EXP1: Introduction to AWS Identity and Access Management (IAM)

Lab Overview

AWS Identity and Access Management (IAM) is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

Task 1: Explore the Users and Groups

In this task, you will explore the Users and Groups that have already been created for you in IAM.

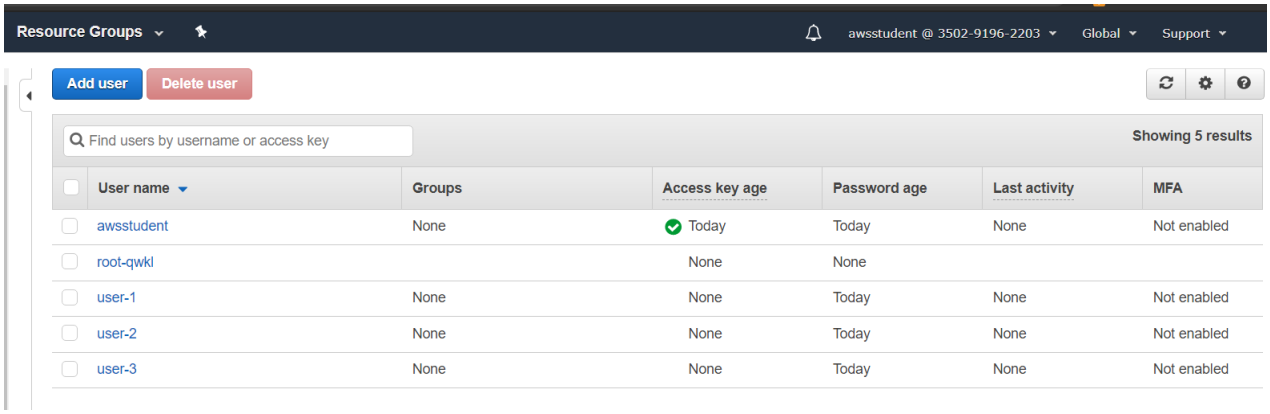
In the AWS Management Console, on the Services menu, click IAM.

In the navigation pane on the left, click Users.

The following IAM Users have been created for you:

- user-1
- user-2
- user-3

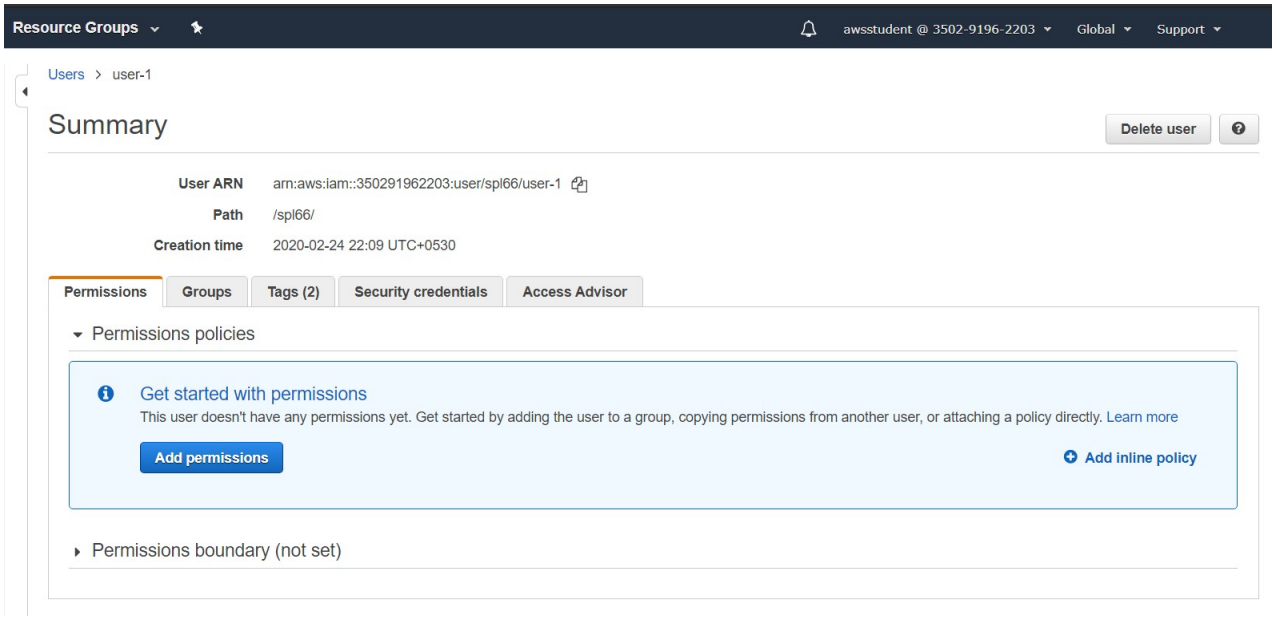
There is also an awsstudent user, which you can ignore for this lab.



Click user-1.

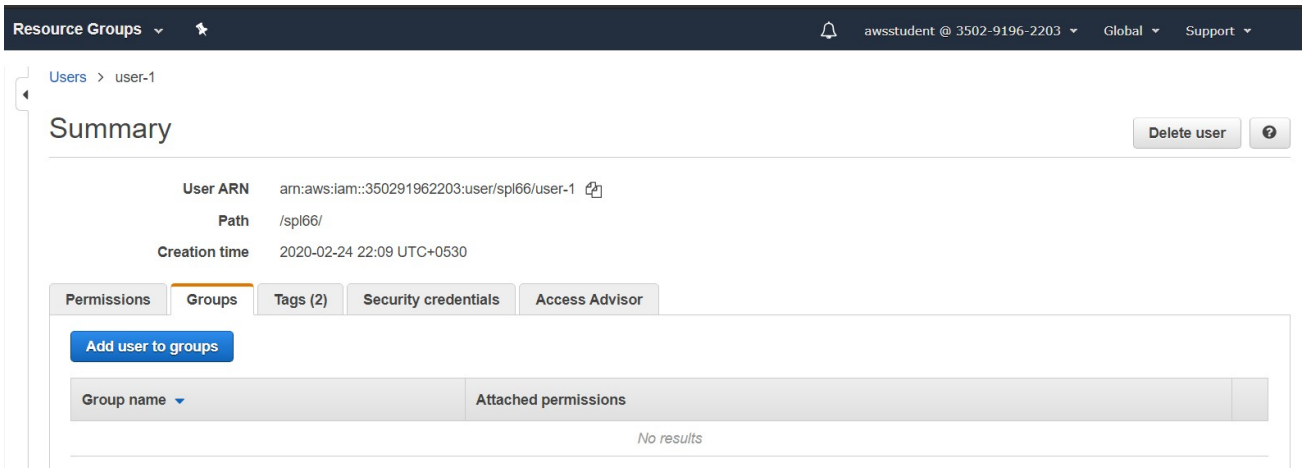
This will bring to a summary page for user-1. The Permissions tab will be displayed.

Notice that user-1 does not have any permissions.



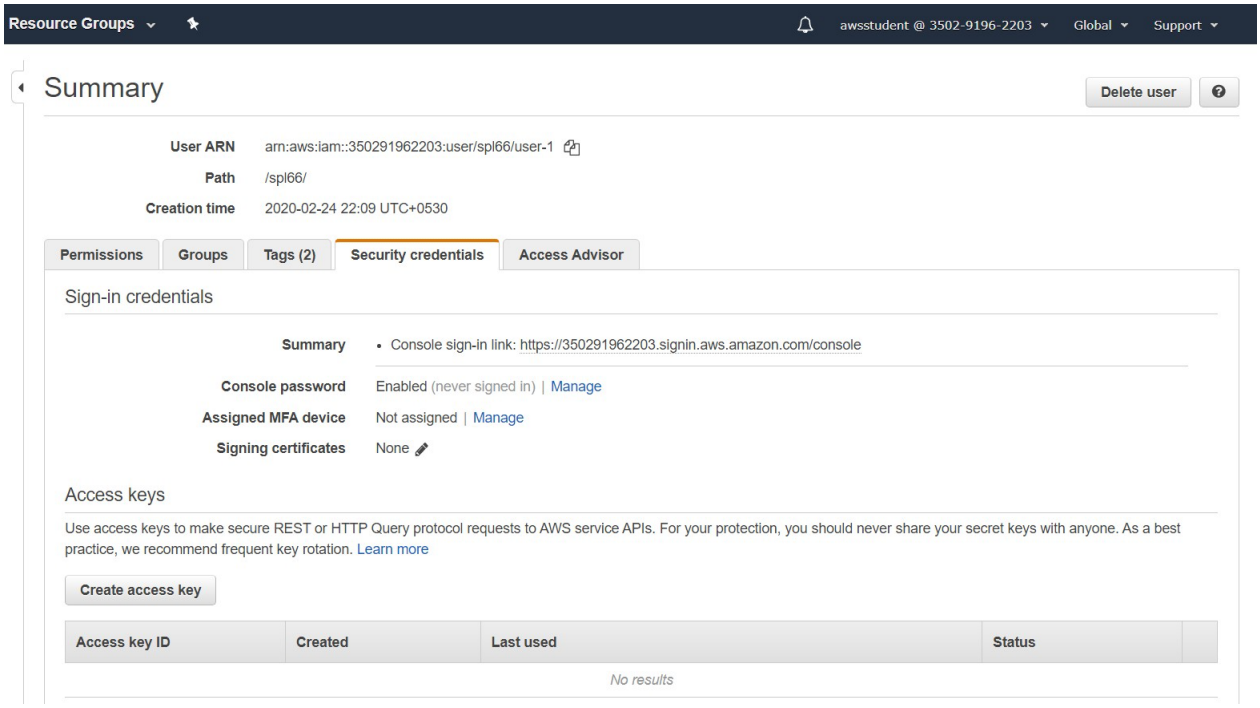
Click the Groups tab.

user-1 also is not a member of any groups.



Click the Security credentials tab.

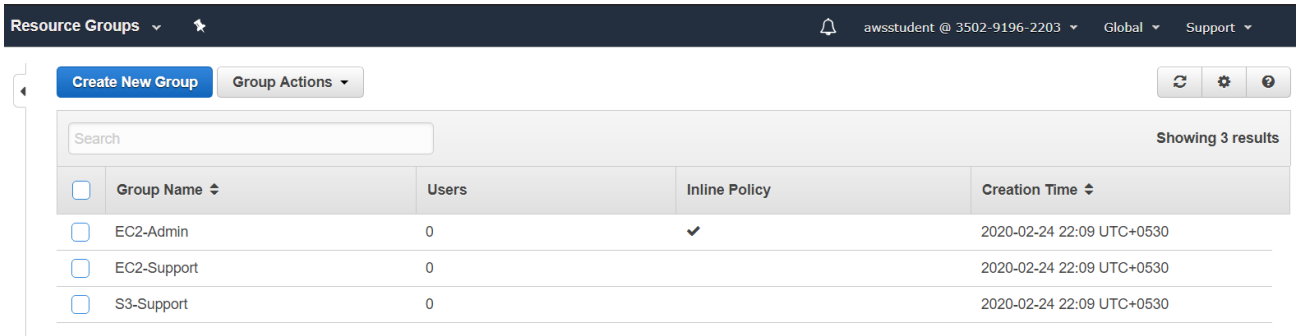
user-1 is assigned a Console password



In the navigation pane on the left, click Groups.

The following groups have already been created for you:

- EC2-Admin
- EC2-Support
- S3-Support



Click the EC2-Support group.

This will bring you to the summary page for the EC2-Support group.

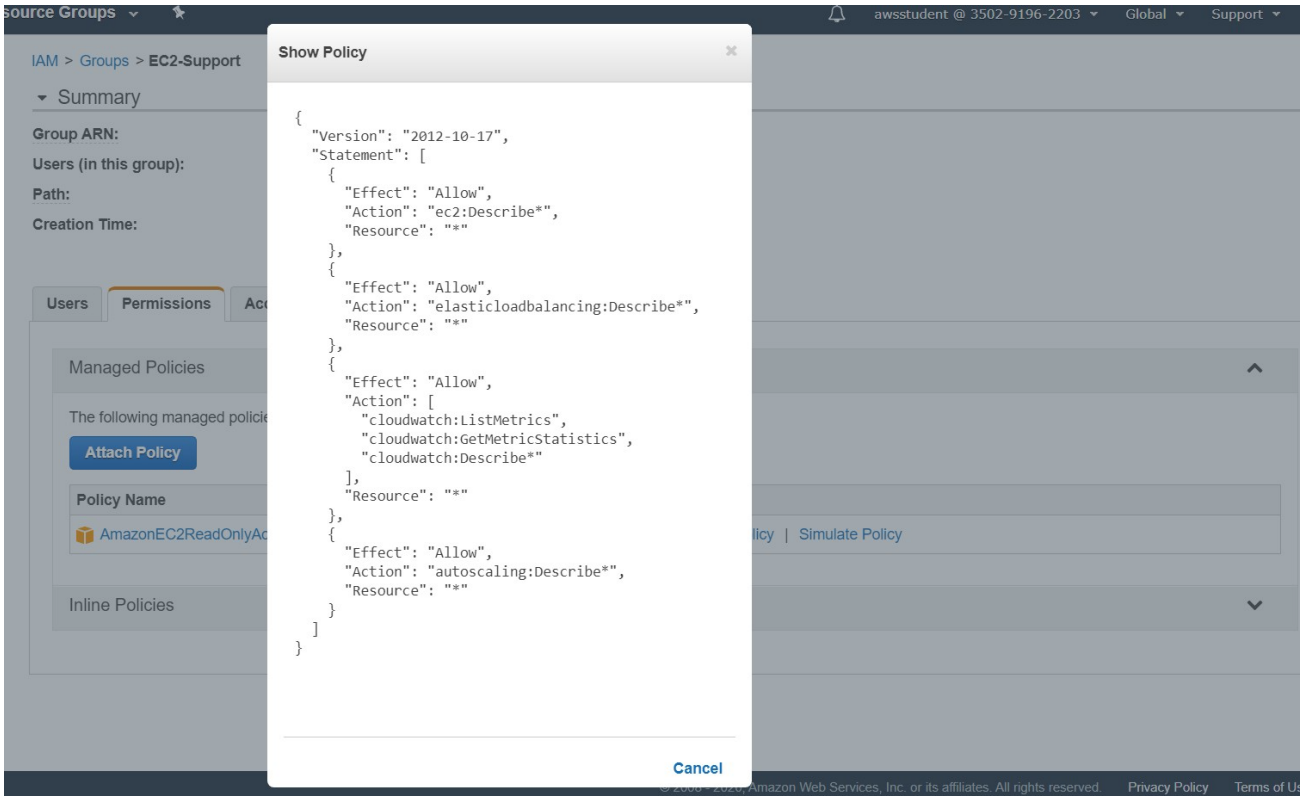
Click the Permissions tab.

