Lab 6: Creating a Batch Processing Cluster

Overview	In this lab, you will use the AWS Management Console to build a basic batch processing cluster. You will: • Use Amazon SQS to create task queues for passing messages to your instances. • Launch an Auto Scaling group of instances. • Schedule work via your task queue. • Observe the output queue. The worker nodes in your cluster have a simple job: to convert some number of individual images into a single montage image. A worker node will download images from a list that you provide and will then stitch them into a composite montage using the ImageMagick tool. While this is not the most CPU-intensive job, it does require some cycles; the larger the size and number of images you provide for each job, the more work each node will have. For this lab, you will provide a newline-delimited list of image URLs. An Amazon EC2 worker node will download each image and produce output such as: 
Objectives	After completing this lab, you will be able to: • Use the AWS Management Console to create an Amazon SQS queue. • Create an Auto Scaling group with scaling policies based on an Amazon SQS queue. • Use the AWS Management Console to pass messages to, and read messages from, an Amazon SQS queue.

Pre-requisites	This lab requires: • Access to a notebook computer with Wi-Fi running Microsoft Windows, Mac OS X, or Linux (Ubuntu, SuSE, or Red Hat). • **Note** The qwikLABS lab environment is not accessible using an iPad or tablet device, but you can use these devices to access the student guide. • For Microsoft Windows users: Administrator access to the computer. • An Internet browser such as Chrome, Firefox, or Internet Explorer 9 (previous versions of Internet Explorer are not supported). • An SSH client, such as PuTTY.

Duration	This lab will require around **40 minutes** to complete.

Task 1: Check the existing VPC and subnets

Overview	Your batch processing nodes will communicate with Amazon SQS to receive processing instructions and will then store results in Amazon S3. Start by creating an IAM role that grants access to both Amazon SQS and Amazon S3. This role will be assigned to your Amazon EC2 instance.

Task 1-1: Create a role	
Overview	In this section of the lab, you will create an IAM role.

Step	Instruction
1.1.1	In the **AWS Management Console**, click **Identity & Access Management**.
1.1.2	In the **Dashboard** pane, click **Roles**.
1.1.3	Click **Create New Role**.
1.1.4	In the **Role Name** box, type **BatchProcessing**
1.1.5	Click **Next Step**.
1.1.6	On the **Select Role Type** page, ensure **AWS Service Roles** is selected.
1.1.7	In the **AWS Service Roles** menu, in the row for **Amazon EC2,** click **Select.**
1.1.8	On the **Attach Policy** page, select the **AmazonSQSFullAccess** policy from the list.
1.1.9	Click **Next Step**.
1.1.10	Click **Create Role**.
1.1.11	Click the name of the newly created **BatchProcessing** role to display the **Summary** view.
1.1.12	To add an additional role policy, click **Attach Policy**.
1.1.13	On the **Attach Policy** page, select the **AmazonS3FullAccess** policy from the list.
1.1.14	Click **Attach Policy**.

Task 2: Creating Two Amazon SQS Task Queues

Overview	In this section, you will use the AWS Management Console to create two Amazon Simple Queue Service (SQS) queues to hold input and output tasks. You will then send a message to your input queue. You will eventually dispatch work via the input queue and view the results provided by your worker nodes in the output queue.

Task 2-1: Create queues	
Overview	In this section of the lab you will create two **Amazon SQS** queues.

Step	Instruction
2.1.1	On the **Services** menu, click **SQS**.
2.1.2	Click **Get Started Now** (or, if **Get Started Now** does not appear, click **Create New Queue**)..
2.1.3	In the **Create New Queue** dialog box, specify the following settings: • **Queue Name:** input • **Default Visibility Timeout:** 90 seconds Leave the remaining settings in their default values.
2.1.4	Click **Create Queue**.
2.1.5	Click **Create New Queue** and repeat steps **2.1.3** and **2.1.4** to create another queue named **output** with a **Default Visibility Timeout** value of **90** seconds.
2.1.6	Select your **input** queue. Ensure that the output queue is not selected.
2.1.7	In the **Queue Actions** drop-down list, click **Send a Message**.
2.1.8	Copy the list of image URLs from the lab's provided Command Reference File and paste it in the **Enter the text of a message you want to send** dialog box.
2.1.9	Click **Send Message**.
2.1.10	Click **Close**.

