Linux Academy

Live! Lab

Auto-Scaling Based Off SQS Message Queue Size

Contents

Lab Connection Information

- Labs may take up to five minutes to build
- Access to an AWS Console is provided on the Live! Lab page, along with your login credentials
- Ensure you are using the N. Virginia region
- Labs will automatically end once the alloted amount of time finishes

Related Courses

AWS CSA - Pro Course

Related Videos

SQS Message Priority

SQS Batch Processing Job Observer

Need Help?

Linux Academy Community

... and you can always send in a support ticket on our website to talk to an instructor!

Introduction

In this lab, we learn how to auto-scale your infrastructure through a combination of *CloudWatch*, *SQS* messages, and AWS's *Auto Scaling* feature. First, we create an SQS queue, and then, through CloudWatch, set up an alarm to trigger every time we hit 40 messages in our queue. When this alarm triggers, another instance is added.

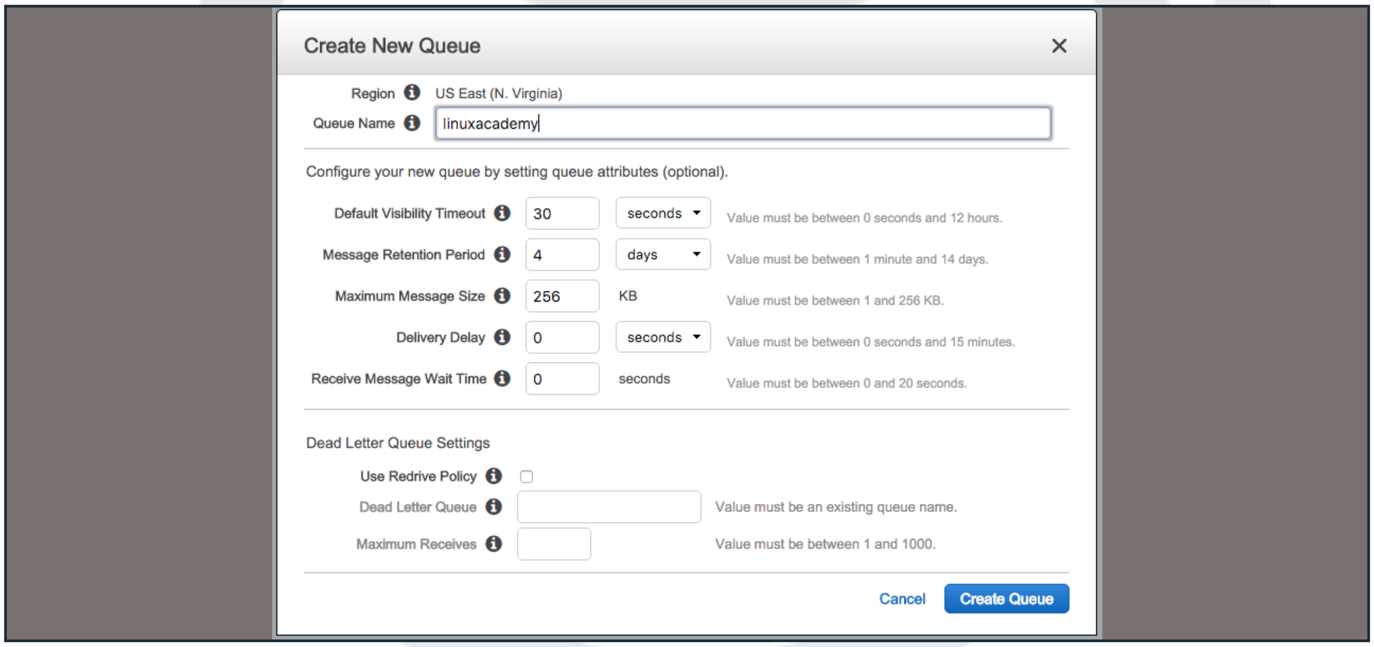
Log in to your **AWS Console** with the credentials provided on the **Live! Lab** page.

Within your **EC2 Dashboard**, you can see an instance called *Ops* — this is the instance we are using to send messages to our SQS queue.

However, we first need to create the queue.