This group has a Managed Policy associated with it, called AmazonEC2ReadOnlyAccess. Managed Policies are pre-built policies (built either by AWS or by your administrators) that

can be attached to IAM Users and Groups. When the policy is updated, the changes to the policy are immediately apply against all Users and Groups that are attached to the policy.

Under Actions, click the Show Policy link.

A policy defines what actions are allowed or denied for specific AWS resources. This policy is granting permission to List and Describe information about EC2, Elastic Load Balancing, CloudWatch and Auto Scaling. This ability to view resources, but not modify them, is ideal for assigning to a Support role.



The basic structure of the statements in an IAM Policy is:

Effect says whether to Allow or Deny the permissions.

Action specifies the API calls that can be made against an AWS Service (eg cloudwatch:ListMetrics).

Resource defines the scope of entities covered by the policy rule (eg a specific Amazon S3 bucket or Amazon EC2 instance, or * which means any resource).

Close the Show Policy window.

In the navigation pane on the left, click Groups.

Click the S3-Support group.

The S3-Support group has the AmazonS3ReadOnlyAccess policy attached.

Below the Actions menu, click the Show Policy link.

This policy has permissions to Get and List resources in Amazon S3.

Close the Show Policy window.

In the navigation pane on the left, click Groups.

Click the EC2-Admin group.

This Group is slightly different from the other two. Instead of a Managed Policy, it has an Inline Policy, which is a policy assigned to just one User or Group. Inline Policies are typically used to apply permissions for one-off situations.

Under Actions, click Edit Policy to view the policy.

This policy grants permission to view (Describe) information about Amazon EC2 and also the ability to Start and Stop instances.

At the bottom of the screen, click Cancel to close the policy.

Business Scenario

For the remainder of this lab, you will work with these Users and Groups to enable permissions supporting the following business scenario:

Your company is growing its use of Amazon Web Services, and is using many Amazon EC2 instances and a great deal of Amazon S3 storage. You wish to give access to new staff depending upon their job function:

User In Group Permissions

user-1 S3-Support Read-Only access to Amazon S3

user-2 EC2-Support Read-Only access to Amazon EC2

user-3 EC2-Admin View, Start and Stop Amazon EC2 instances

Task 2: Add Users to Groups

You have recently hired user-1 into a role where they will provide support for Amazon S3. You will add them to the S3-Support group so that they inherit the necessary permissions via the attached AmazonS3ReadOnlyAccess policy.

Add user-1 to the S3-Support Group

In the left navigation pane, click Groups.

Click the S3-Support group.

Click the Users tab.

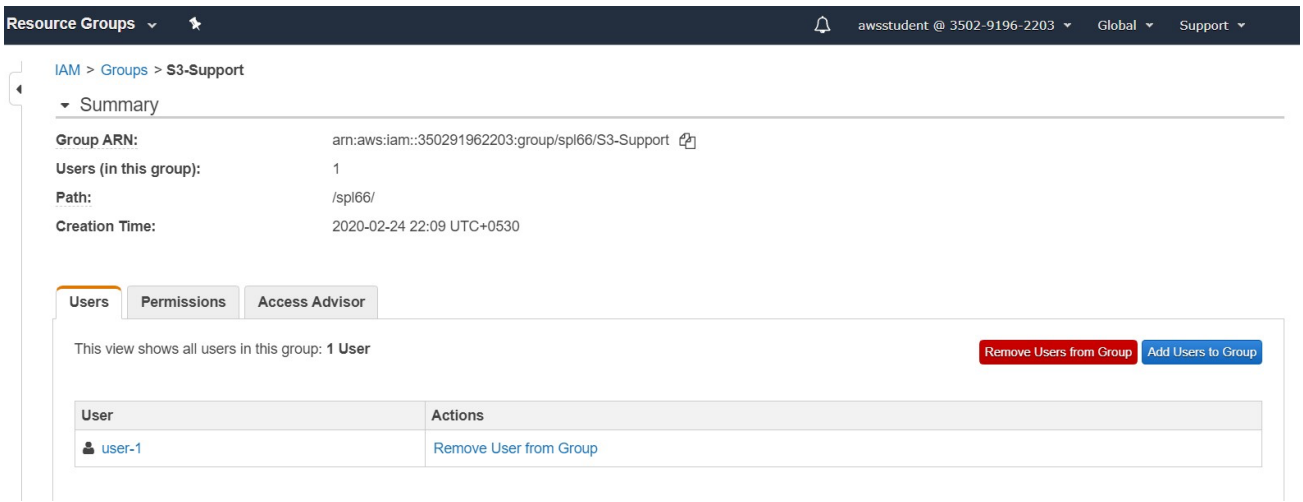
In the Users tab, click Add Users to Group.

In the Add Users to Group window, configure the following:

Select user-1.

At the bottom of the screen, click Add Users.

In the Users tab you will see that user-1 has been added to the group.

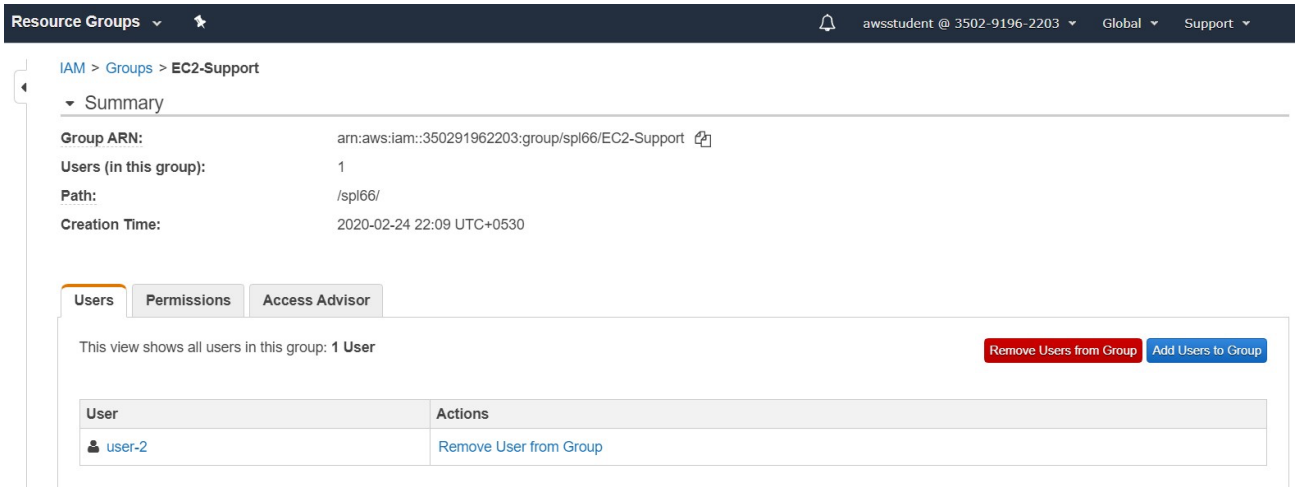


Add user-2 to the EC2-Support Group

You have hired user-2 into a role where they will provide support for Amazon EC2.

Using similar steps to the ones above, add user-2 to the EC2-Support group.

user-2 should now be part of the EC2-Support group.

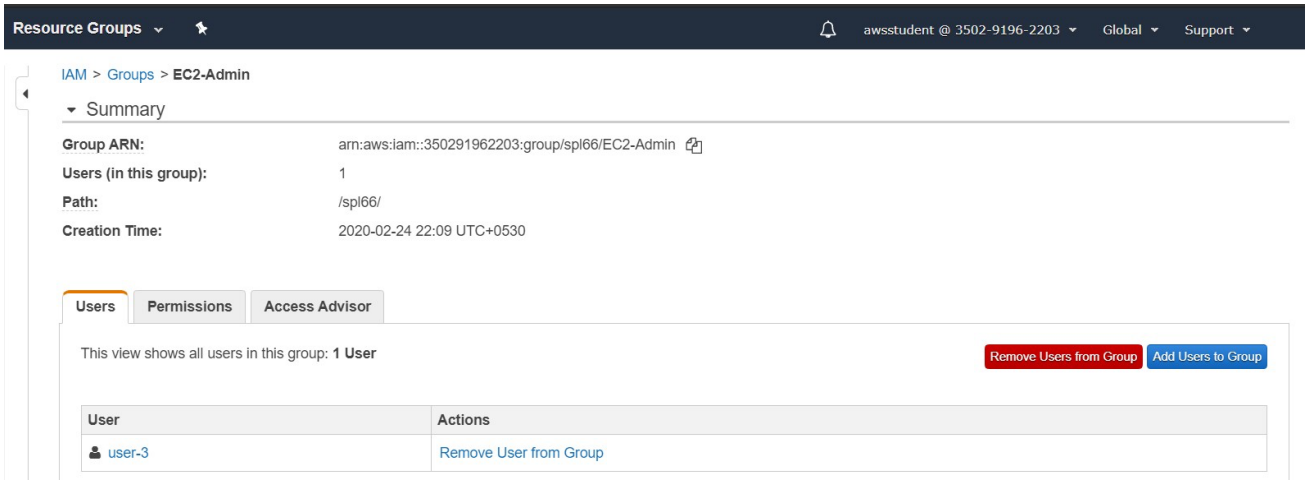


Add user-3 to the EC2-Admin Group

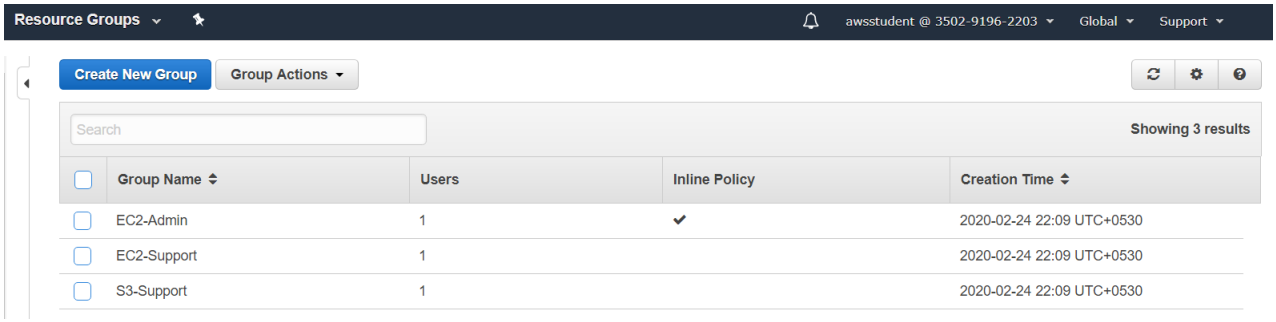
You have hired user-3 as your Amazon EC2 administrator, who manage your EC2 instances.

Using similar steps to the ones above, add user-3 to the EC2-Admin group.

user-3 should now be part of the EC2-Admin group.



In the navigation pane on the left, click Groups.



Each Group should have a 1 in the Users column for the number of Users in each Group.

Task 3: Sign-In and Test Users

In this task, you will test the permissions of each IAM User.

In the navigation pane on the left, click Dashboard.

An IAM users sign-in link is displayed It will look similar to:
https://123456789012.signin.aws.amazon.com/console

This link can be used to sign-in to the AWS Account you are currently using.

Copy the IAM users sign-in link to a text editor.

Open a private window.

Paste the IAM users sign-in link into your private window and press Enter.

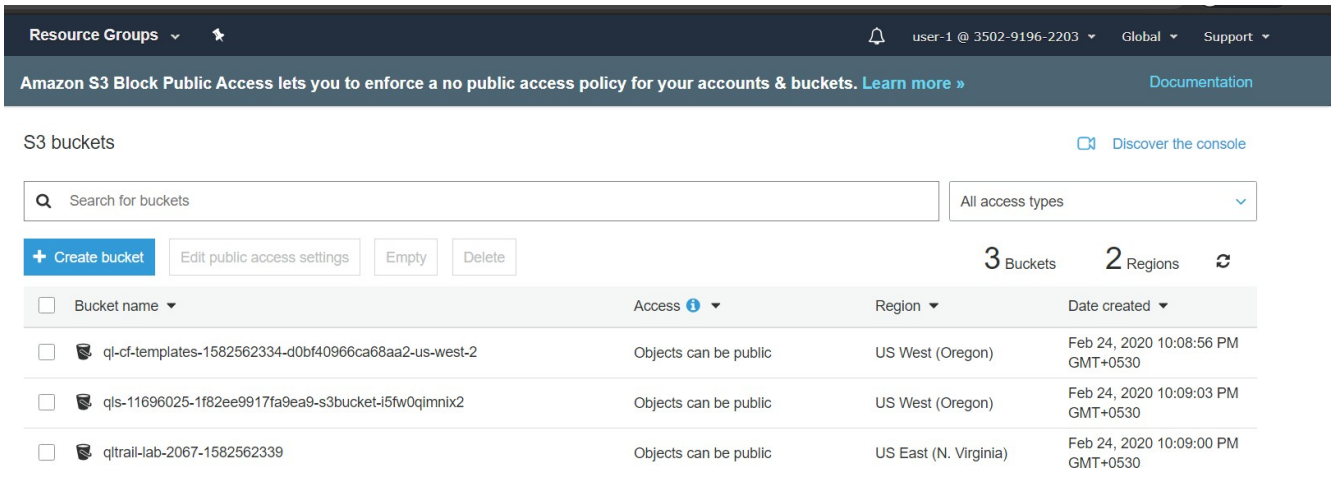
You will now sign-in as user-1, who has been hired as your Amazon S3 storage support staff.

Sign-in with:

IAM user name: user-1

Password: Paste the value of Password located to the left of these instructions.

In the Services menu, click S3.