Task 3: Creating an Amazon S3 Bucket

Overview	In this section, you will create an Amazon S3 bucket to hold the output from your worker nodes.

Task 3-1: Create an Amazon S3 bucket	
Overview	In this section of the lab, you will create an Amazon S3 bucket to hold your processed image files.

Step	Instruction
3.1.1	On the **Services** menu, click **S3**.
3.1.2	Click **Create Bucket**.
3.1.3	In the **Bucket Name** box, type **image-bucket-**<*number*> **Note** Your bucket name must be unique; so, change the number to generate a unique bucket name, such as *image-bucket-1029*.
3.1.4	To accept the default region and create your S3 bucket, click **Create**.

Task 4: Launching Worker Nodes

Overview	In this part of the lab, you will create an Auto Scaling group of worker nodes to process your work. After you successfully test the initial node, you will add Scaling Policies to automatically expand the size of the Auto Scaling group.

Task 4-1: Create a launch configuration	
Overview	In this section of the lab, you will create an Auto Scaling group that responds to a need to scale out by launching new instances.

Step	Instruction
4.1.1	On the **Services** menu, click **EC2**.
4.1.2	In the **EC2 Dashboard** pane, click **Security Groups**.
4.1.3	Click **Create Security Group**.
4.1.4	In the **Create Security Group** dialog, specify the following settings: • **Security group name:** BatchProcessing • **Description:** Batch Processing Security Group • **VPC:** LabVPC
4.1.5	Click **Add Rule** and specify the following settings: • **Type: SSH** • **Source: Anywhere**
4.1.6	Click **Create**.
4.1.7	In the **EC2 Dashboard** pane, click **Launch Configurations**.
4.1.8	Click **Create Auto Scaling group**.
4.1.9	Click **Create launch configuration**.
4.1.10	On the **Create Launch Configuration** page, in the row for **Amazon Linux AMI… (HVM) ,** click **Select.**
4.1.11	To accept the default instance type, click **Next: Configure details**.
4.1.12	On the **Create Launch Configuration** page, specify the following settings: • **Name: Workers** • **IAM role: BatchProcessing**
4.1.13	Click **Advanced Details** to expand it.

4.1.14	Next, you will supply a user data field to the Amazon EC2 Launch Configuration so that the instances it creates are configured to automatically run the image conversion process. In the **User data** box, verify that the **As text** option is selected. Using the command reference file that you used earlier, copy the user data text for this procedure into the **User data** box.
4.1.15	For **IP Address Type**, select the **Assign a public IP address to every instance** option.
4.1.16	Click **Next: Add Storage**.
4.1.17	Click **Next: Configure Security Group** to accept the default storage setting.
4.1.18	For **Assign a security group**, select the **Select an existing security group** option. From the list of security groups, select **Batch Processing Security Group**.
4.1.19	Click **Review**.
4.1.20	Click **Create launch configuration**.
4.1.21	To accept the key pair generated by qwikLABS, select the acknowledgement check box and then click **Create launch configuration**.

Task 4-2: Create an Auto Scaling group	
Overview	In this section of the lab, you will create an Auto Scaling group that responds to a need to scale out by launching new instances.

Step	Instruction
4.2.1	On the **Configure Auto Scaling group details** page, specify the following settings: • **Group name:** WorkerGroup • **Network:** LabVPC • Click in the **Subnet** box, and select **PublicSubnet1.**
4.2.2	Click **Next: Configure scaling policies**.
4.2.3	Select the **Use scaling policies to adjust the capacity of this group** option and change the policy to "Scale between **1** and **4** instances."
4.2.4	In the **Increase Group Size** section, do the following: • Click **Create a simple scaling policy.** • Change **Take the action** to **Add 1 instances.** • Change the **And then wait** value to **"90 seconds before allowing another scaling activity."** **Note** You've specified the scaling policies but the alarm you'll need to create can't be created here. You'll create the alarms in a later task, using **CloudWatch.**
4.2.5	In the **Decrease Group Size** section, click the **X** one the right side of the box.
4.2.6	Click **Review.**
4.2.7	Click **Create Auto Scaling group** and then click **Close**. Your Auto Scaling group has been configured to run only a single instance now, and no alarms have been attached to its scaling policies. After you verify that the worker node is functioning correctly, you can create alarms to automatically adjust the number of worker nodes.
4.2.8	In the navigation pane, click **Instances**. A new instance will start now. Wait until **Instance State** changes to *running* and **Status Checks** shows *2/2 checks passed.*