Creating the SQS Queue

Navigate to the **SQS Dashboard** and press **Get Started Now**.


Create New Queue

Region: US East (N. Virginia)
Queue Name: linuxacademy

Configure your new queue by setting queue attributes (optional).

Default Visibility Timeout	30	seconds	Value must be between 0 seconds and 12 hours.
Message Retention Period	4	days	Value must be between 1 minute and 14 days.
Maximum Message Size	256	KB	Value must be between 1 and 256 KB.
Delivery Delay	0	seconds	Value must be between 0 seconds and 15 minutes.
Receive Message Wait Time	0	seconds	Value must be between 0 and 20 seconds.

Dead Letter Queue Settings

Use Redrive Policy	☐	
Dead Letter Queue		Value must be an existing queue name.
Maximum Receives		Value must be between 1 and 1000.

Enter a **Queue Name**. We choose *linuxacademy* for our queue. We left the *default options* for the rest of the queue settings.

For information regarding these attributes, review the ***SQS videos*** *for the* ***AWS Certified Developer - Associate*** *course.*

Press **Create Queue**.

Sending a Message

To send an initial message to our new SQS queue, we need to log in to our *Ops* instance. Open your terminal on your workstation. You can retrieve the *IP address* of your instance from the **EC2 Dashboard**. The username you are logging in with is *linuxacademy* with a password of *123456*.

`su` into the root user. The password is also *123456*.

Now configure the AWS Cli:

```
aws configure
```

You are asked for an **AWS Access Key ID**, press *enter* to select the default, *none*. Do the same for the **AWS Secret Access Key**. The **Default region name** should be set to *us-east-1*. The **Default output format** should also be left to *none*.

To send a message to our SQS queue, we need the URL to that queue. From the **SQS Dashboard**, select your *linuxacademy* queue, and *copy the URL provided*. The URL is located in the **Details** section.

Send a message to the queue from the command line, replacing `https://sqs.us-east-1.amazonaws.com/642132563443/linuxacademy` with *your queue's URL*:

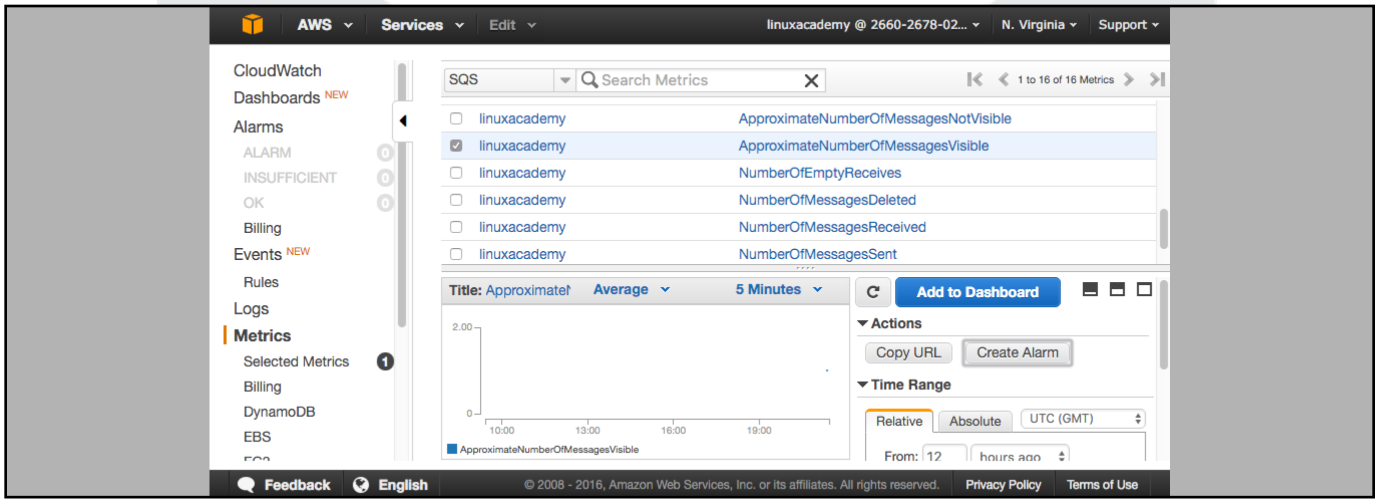
```
aws sqs send-message --queue-url https://sqs.us-east-1.amazonaws.com/642132563443/
linuxacademy --message-body linuxacademytest
```

The `message-body` can be any form of example text.