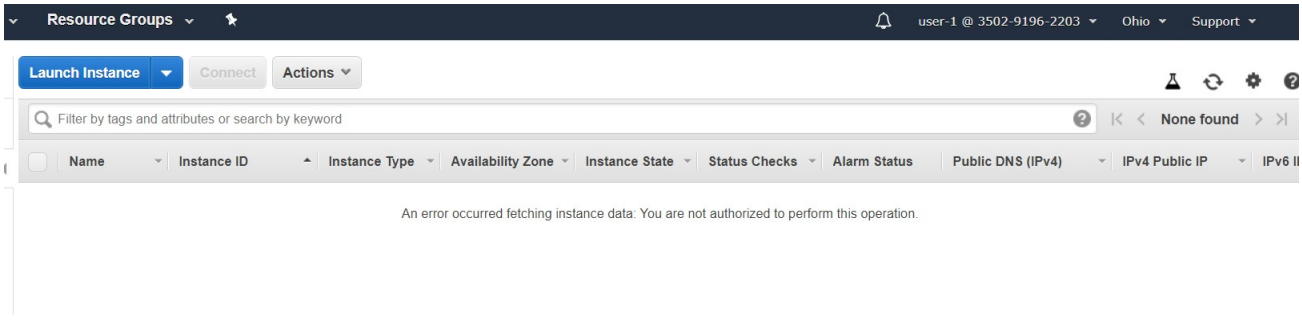
Click the name of one of your buckets and browse the contents.

Since your user is part of the S3-Support Group in IAM, they have permission to view a list of Amazon S3 buckets and their contents.

Now, test whether they have access to Amazon EC2.

In the Services menu, click EC2.

In the left navigation pane, click Instances.



You cannot see any instances! Instead, it says An error occurred fetching instance data: You are not authorized to perform this operation.. This is because your user has not been assigned any permissions to use Amazon EC2.

You will now sign-in as user-2, who has been hired as your Amazon EC2 support person.

Sign user-1 out of the AWS Management Console by configuring the following:

At the top of the screen, click user-1

Click Sign Out

Paste the IAM users sign-in link into your private window and press Enter.

This links should be in your text editor.

Sign-in with:

IAM user name: user-2

Password: Paste the value of Password located to the left of these instructions.

In the Services menu, click EC2.

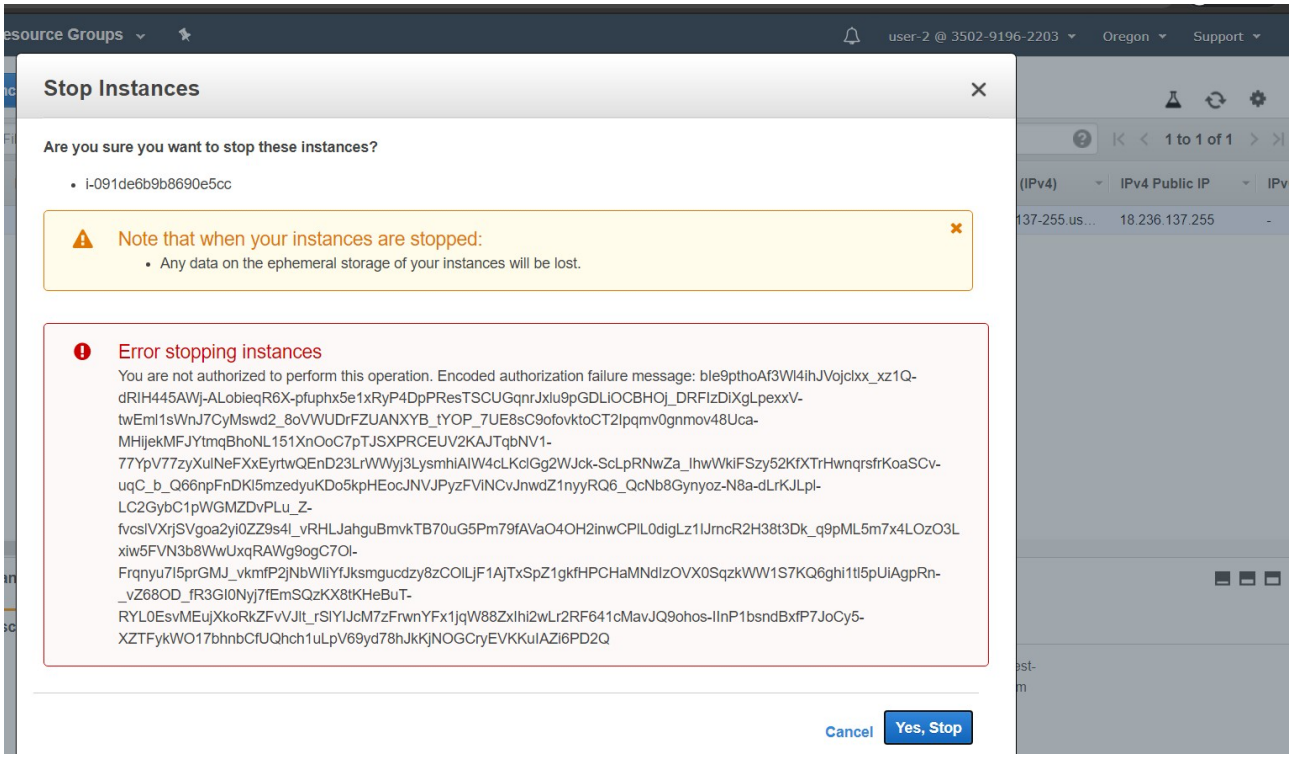
In the navigation pane on the left, click Instances.

You are now able to see an Amazon EC2 instance because you have Read Only permissions. However, you will not be able to make any changes to Amazon EC2 resources.

Your EC2 instance should be selected . If it is not selected, select it.

In the Actions menu, click Instance State > Stop.

In the Stop Instances window, click Yes, Stop.



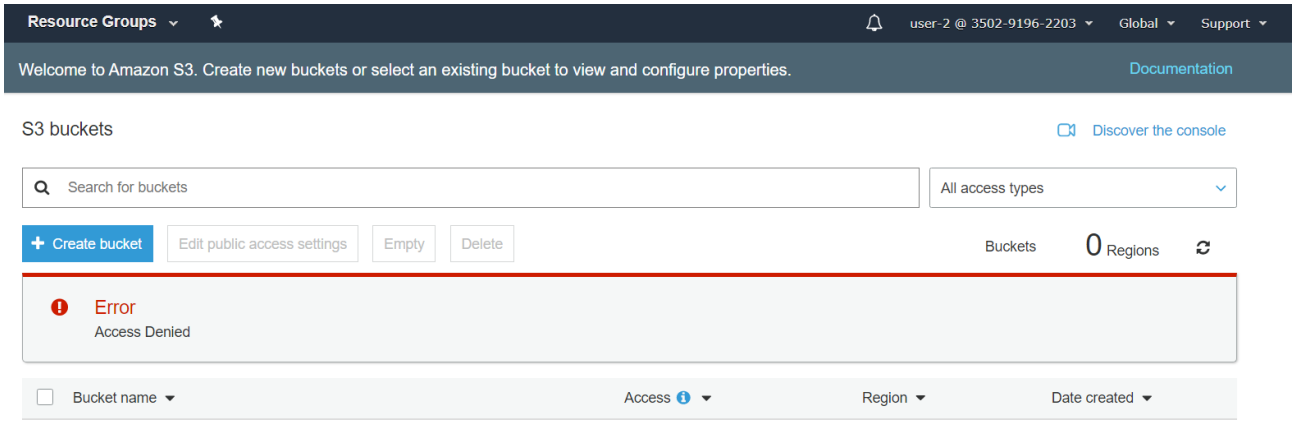
You will receive an error stating You are not authorized to perform this operation. This demonstrates that the policy only allows you to information, without making changes.

At the Stop Instances window, click Cancel.

Next, check if user-2 can access Amazon S3.

In the Services, click S3.

You will receive an Error Access Denied because user-2 does not permission to use Amazon S3.



You will now sign-in as user-3, who has been hired as your Amazon EC2 administrator.

Sign user-2 out of the AWS Management Console by configuring the following:

Paste the IAM users sign-in link into your private window and press Enter.

Paste the sign-in link into your web browser address bar again. If it is not in your clipboard, retrieve it from the text editor where you stored it earlier.

Sign-in with:

IAM user name: user-3

Password: Paste the value of Password located to the left of these instructions.

In the Services menu, click EC2.

In the navigation pane on the left, click Instances.

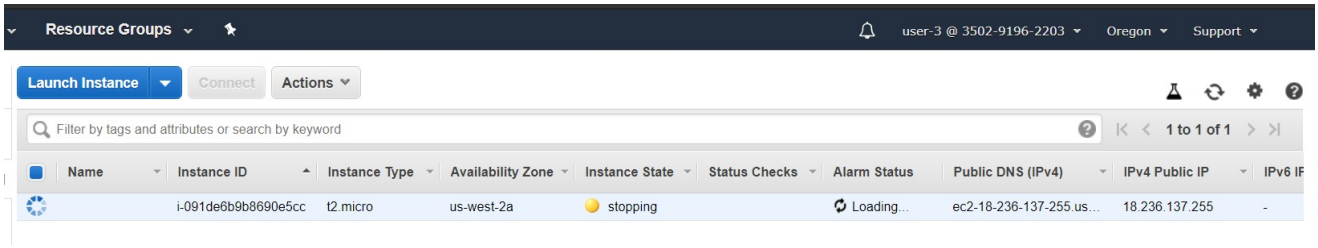
As an EC2 Administrator, you should now have permissions to Stop the Amazon EC2 instance.

Your EC2 instance should be selected . If it is not, please select it.

In the Actions menu, click Instance State > Stop.

In the Stop Instances window, click Yes, Stop.

The instance will enter the stopping state and will shutdown.



Close your private window.

EXP 2: S3- Multi-region Storage Backup with Cross-Region Replication

Lab overview

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It's a simple storage service that offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.

Cross-region replication (CRR) enables automatic, asynchronous copying of objects across buckets in different AWS Regions. Cross-Region Replication can help you do the following:

1. **Comply with compliance requirements**
2. **Minimize latency**
3. **Increase operational**
4. **Maintain object copies under different ownership**

This lab demonstrates the process of configuring Cross-Region Replication (CRR) between two S3 buckets in separate regions.

Task 1: Create and configure source and destination buckets

Before Cross-Region Replication (CRR) can be enabled, you must first create the source and destination buckets. Versioning must be enabled for both buckets in order to configure CRR. Any objects that reside in the bucket before versioning is enabled will not be replicated.

In this task, you will:

Create the source and destination buckets.

Enable versioning on each bucket.

At the top-left of the AWS Management Console, click the Services menu, and then click S3.

Click Create bucket and then configure:

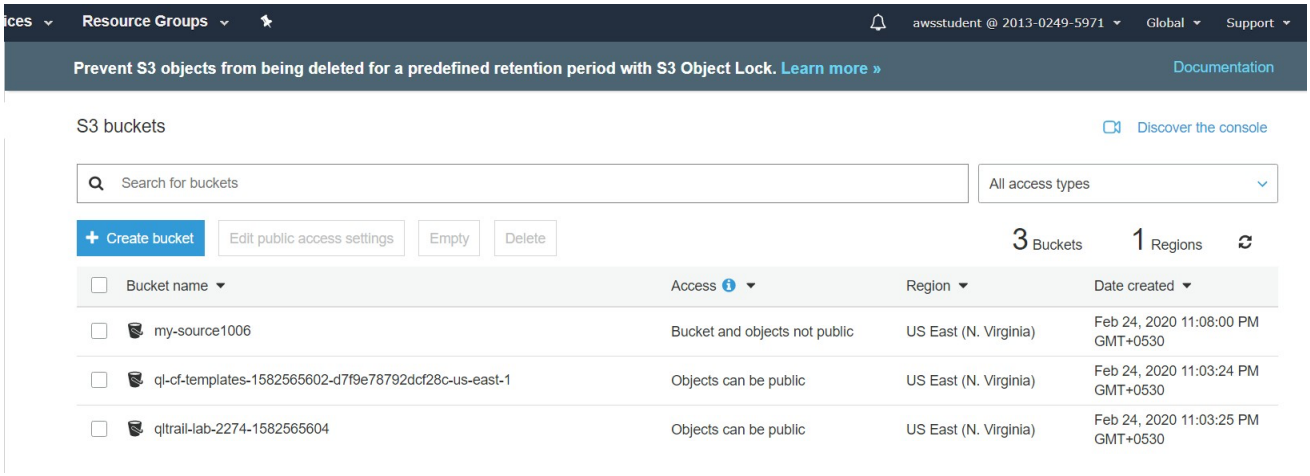
Bucket name: my-sourceNUMBER

Replace NUMBER with a random four digit number

Note: Bucket names cannot contain uppercase letters or spaces

Region: US East (N. Virginia)

Click Create to accept the default options and create the bucket.

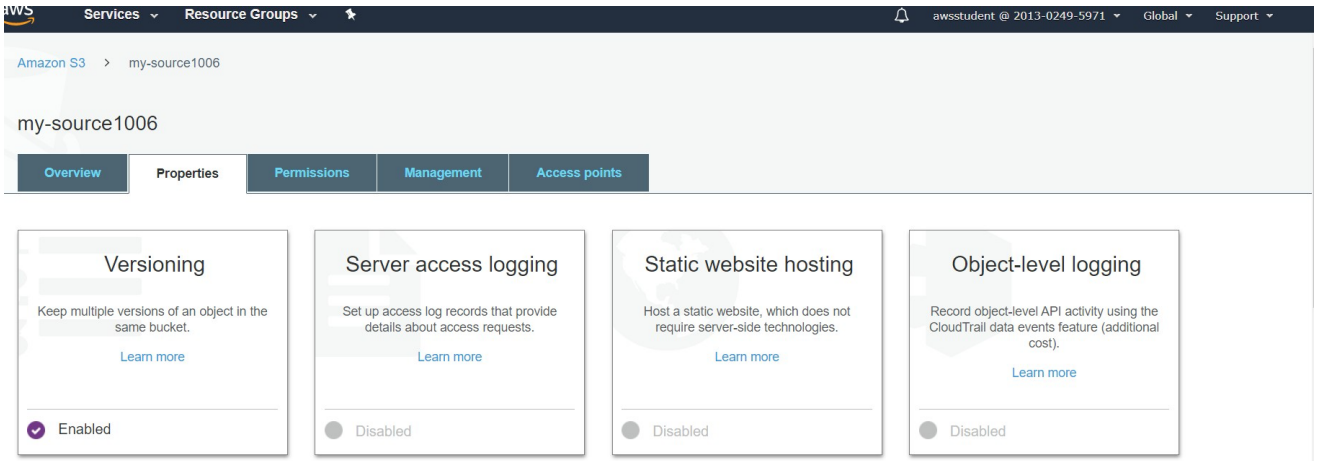


Click the name of the bucket you created in the previous step, and then click the Properties tab.