Task 5: Dispatching Work and Viewing Results

Overview	In this section, you will use the SQS Dashboard to view the output of your last message, access your worker instance using SSH and use a message producer script to add more messages to your Amazon SQS input queue in order to use CloudWatch SQS metrics in the next section.

Task 5-1: Check your queue for messages	
Overview	In this section of the lab, you will verify that the message you sent to the input queue earlier has been processed and moved to the output queue.

Step	Instruction
5.1.1	On the **Services** menu, click **SQS**. Confirm that there is **1 Message Available** in your **output** queue. If the message is still in your input queue, click **Refresh** periodically. It may take a few minutes after your worker instances were created for the message to move to the **output** queue from the **input** queue. **Note** If there is no message in your **output** queue: • Ensure that your queues are named **input** and **output** (in lowercase). • Ensure that your **BatchProcessing** role has granted full permissions for Amazon SQS and Amazon S3. • Ensure that your worker node is using the **BatchProcessing** IAM role (defined in the Launch Configuration). • Ensure that your worker node is running. Ask your instructor for assistance in successfully running your worker node.
5.1.2	Select the **output** queue. Ensure that the **input** queue is not selected.
5.1.3	In the **Queue Actions** drop-down list, click **View/Delete Messages**.
5.1.4	Click **Start Polling for Messages**.
5.1.5	Find your message and click **More Details** to view the message body. The message contains the output. To view the image, select and copy the URL portion of the message. Then paste the link into a new browser window. You should see a composite image of the five images that were linked individually in the message you sent to the **input** queue in **Step 2.1.9**.
5.1.6	Click **Close** to close the **Message Details** window.
5.1.7	Click **Close** to close the **View/Delete Messages in output** window.

Task 5-2: Download your key pair	
Overview	In this section of the lab, you will download your key pair file which was generated by qwikLABS.

Step	Instruction
5.2.1	Return to the qwikLABS web page and click the **Download PEM/PPK** drop-down list. • Windows users: click **Download PPK**. • Mac/Linux users: click **Download PEM**.
5.2.2	Save the file to the directory of your choice.

Task 5-3: Connect to your worker instance and run the message production script (Windows)	
Overview	**Note** This section is for **Windows** users only. If you are running MacOS or Linux, skip to **Task 5-4**. In this section of the lab, you will connect to your worker instance via SSH and run the message production script.

Step	Instruction
5.3.1	Launch **PuTTY**.
5.3.2	On the **Services** menu, click **EC2**.
5.3.3	In the **EC2 Dashboard** pane, click **Instances**.
5.3.4	Select your worker instance. **Note** Your worker instance will be unnamed.
5.3.5	Copy the **Public IP** of your worker instance from the EC2 Dashboard and paste it into the **Host Name** box in your **PuTTY** client.
5.3.6	In the **Connection** list, expand **SSH**, and then click **Auth**.
5.3.7	In the **Private key file for authentication** box, browse to the **PPK** file that you downloaded in **Task 5.2**.
5.3.8	Click **Open** to initiate the connection, in the PuTTY security message, click **Yes** and then log in as **ec2-user**.
5.3.9	Run the following command: `sudo python MessageProducer.py` **Note** You should receive a message that confirms that the message was sent to your input queue. If you do not receive this confirmation, notify your instructor. If attempts to run the script continue to fail, you can emulate this script's behavior by using steps **2.1.6–2.1.9** to send at least 35 messages from your input queue. The purpose of this script is to generate a long backup of messages in your input queue, so that the Amazon CloudWatch metric associated with visible queue messages will be available in the next task.
5.3.10	Proceed to **Task 6: Monitoring the Cluster**.