If you return to/refresh your queue, you should now see *1 Message[s] Available*.

Scaling with CloudWatch

Through the use of CloudWatch, we can configure our infrastructure to scale based on how many messages are in our queue. From the **CloudWatch Dashboard**, select **SQS** from the left menu.



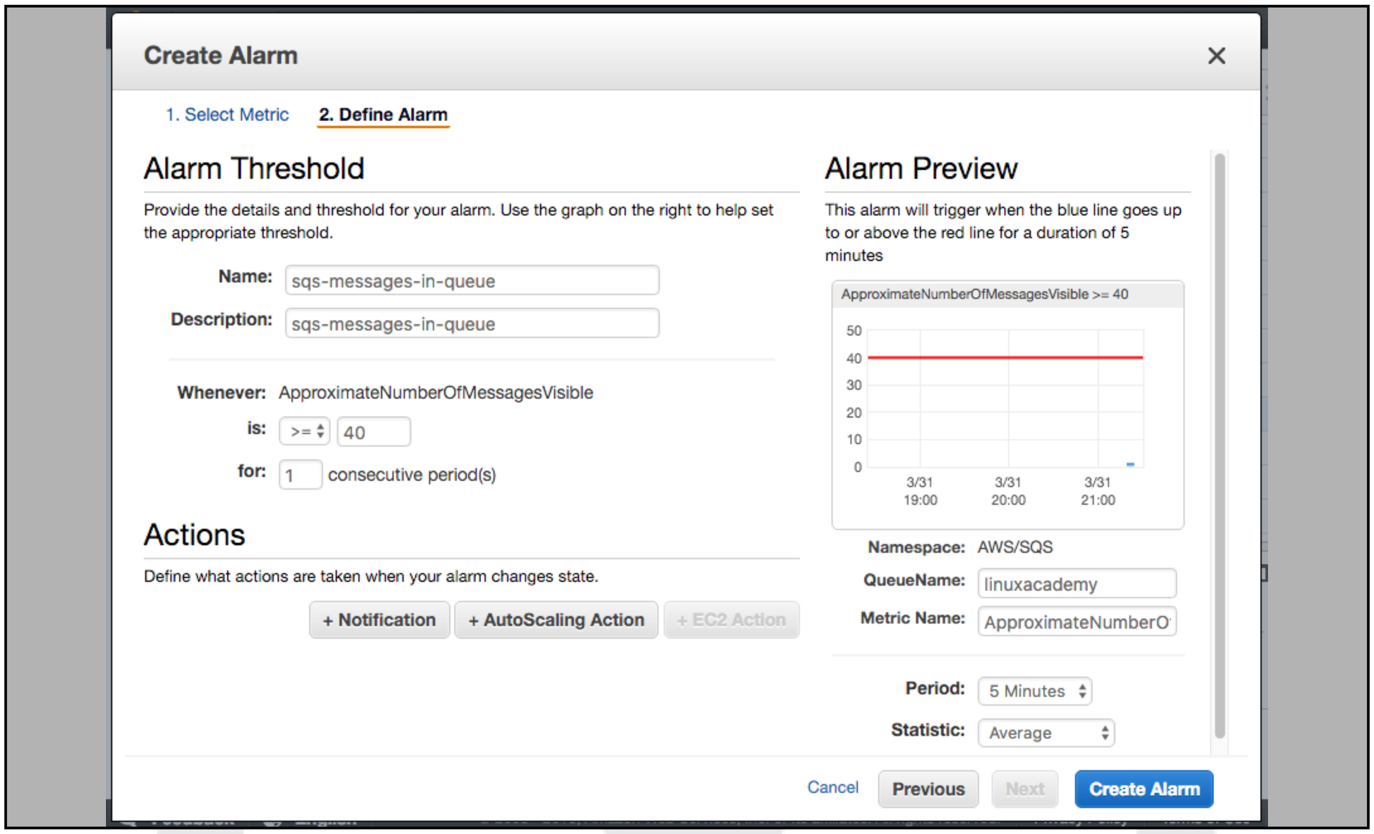
☐	linuxacademy	ApproximateNumberOfMessagesNotVisible
☑	linuxacademy	ApproximateNumberOfMessagesVisible
☐	linuxacademy	NumberOfEmptyReceives
☐	linuxacademy	NumberOfMessagesDeleted
☐	linuxacademy	NumberOfMessagesReceived
☐	linuxacademy	NumberOfMessagesSent