Click the Versioning card, then:

Select Enable versioning

Click Save



Now that you have created the source bucket and enabled versioning on it, you will create the destination bucket to replicate to. The destination bucket must have versioning enabled as well, but this time you will enable it using the bucket creation wizard.

Click the Amazon S3 link at the top-left to return to the S3 console main page

Click Create bucket then configure:

Bucket name: my-destinationNUMBER

Replace NUMBER with a random four digit number

Region: US West (Oregon)

Click Next

In the Versioning section, select Keep all versions of an object in the same bucket.

Doing so has the same effect as enabling versioning via the properties tab, which you did for the source bucket

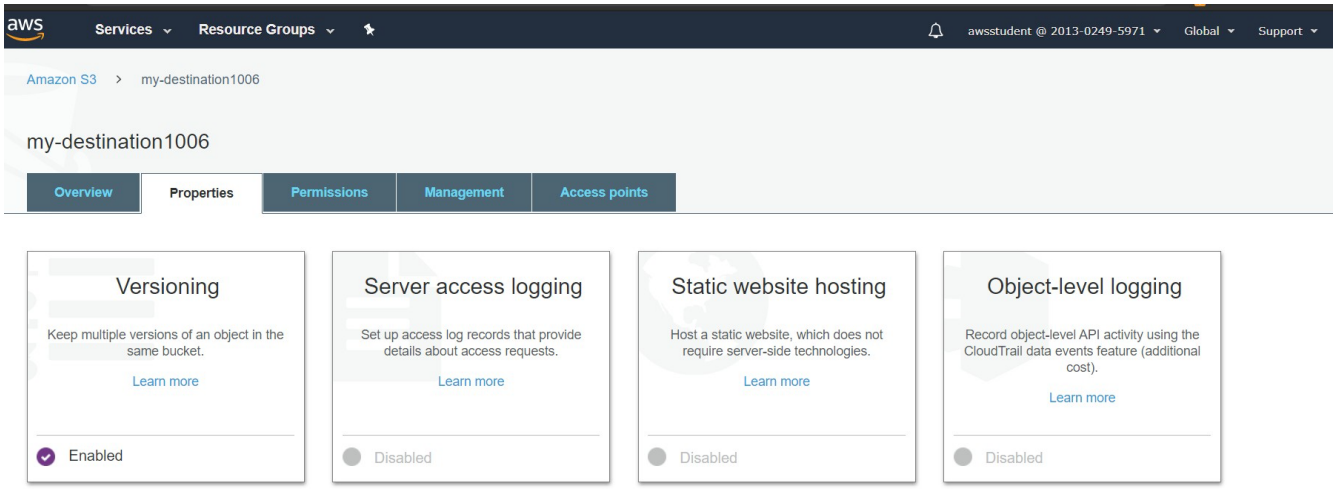
Keep the remaining default settings, and then click Next

On the Set permissions screen, keep the default settings and click Next

On the Review screen, verify the bucket name, region, and that versioning is Enabled, and then click Create bucket

Click the name of the destination bucket you just created, and then click the Properties tab.

Verify that versioning is Enabled.



Click the Amazon S3 link at the top-left to return to the S3 console main page.

Task 2: Enable Cross-Region Replication on a bucket

Now that the source and destination buckets have been created and configured, replication can be enabled. Cross-Region Replication policies are used to determine which objects in a bucket are replicated. You can replicate an entire bucket, a specific folder within a bucket, or any objects with a specified tab. However, objects that already exist in the bucket before replication is enabled will NOT be replicated.

In this task, you will:

Create files to test replication with.

Create a replication policy to enable replication of an entire bucket.

Validate the object replicated properly.

On your local system, create four text files to use throughout this task:

pre-crr.txt

crr-bucket.txt

crr-folder.txt

crr-tag.txt

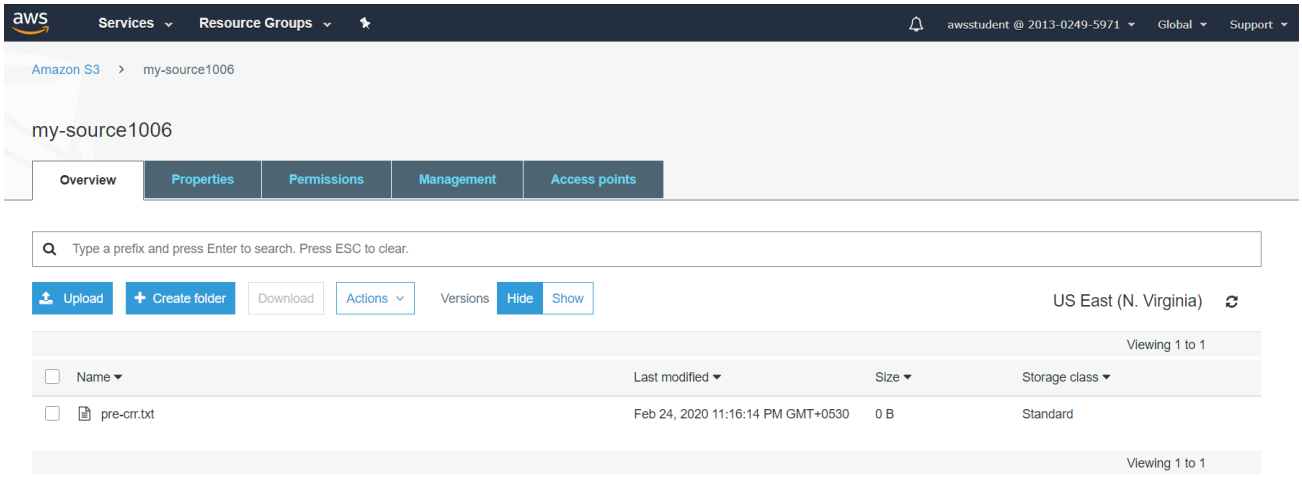
On the S3 console main page, click the name of your source bucket.

Click Upload

Click Add files and then browse to and select the pre-crr.txt file you created in step 1 of this task.

Click Upload

Verify the pre-crr.txt file is listed in the bucket.



Click the Management tab, and then click Replication

Click Add rule and then configure:

Set source: Entire bucket

Click Next

Destination bucket: Buckets in this account

Click your my-destination bucket - You may have to scroll down to see the bucket

Click Next

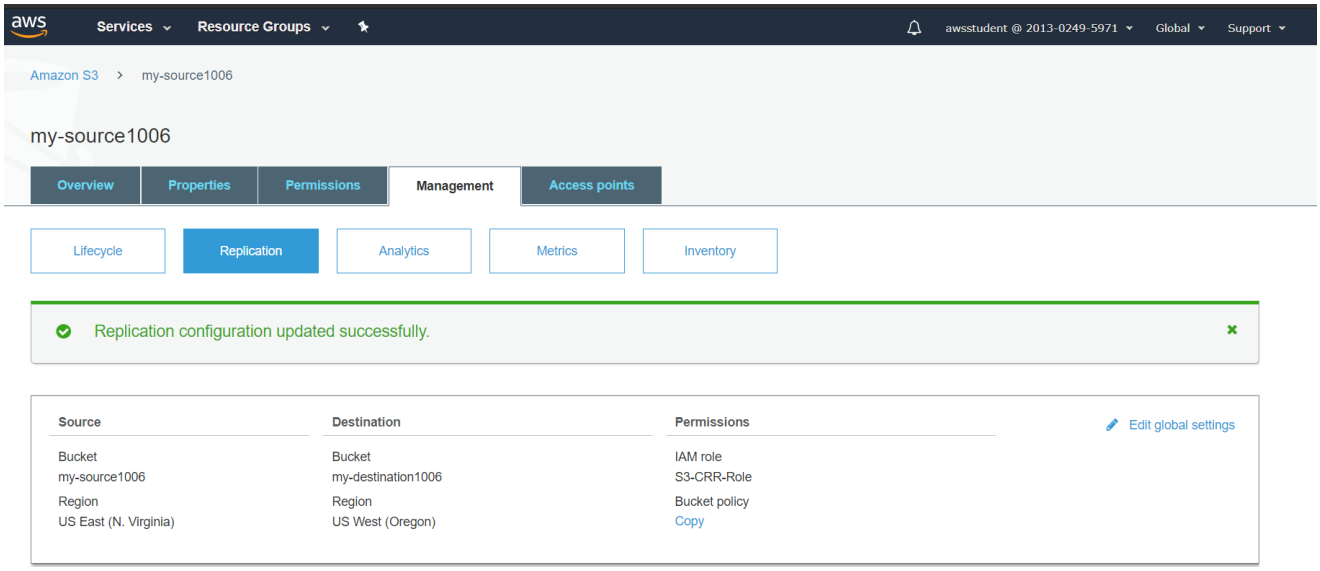
On the Configure rule options screen, configure:

IAM role: S3-CRR-Role

Rule name: crr-full-bucket

Click Next

Review your settings, and then click Save

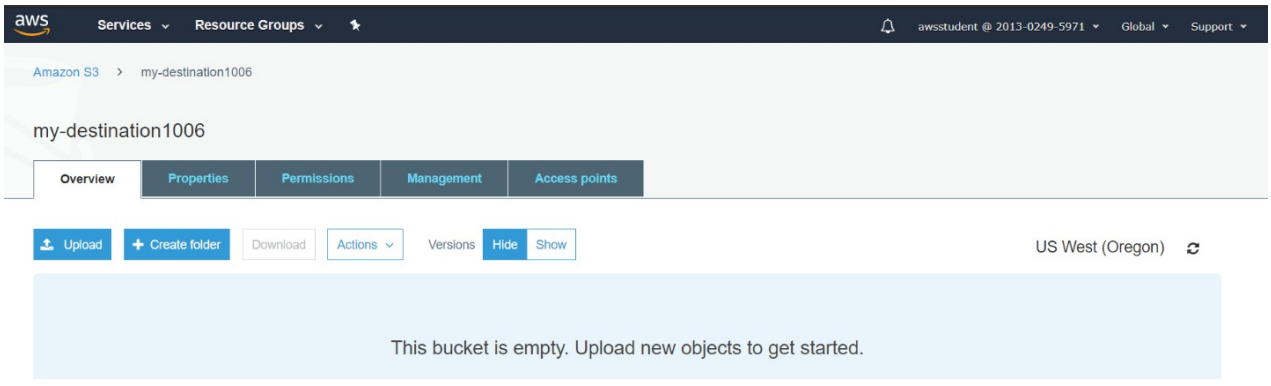


Now that replication is enabled on the bucket, verify whether or not the file in the bucket has been replicated.

Click the Amazon S3 link at the top-left to return to the S3 console main page.

Click the name of your destination bucket to open it.

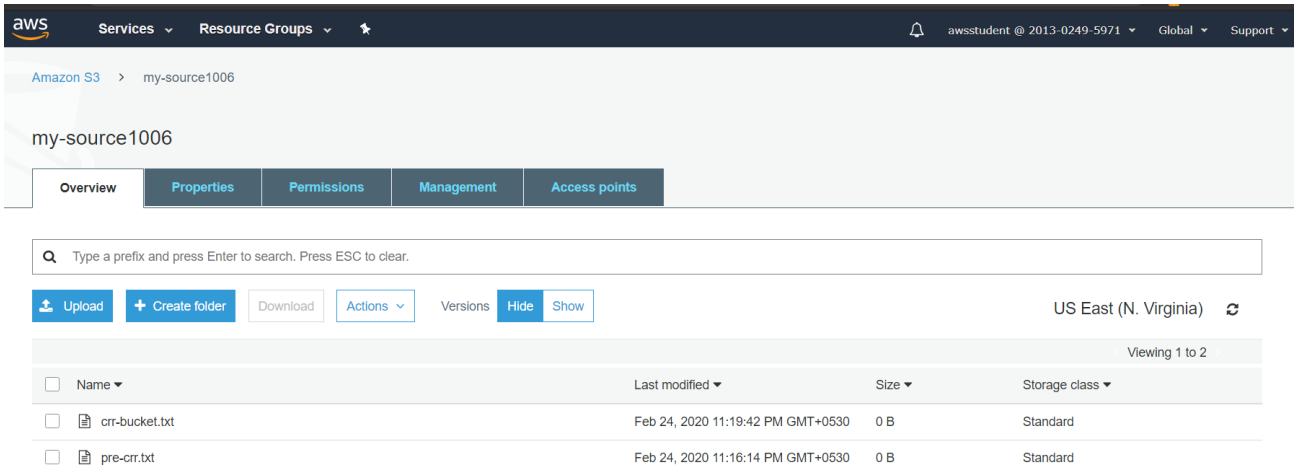
Notice that the destination bucket is empty, even though replication is enabled and the source bucket contains a file. This is because only new files uploaded to the source bucket after replication is enabled will be replicated to the destination bucket.



Click the Amazon S3 link at the top-left to return to the S3 console main page.

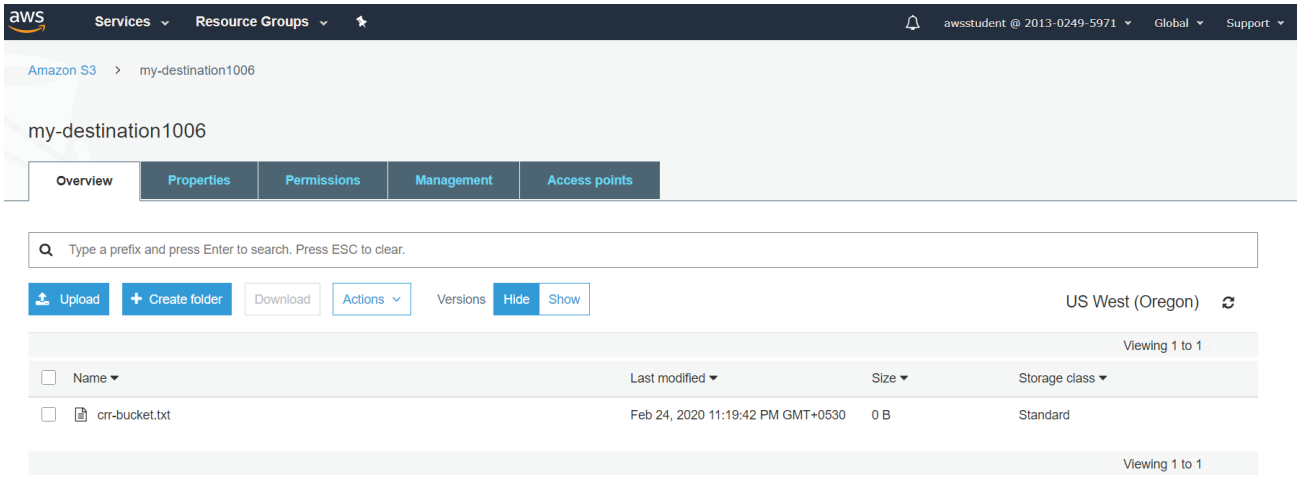
Click the name of your source bucket.

