the database and the AWS Lambda function. This will also delete the database security group.
- Open the VPC dashboard and delete the **labels-lambda-sg** security group. Then delete the **web-server-sg** security group.
- Open the AWS Cloud9 dashboard and delete the **BuildingOnAWS** AWS Cloud9 environment. It will take about 5 minutes for the AWS Cloud9 environment instance to be fully removed.
- Open the AWS CloudFormation dashboard and delete the **edx-vpc-stack** AWS CloudFormation stack. Deleting the AWS CloudFormation stack will also delete the NAT instance.
- Open the Amazon SNS dashboard and delete the **uploads-topic** Amazon SNS topic.
- Open the Amazon SQS dashboard and delete the **uploads-queue** Amazon SQS queue.

- Open the Amazon S3 dashboard and delete the Amazon S3 photos and deployment buckets.
- Open the Amazon Cognito dashboard and delete the **photos-pool** Amazon Cognito user pool.