Title: Approximatel Average 5 Minutes

From: 12 hours ago

We are working with the *linuxacademy* queue (under **QueueName**) — if it is not yet available, refresh the page until it is.

We want to create an alarm based on the **ApproximateNumberOfMessagesVisible** metric. Select this, and press **Create Alarm**.



Create Alarm

1. Select Metric 2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: sqs-messages-in-queue
Description: sqs-messages-in-queue

Whenever: ApproximateNumberOfMessagesVisible
is: >= 40
for: 1 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

Namespace: AWS/SQS
QueueName: linuxacademy
Metric Name: ApproximateNumberO
Period: 5 Minutes
Statistic: Average

For **Name** and **Description** we put *sqs-messages-in-queue*.

We want CloudWatch to set an alarm when we hit a certain number of messages. This alarm deploys out new instances based on the auto-scaling policy we use.

Set the **Whenever: ApproximateNumberOfMessagesVisible** to *>= 40*, for a **period** of *1 Minute*. Delete the notification option. Select **Create Alarm**.

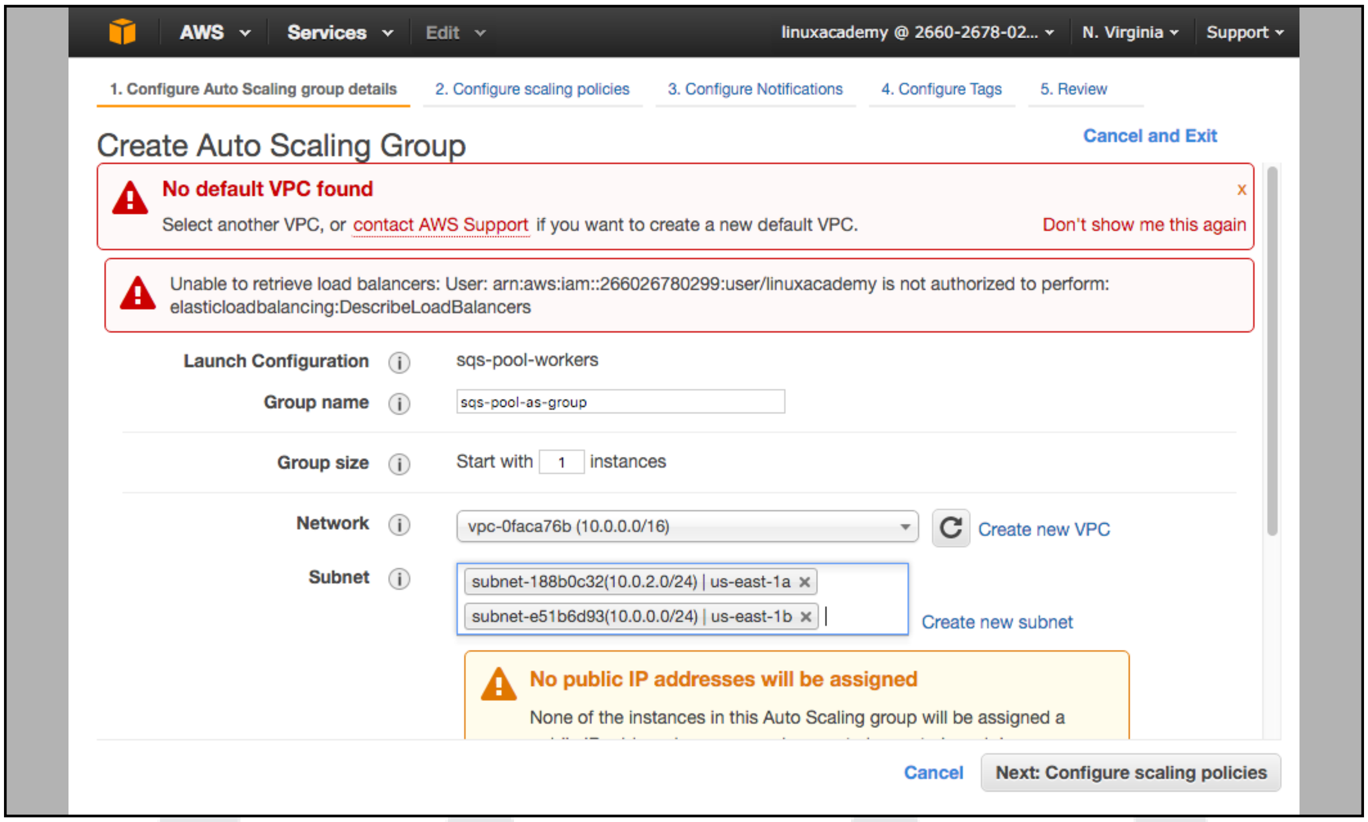
Configuring EC2 Launch Configurations

From the **EC2 Dashboard**, select **Launch Configurations**, **Create Auto Scaling group**. Press **Create Launch Configuration**.

Select the **Amazon Linux AMI** and the *General purpose t2.micro*. Select **Next: Configure details**.

We gave our configuration a **Name** of *sqs-pool-workers*, and selected **Next: Add Storage**. Leave all storage settings as-is, and press **Next: Configure security group**. This can also be left as-is. Press **Review, Create**

launch configuration. A pop-up asks you to select a key pair; from the inital drop-down, select *proceed without a key pair*. **Create launch configuration**.


AWS
Services
Edit
linuxacademy @ 2660-2678-02...
N. Virginia
Support
1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review
Cancel and Exit
Create new VPC
Create new subnet
Cancel
Next: Configure scaling policies

Create Auto Scaling Group

No default VPC found
Select another VPC, or contact AWS Support if you want to create a new default VPC. Don't show me this again