Upload the crr-bucket.txt file to your bucket.



Click the Amazon S3 link at the top-left to return to the S3 console main page, and then click the name of your destination bucket to open it.

Notice that the destination bucket now contains the crr-bucket.txt file you uploaded to the source bucket.



Task 3: Configure replication of a single folder

In Amazon S3, folders are considered prefixes. For example, a folder in your S3 bucket named Source would be a prefix notated as Source/. A file inside that folder would be notated as Source/File.

Knowing this allows you to create a replication policy based on a prefix, which will include all objects in the folder. Choosing a folder to replicate allows you to replicate a specific set of objects easily, rather than an entire bucket.

In this task, you will:

Create a replication policy to enable replication of the contents of a folder (prefix).

Create the folder to replicate and upload a file into it.

Validate the object replicated properly.

If you are not already on the S3 console main page, click the Amazon S3 link at the top-left to navigate to it.

Click the name of your source bucket.

Click Create folder and then configure:

Folder name: crr-test

Click Save

Click the Management tab, and then click Replication

Select the crr-full-bucket policy you created previously.

Click the Actions menu, and then click Disable rule.

Click Confirm to disable the rule.

Click Add rule

Under Set source, click Prefix or tags, and then:

Enter crr-test/

Click prefix crr-test

Click Next

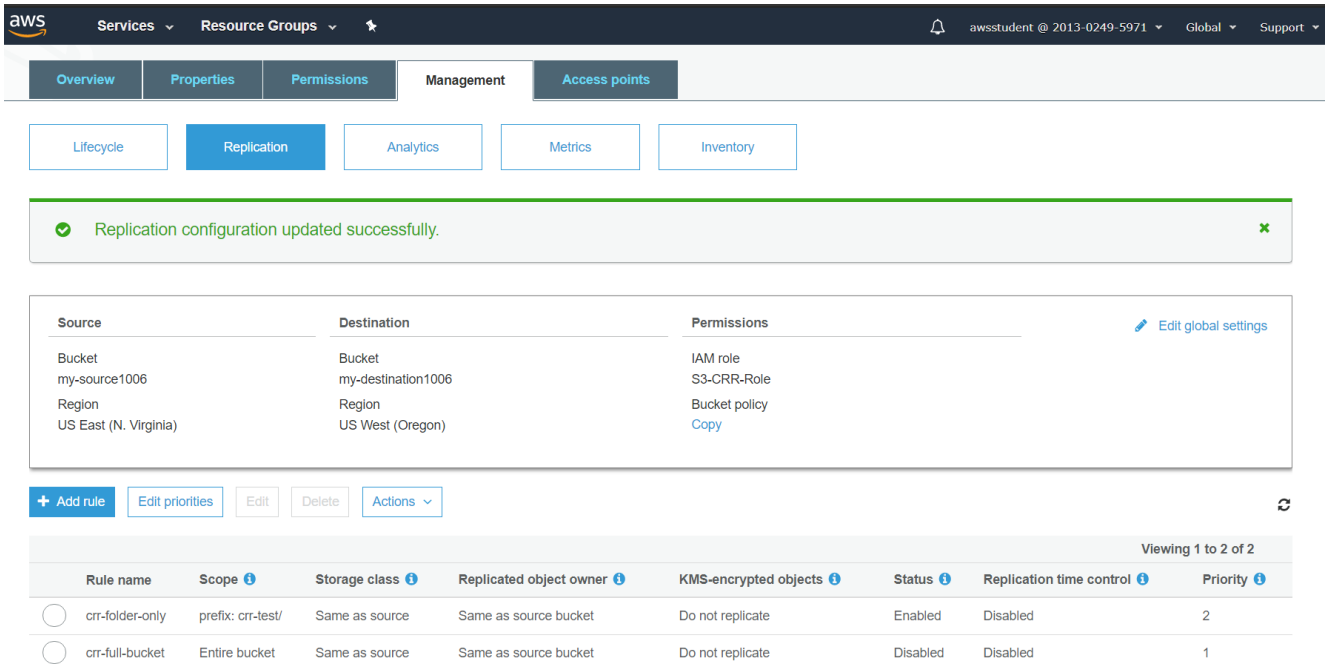
On the Set destination screen, verify your destination bucket is displayed, and then click Next

On the Configure rule options screen, configure:

Rule name: crr-folder-only

Click Next

On the Review screen, verify your settings, and then click Save



Click the Overview tab to return to the bucket file list.

Open the crr-test folder.

In the crr-test folder, upload your crr-folder.txt file.

Navigate to your my-destination bucket.

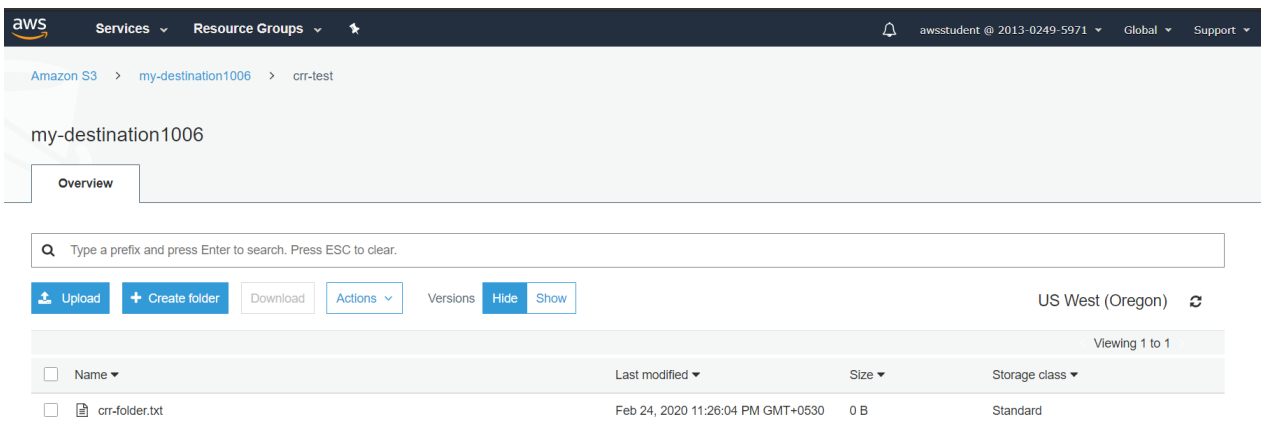
You should see two objects listed in the destination folder:

crr-test (folder)

crr-bucket.txt (file)

Open the crr-test folder. You should see the crr-folder.txt file you uploaded to the source bucket.

Click the Amazon S3 link at the top-left to return to the S3 console main page



Congratulations! You've successfully configured Cross-Region Replication for a single folder within an S3 bucket.

Task 4: Configure replication using tags

Tags can be used to identify specific objects to replicate, rather than replicating the entire bucket or folder.

In this task, you will:

Create a replication policy to replicate any object with a specific tag.

Upload a file and add the tag to it.

Validate the object replicated properly.

Much like versioning, objects with tags must be uploaded to the source bucket after the replication policy using tags has been created and enabled. Objects that are uploaded and tagged prior to the policy being created will not replicate.

If you are not already on the S3 console main page, click the Amazon S3 link at the top-left to navigate there.

On the S3 console main page, click the name of your source bucket.

Click the Management tab, and then click Replication

Click Add rule and then configure:

Set source: Prefix or tags

Enter replicate

Click tag replicate

Enter yes

Press Enter

Tag keys and values are case sensitive. Refer to the additional resources section at the end of the lab for more information.

Click Next

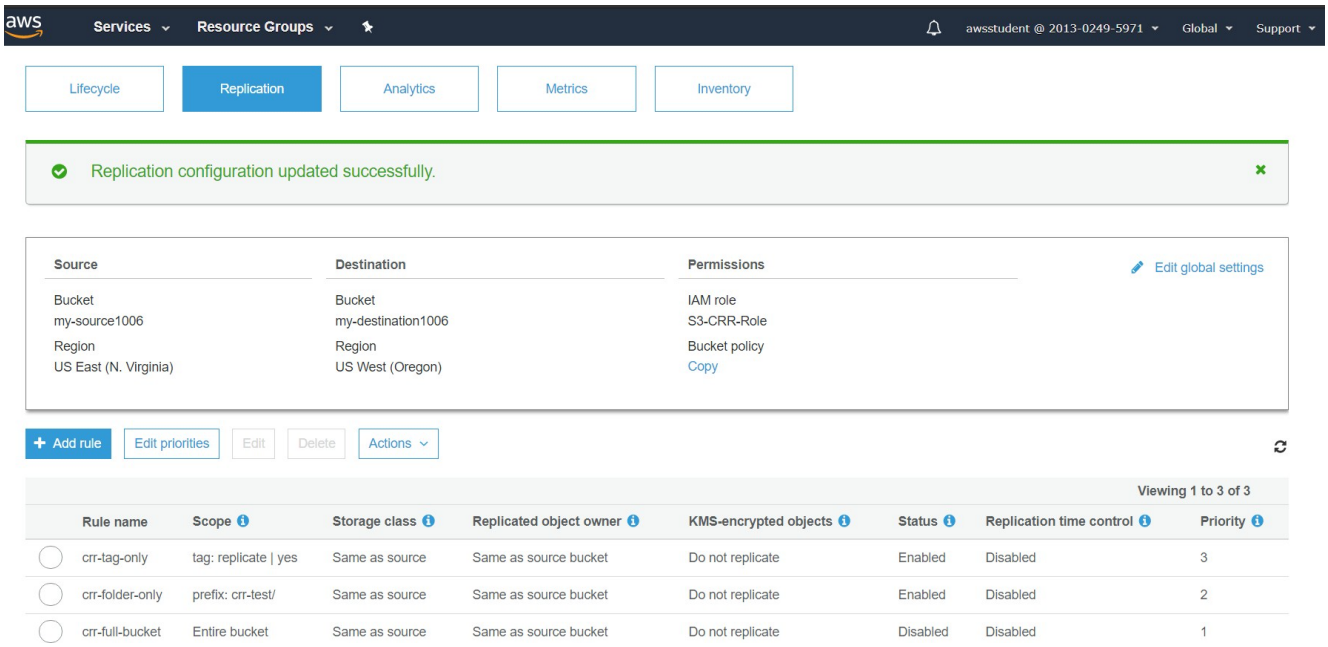
On the Set destination screen, verify your destination bucket is displayed, and then click Next

On the Configure rule options screen, configure:

Rule name: crr-tag-only

Click Next

On the Review screen, verify your settings, and then click Save



Click the Overview tab.

Click Upload

Click Add files

Select the crr-tag.txt file you created previously, and then click Next

On the Set permissions screen, accept the defaults, and then click Next

On the Set properties screen, scroll down to the Tag section.

In the Tag section, configure:

Key: replicate

Value: yes

Tags are case-sensitive.

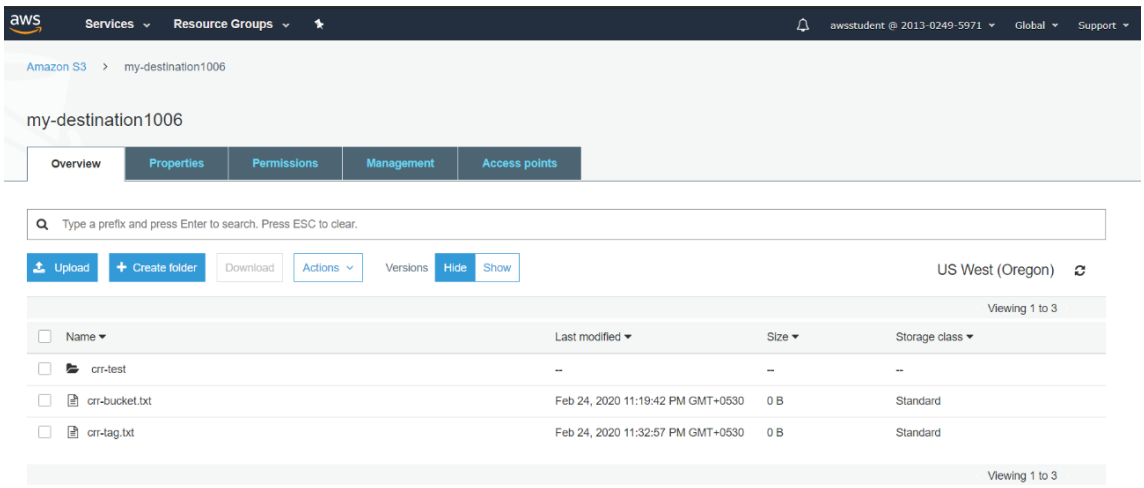
Click Save and then click Next

On the Review screen, review the settings, and then click Upload

Verify the crr-tag.txt file was uploaded successfully.

Navigate to your destination bucket.

You should notice that the destination bucket now contains the crr-tag.txt file you just uploaded to the source bucket.



If no files are listed, wait a few seconds, then click the refresh button at the top-right.

Click the Amazon S3 link at the top-left to return to the S3 console main page.

Congratulations! You've successfully configured Cross-Region Replication using a tag.

Task 5: Deleting Replicated Files

To protect against malicious intent and accidental deletion, object deletions that occur in a source bucket are not replicated to the destination bucket.

In this task, you will:

Delete a file that has been replicated, then observe the results.

If you are not already on the S3 console main page, click the Amazon S3 link at the top-left to navigate there.

Navigate to your source bucket.