Task 5-4: Connect to your worker instance and run the message production script (MacOS and Linux)	
Overview	**Note** This section is for **MacOS and Linux** users only. If you are running Windows and have not yet connected to your instance, go back to **Task 5-3**. If you have already connected to your instance, skip ahead to **Task 6**. In this section of the lab, you will connect to your worker instance via SSH and run the message production script using a OS X or Linux machine:

Step	Instruction
5.4.1	On the **Services** menu, click **EC2**.
5.4.2	In the navigation pane, click **Instances**.
5.4.3	Select your worker instance. **Note** Your worker instance will be unnamed.
5.4.4	Copy the **Public IP** of your worker instance from the EC2 Dashboard.

5.4.5	Run the following command to connect to the Amazon EC2 instance:
	
	`chmod 400 <path and name of pem>`
	
	`ssh -i <path and name pem> ec2-user@<Public IP>`
	
	Substitute the Public IP value for *<Public IP>*.

5.4.6	Run the following command: `sudo python MessageProducer.py` **Note** You should receive a message that confirms that the message was sent to your input queue. If you do not receive this confirmation, notify your instructor. If attempts to run the script continue to fail, you can emulate this script's behavior by using steps 2.1.6–2.1.9 to send at least 35 messages from your input queue. The purpose of this script is to generate a long backup of messages in your input queue, so that the Amazon CloudWatch metric associated with visible queue messages will be available in the next task.
5.4.7	Proceed to **Task 6: Monitoring the Cluster**.

Task 6: Monitoring the Cluster

Overview	You can now use CloudWatch to monitor your cluster. You will define a CloudWatch alarm for use with Auto Scaling policies.

Task 6-1: Create an alarm	
Overview	In this section of the lab, you will create an alarm that triggers your Auto Scaling group.

Step	Instruction
6.1.1	On the **Services** menu, click **CloudWatch**.
6.1.2	Click **Browse Metrics**.
6.1.3	To expand the drop-down list, click **Browse Metrics** again.
6.1.4	In the **Browse Metrics** drop-down list, click **SQS**. **Note** If the **SQS Metrics** header is not visible, return to your **input** queue and ensure that there are messages queued for processing. This will trigger metrics to be sent to CloudWatch after a few minutes. The **Visible** count may take a bit longer to appear. If the list of messages waiting in your input queue is lower than 25, run the message producer script again.
6.1.5	Select the check box next to the **ApproximateNumberOfMessagesVisible** metric for the **input** queue.
6.1.6	On the **Tools** menu in the lower-right corner, click **Create Alarm**.
6.1.7	On the **Alarm Threshold** page, specify the following settings: • **Name:** LongQueue • **Description:** Queue too long • Enter **10** so that the threshold is: **Whenever:** ApproximateNumberOfMessagesVisible **is:** >= **10** **for:** **1** consecutive period(s)
6.1.8	Under the **Actions** section, click **Delete** on the existing Notification Action.
6.1.9	Click **+ AutoScaling Action**, and specify the following settings: • **From the group:** WorkerGroup • **Take this action:** Increase Group Size – Add 1 instance

6.1.10	Click **Create Alarm**. **Note** These configuration settings will automatically **add 1 instance** to your Auto Scaling group called **WorkerGroup** whenever the **input** queue has **10 or more messages visible**.

Task 6-2: Test your Auto Scaling group	
Overview	In this section of the lab, you will test and verify that your Auto Scaling group adds and removes instances.

Step	Instruction
6.2.1	In the navigation pane, click **CloudWatch**. Your **LongQueue** alarm should appear within the **Alarm Summary** section.
6.2.2	Refresh the **Alarm Summary** section until it displays **"You have 1 alarm in ALARM state…"**
6.2.3	Switch back to the **EC2 Dashboard.** In the navigation pane, click **Instances**. Within a few minutes, you should be able to see the additional instances that have been launched by your Auto Scaling group. As a result, the messages in your queue will be processed more quickly.
6.2.4	In the navigation pane, click **Auto Scaling Groups** and select your **WorkerGroup**. On the **Activity History** tab, you can verify that Auto Scaling has launched new instances. If you regularly switch between CloudWatch and your Auto Scaling group's Activity History pane, you can see how your group responds to the number of messages CloudWatch sees waiting in your input queue.