Unable to retrieve load balancers: User: arn:aws:iam::266026780299:user/linuxacademy is not authorized to perform: elasticloadbalancing:DescribeLoadBalancers

Launch Configuration: sqs-pool-workers
Group name: sqs-pool-as-group
Group size: Start with 1 instances
Network: vpc-0faca76b (10.0.0.0/16)
Subnet: subnet-188b0c32(10.0.2.0/24) | us-east-1a; subnet-e51b6d93(10.0.0.0/24) | us-east-1b

No public IP addresses will be assigned
None of the instances in this Auto Scaling group will be assigned a

We now need to create an Auto Scaling group. Our **Group** name is set to *sqs-pool-as-group*. Under **Network**, select the single available VPC, and then choose both subnets for the **Subnet** values. Press **Next: Configuring scaling policies**.

We want to *Use scaling policies to adjust the capacity of this group* to scale between *1* and *4 instances*.

For **Execute policy when**, select your *sqs-messages-in-queue* alarm, and set **Take the action** for *1 instance*. Our **Instances need** is set to *10 seconds*.

Because this is a lab environment, we are working only with scaling up. Remove the decrease policy below, or use your new skills to try to set it up yourself.

Press **Review, Create Auto Scaling group**.

Testing the Scale

To test our new policies, we need to quickly generate forty or more messages for our queue. Return to your teminal, where you should still be logged into the *Ops* instance as *root*.

Using your text editor of choice, create a script called *sqs.sh*, replacing the `queue-url` appropriatel:

```
#! /bin/bash
for i in $(seq 1 60)
  do
    aws sqs send-message --queue-url https://sqs.us-east-1.amazonaws.com/642132563443/
linuxacademy --message-body linuxacademytest
done
```

Make the script executable:

```
chmod +x sqs.sh
```

Finally, run the script:

```
./sqs.sh
```

If you return to your **SQS queue**, you can refresh to see your number of **Messages Available**. When the number reaches *61*, it is finished.

Navigate to your **EC2 Dashboard** to confirm more instances are spinning up.