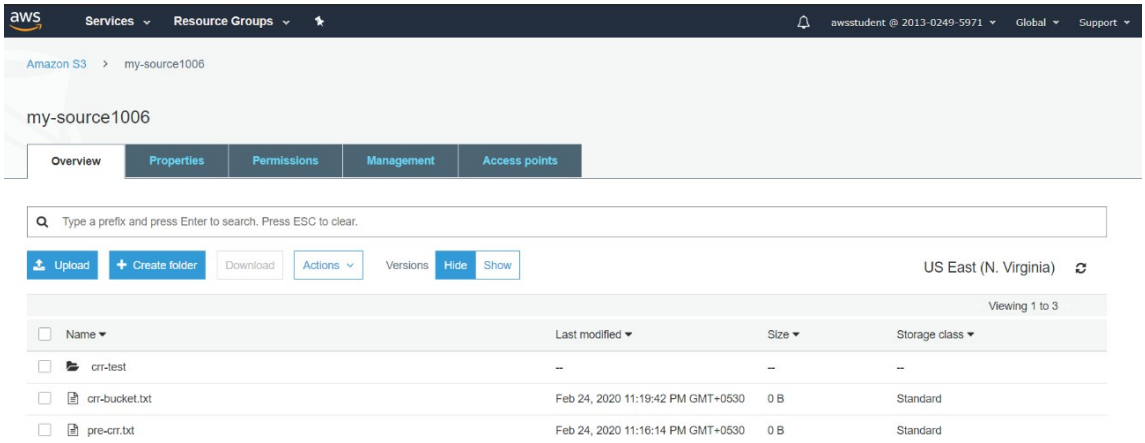
In your source bucket:

Select crr-tag.txt

Click the Actions menu, and then click Delete.

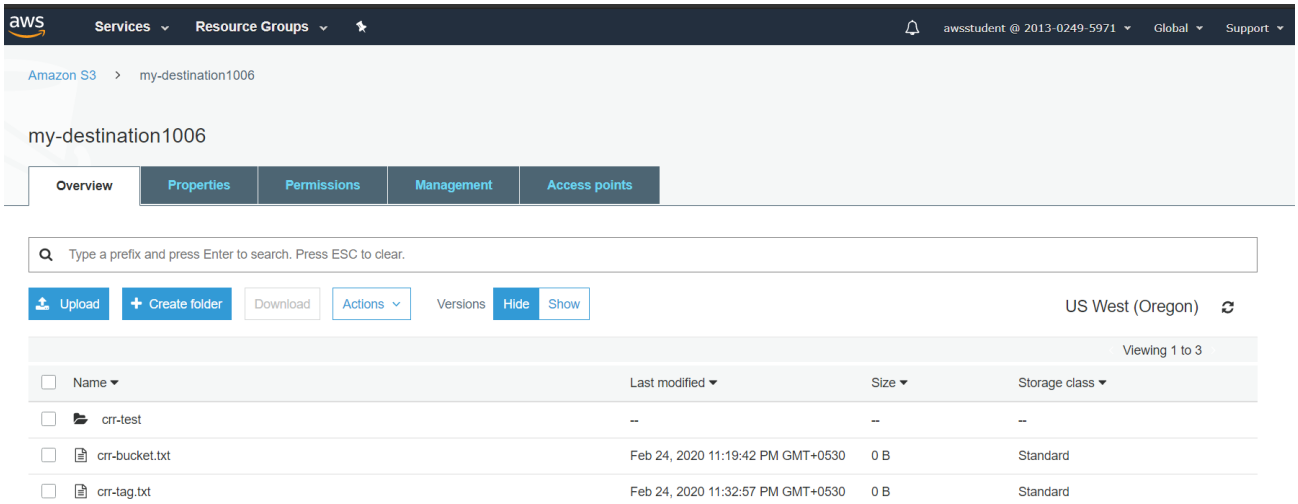
In the confirmation window that opens, click Delete

Verify the crr-tag.txt file has been deleted from the source bucket.



Navigate to your destination bucket.

You should notice that the crr-tag.txt file still exists in the destination bucket.



EXP 3 : Introduction to Amazon API Gateway

Task 1: Create a Lambda Function

In the AWS Management Console, on the Services menu, click Lambda.

Click Create function

Below Author from scratch, Configure:

Function name: FAQ

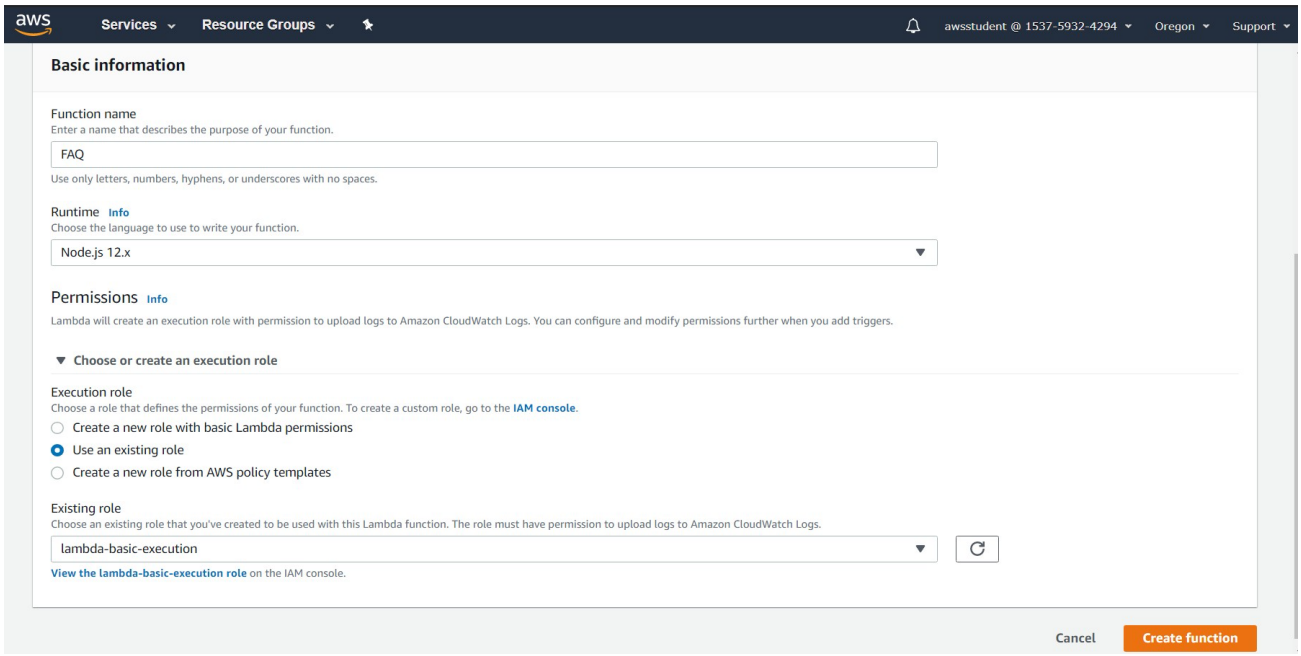
Runtime: Node.js 12.x

Expand Choose or create an execution role

Execution role: Use an existing role

Existing role: lambda-basic-execution

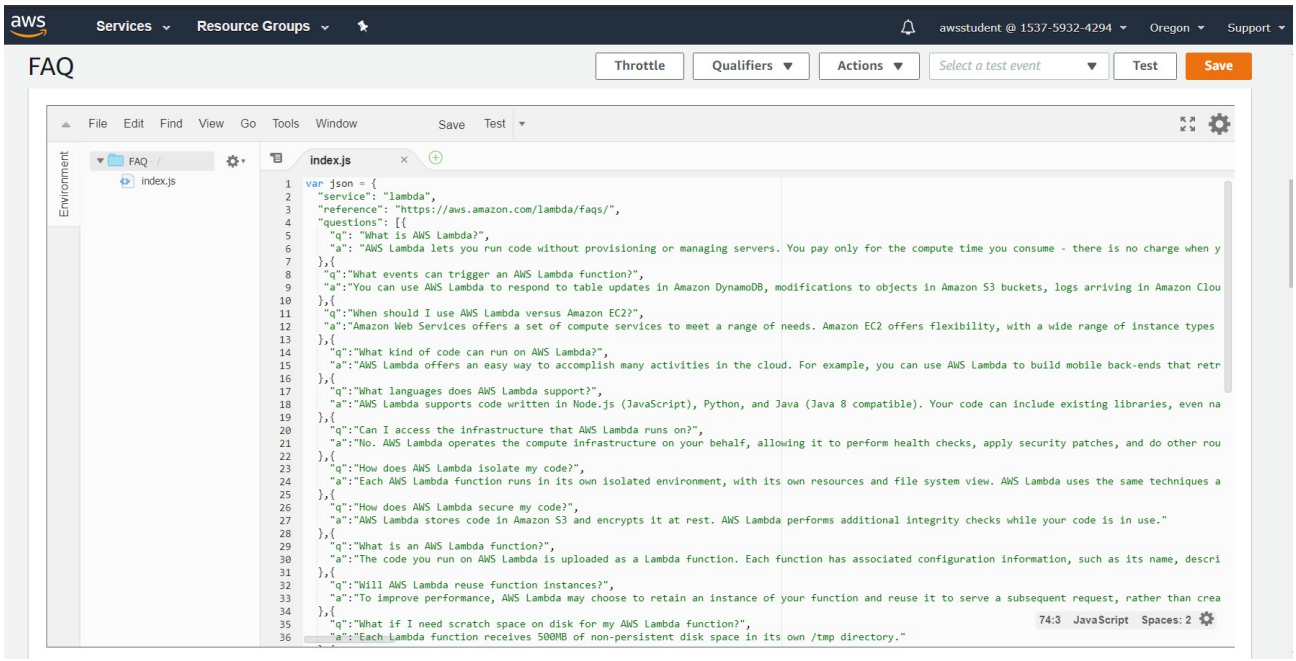
Click Create function



A page is displayed with the function configuration.

Scroll down to the Function code section and delete all of the code that appears in the code editor.

Copy the code below and paste it into the index.js tab.



The code performs the following steps:

- Defines a list of Frequently Asked Questions (FAQs)
- Returns a random FAQ
- Scroll down to the Basic settings section, then click Edit

For Description, enter:

Provide a random FAQ

Click Save

Scroll up to the Designer section.

Create an API Gateway endpoint.

Click Add trigger then configure:

Select a trigger: API Gateway

API: Create a new API

Choose a template: Rest API

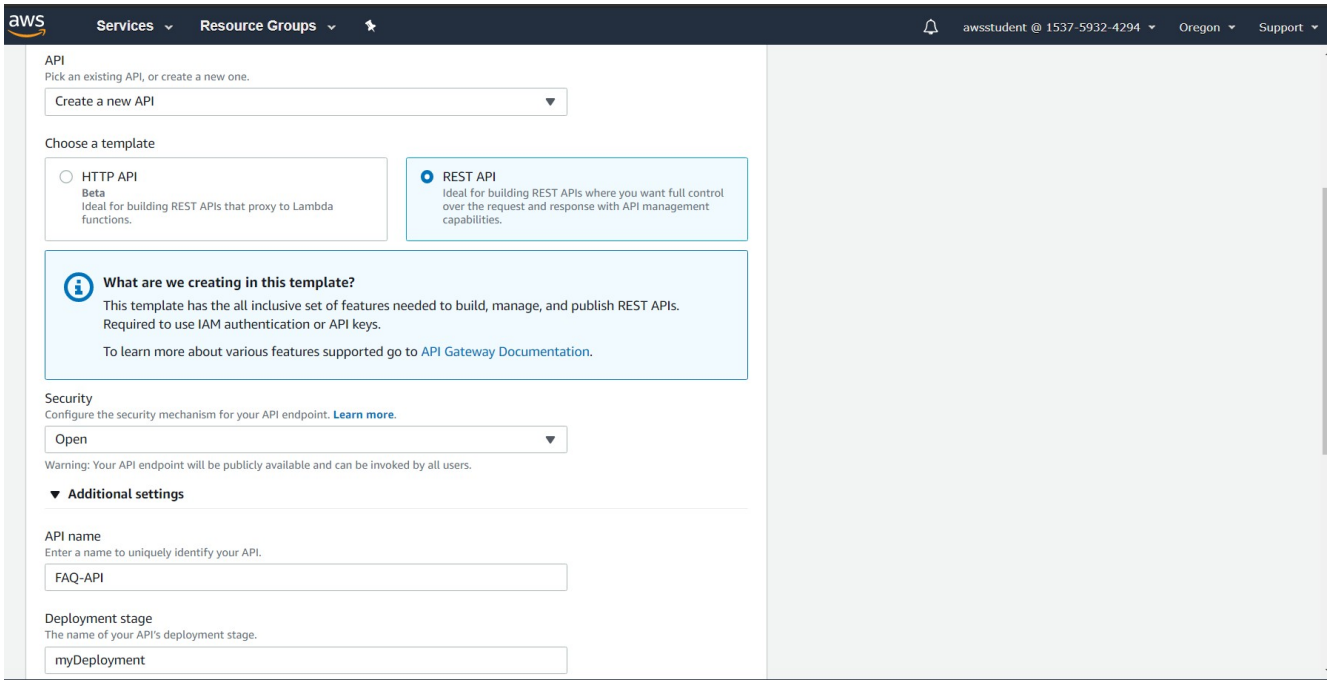
Security: Open

Expand Additional settings

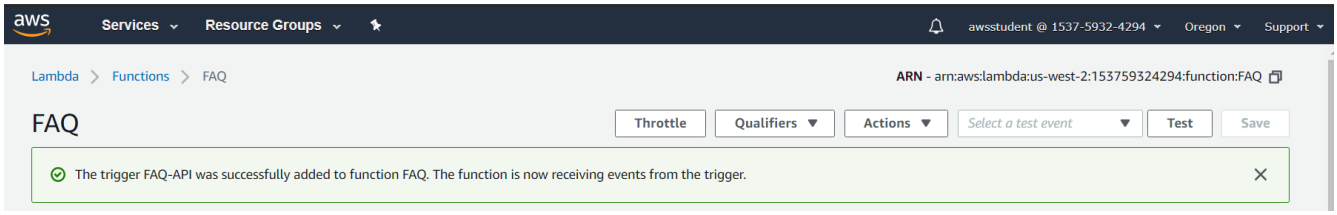
API name: FAQ-API

Deployment stage: myDeployment

At the bottom right-side of the screen, click Add.



Click Save at the top right corner.



Task 2: Test the Lambda function

On the FAQ Lambda function page,

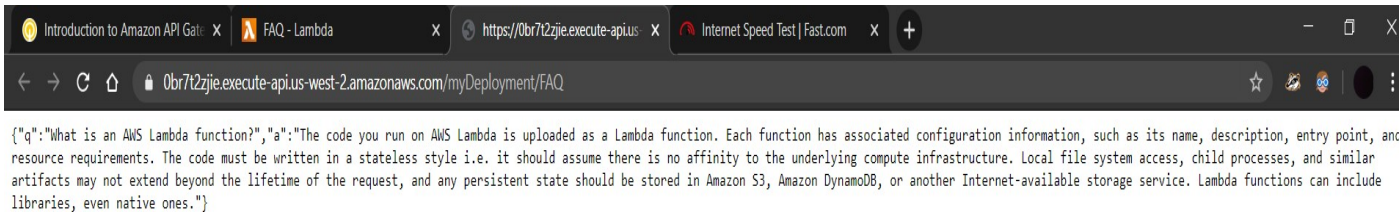
Under API Gateway, click the right arrow to view the details of the API.

Copy the API endpoint to the clipboard, then:

In a new browser tab, paste the API endpoint

Press Enter to go to the URL

A new browser tab will open. A random FAQ entry can be seen, such as:

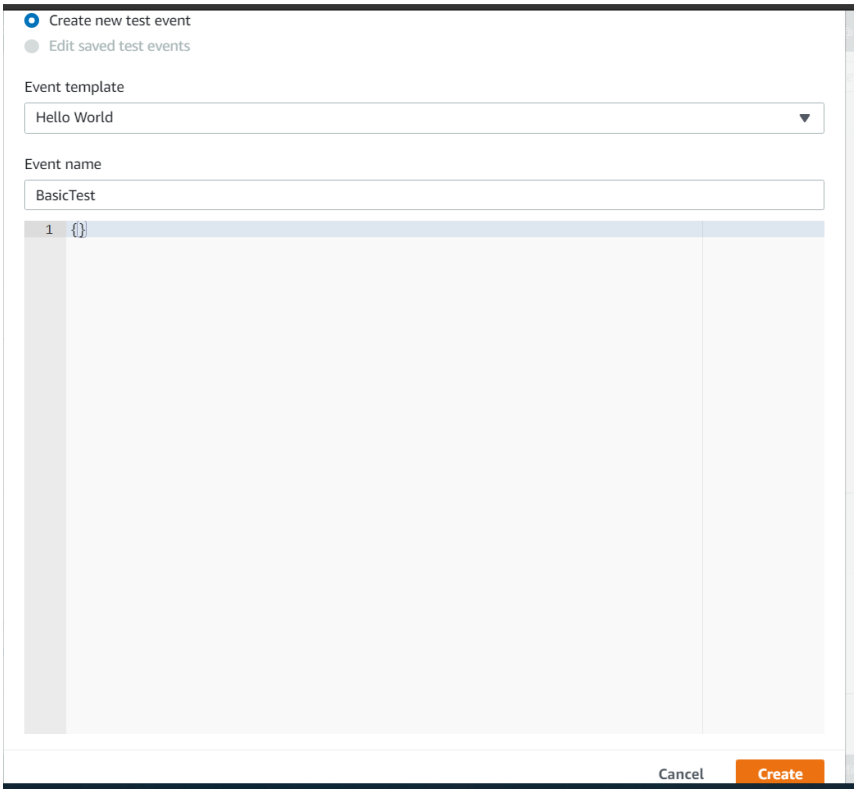


The Lambda function can also be tested in isolation.

At the top of the screen, Click Test then configure:

Event name: BasicTest

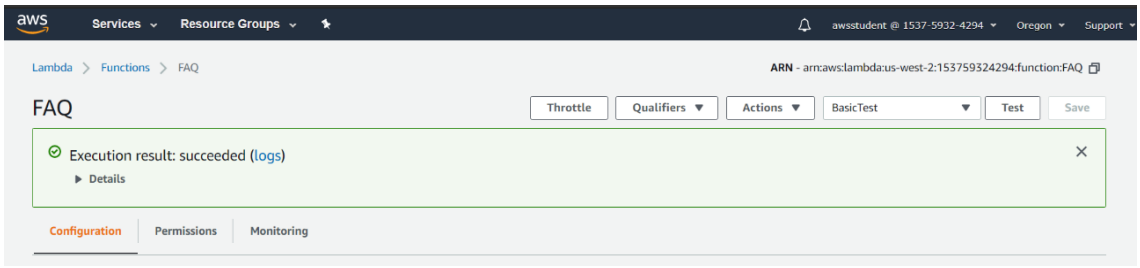
Delete the provided keys and values, retaining an empty {} to represent an empty JSON object:{}



At the bottom of the screen, click Create

At the top of screen, Click Test

In the Execution result: succeeded window, expand Details.

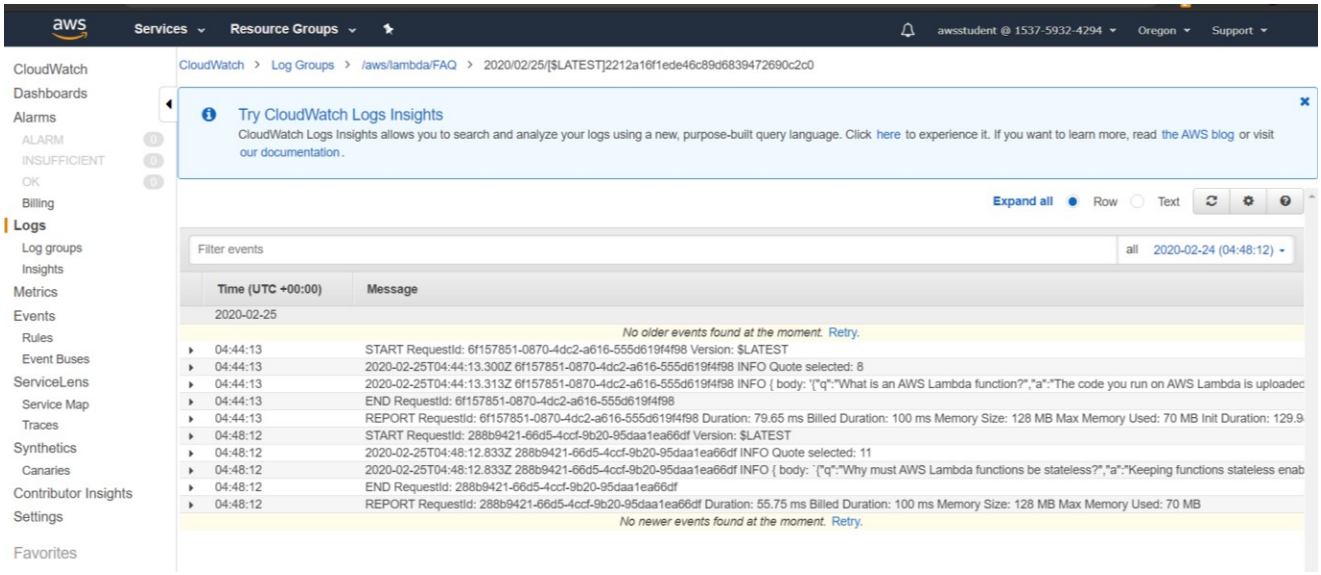


The output shows the FAQ entry wrapped inside a body parameter.

Below the Execution result are two columns. The Summary displays the total execution time for the Lambda function and the resources consumed. The Log output displays logging information.

Click the Monitoring tab.

Click View logs in CloudWatch



Click on one of the log streams.

EXP 4: Introduction to AWS Lambda

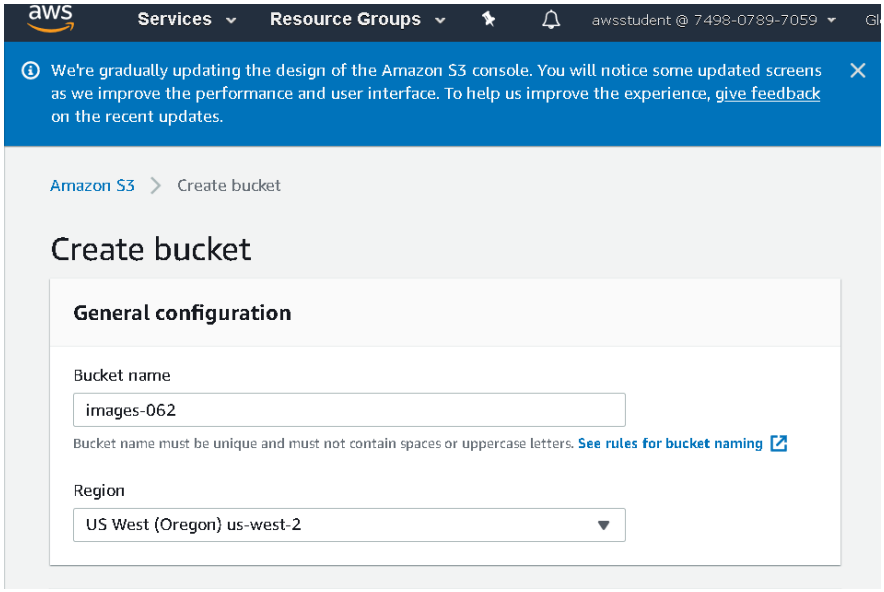
Task 1: Create the Amazon S3 Buckets

In this task, create two Amazon S3 buckets -- one for input and one for output.

Amazon S3 buckets require unique names, so add a random number to the bucket name.

In the AWS Management Console, on the Services menu, click S3.

Click Create bucket and then configure:



Bucket name: images-NUMBER

Replace NUMBER with a random number

Copy the name of your bucket to a text editor

Click Create bucket

Now create another bucket for output.

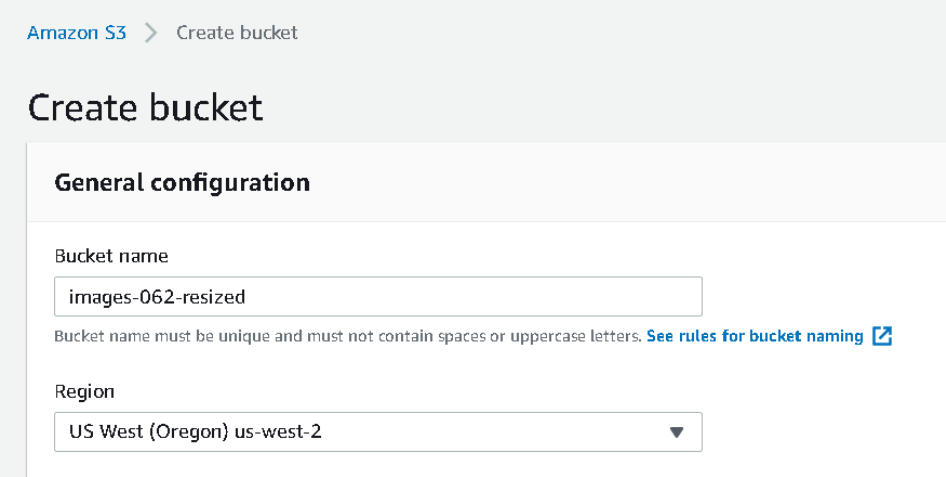
Click Create bucket and then configure:

Bucket name: Paste the name of the images bucket

At the end of the bucket name, append -resized

Click Create bucket

Do not change the Region.



Now upload a picture for testing purposes.

Right-click this link and download the picture to the computer: HappyFace.jpg

Name the file HappyFace.jpg.

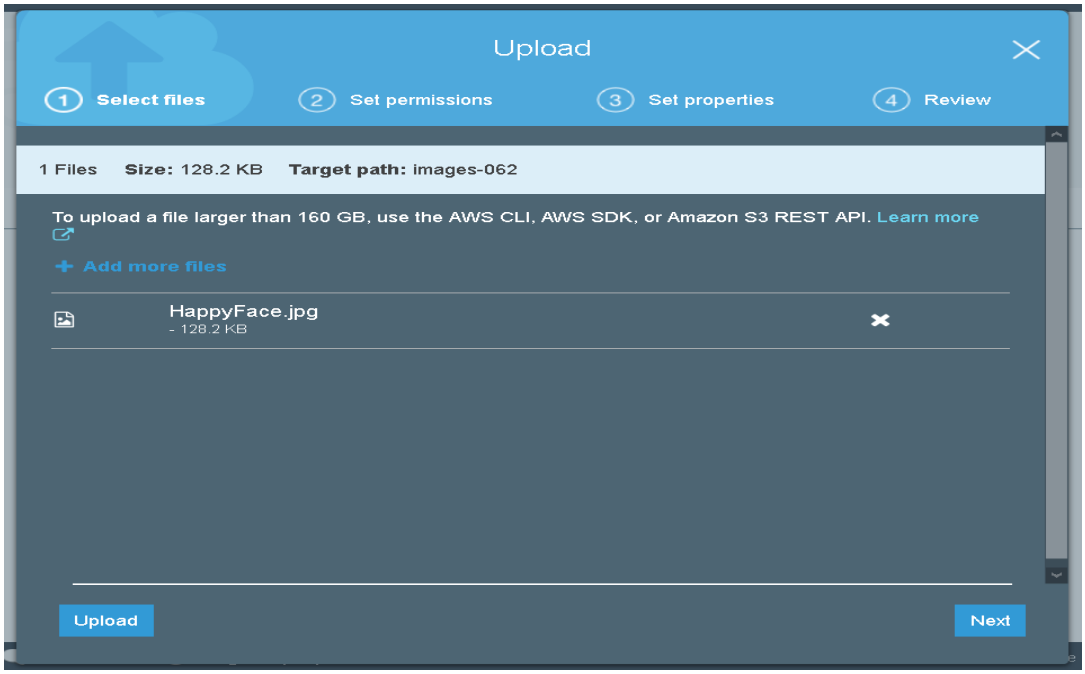
Open the image on the computer.It is a large picture, with dimensions of 1280 x 853.

In the S3 Management Console, click the images- bucket. (Not the -resized bucket)

Click Upload

In the Upload window, click Add files

Browse to and select the HappyFace.jpg picture.



Click Upload

Later in this invoke the Lambda function manually by passing sample event data to the function. The sample data will refer to this HappyFace.jpg image.

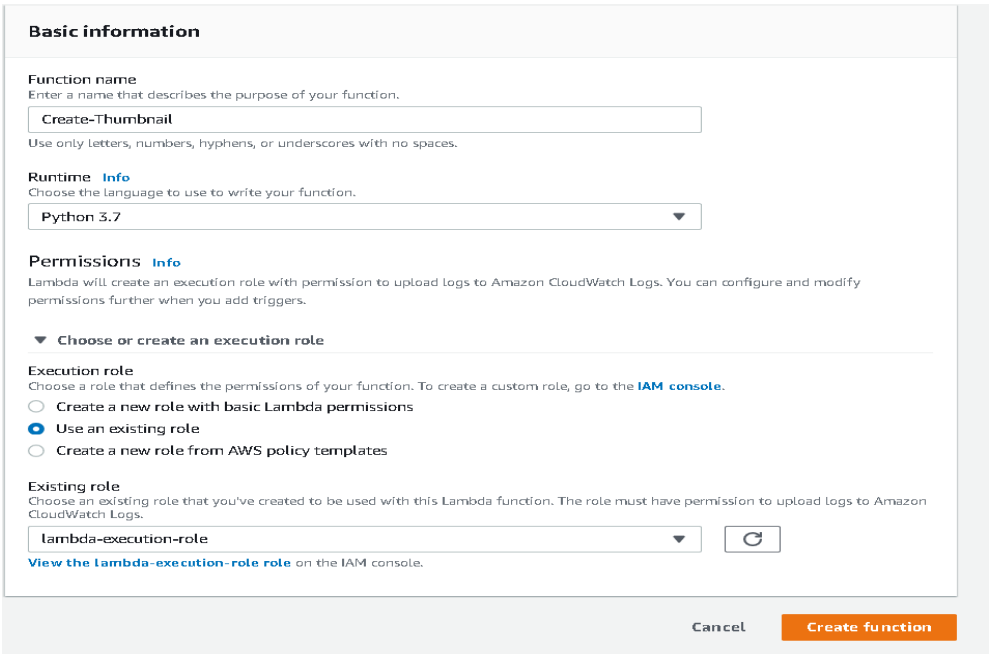
Task 2: Create an AWS Lambda Function

In this task, create an AWS Lambda function that reads an image from Amazon S3, resizes the image and then stores the new image in Amazon S3.

On the Services menu, click Lambda.

Do not change the Region.

Click Create function



In the Create function window, configure:

- Function name: Create-Thumbnail
- Runtime: Python 3.7

Make sure to select Python 3.7 under Other Supported runtime

Expand Choose or create an execution role

Execution role: Use an existing role

Existing role: lambda-execution-role

This role grants permission to the Lambda function to access Amazon S3 to read and write the images.

Click Create function

A page will be displayed with the function configuration.

Click Add trigger then configure:

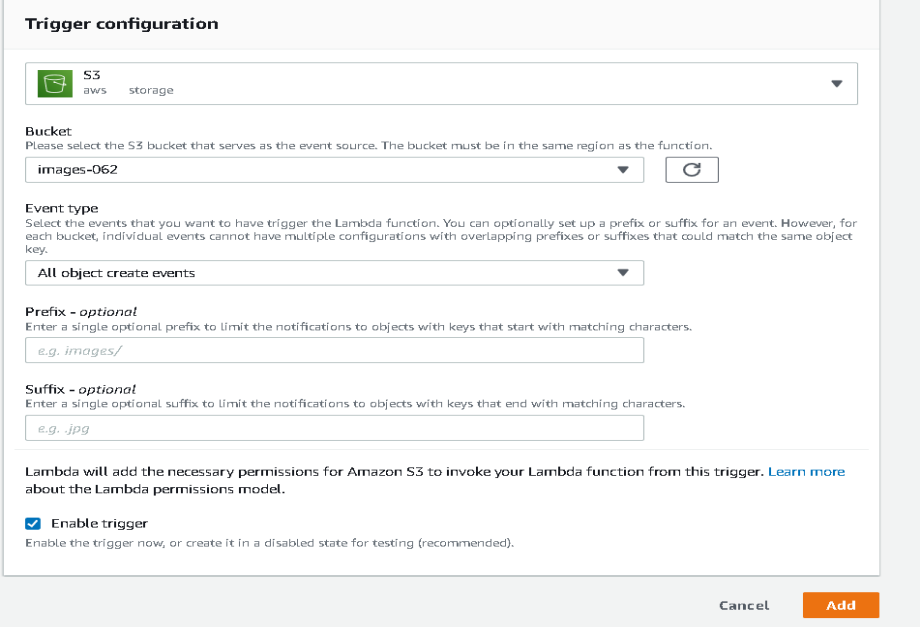
Select a trigger: S3

Bucket: Select your images- bucket (e.g. images-123)

Event type: All object create events

Scroll to the bottom of the screen, then click Add

Click Create-Thumbnail at the top of the diagram:



Now configure the Lambda function.

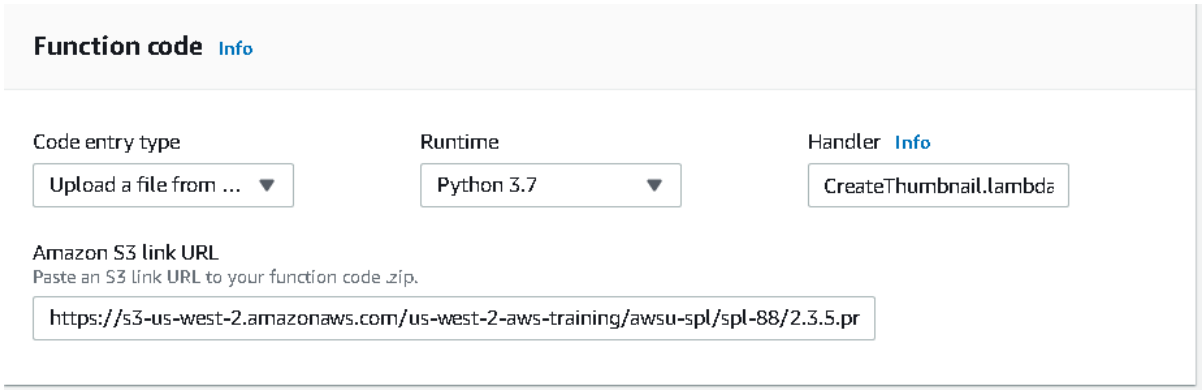
Scroll down to the Function code section and configure the following settings (and ignore any settings that aren't listed):

Code entry type: Upload a file from Amazon S3

Runtime: Python 3.7

Handler: CreateThumbnail.handler

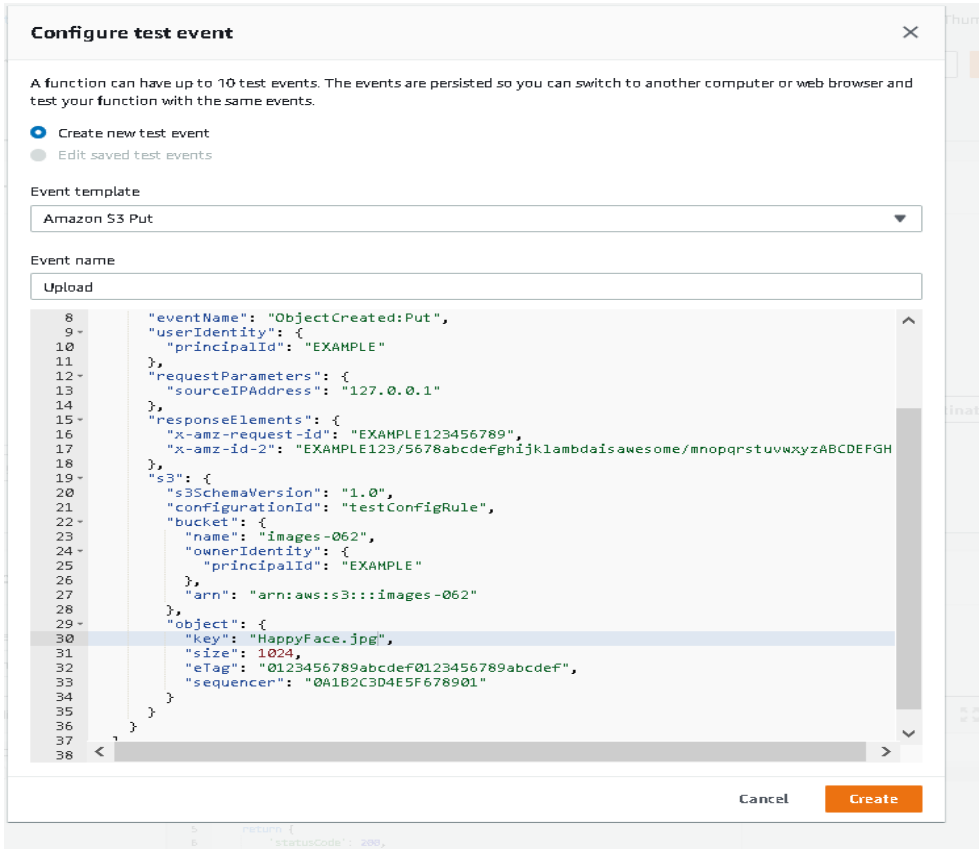
Make sure to set the Handler field to the above value, otherwise the Lambda function will not be found.



Amazon S3 link URL: Copy and paste this URL into the field:

https://s3-us-west-2.amazonaws.com/us-west-2-aws-training/awsu-spl/spl-88/2.3.5.prod/scripts/CreateThumbnail.zip

The CreateThumbnail.zip file contains the following Lambda function:



Examine the code. It is performing the following steps:

Receives an Event, which contains the name of the incoming object (Bucket, Key)

Downloads the image to local storage

Resizes the image using the Pillow library

Uploads the resized image to the -resized bucket

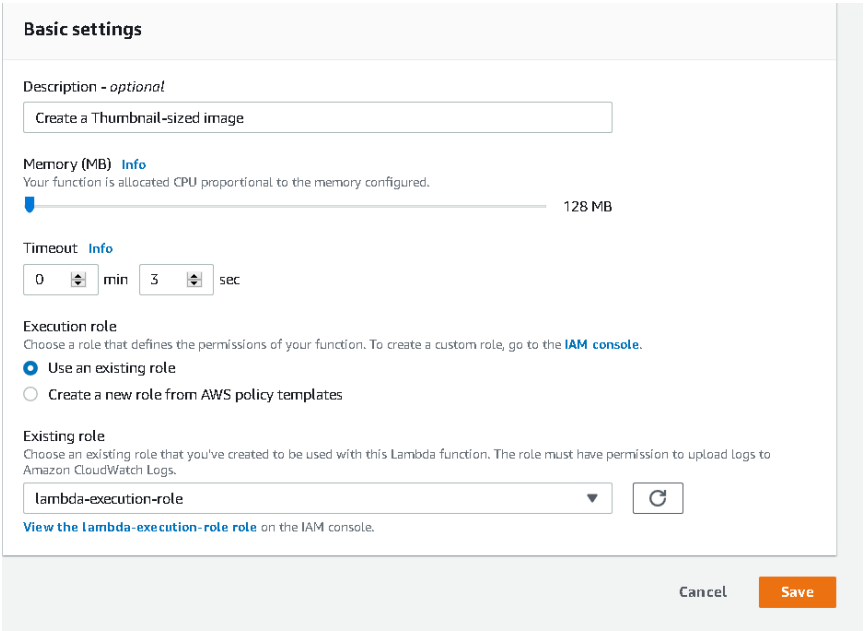
In the Basic settings section towards the bottom of the page, click Edit

Description enter:

Create a thumbnail-sized image

Click Save

You will leave the other settings as default, but here is a brief explanation of these settings:



Memory defines the resources that will be allocated to the function. Increasing memory also increases CPU allocated to the function.

Timeout sets the maximum duration for function execution.

VPC (under Network) provides the Lambda function access to resources within a Virtual Private Cloud (VPC) network.

Ignore the warning message "You don't have permission to configure a VPC

Dead Letter Queue (DLQ) Resource (under Debugging and error handling) defines how to handle failed function executions.

Enable active tracing allows tracing and monitoring of distributed code via AWS X-Ray.

Click Save at the top of the window.

Lambda function has now been configured.

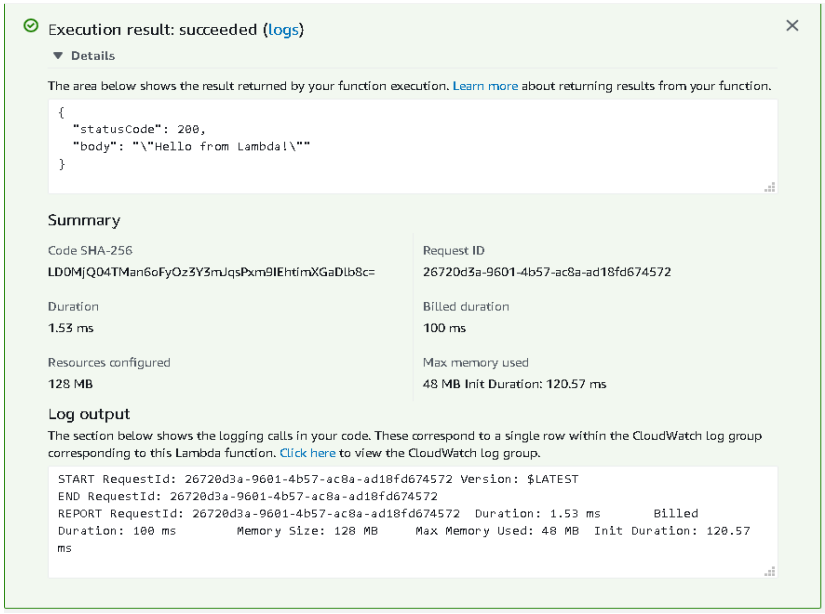
Task 3: Test Your Function

In this task, test your Lambda function. This is done by simulating an event with the same information normally sent from Amazon S3 when a new object is uploaded.

At the top of the screen, click Test then configure:

Event template: Amazon S3 Put

Event name: Upload



A sample template will be displayed that shows the event data sent to a Lambda function when it is triggered by an upload into Amazon S3. Edit the bucket name so that it uses the bucket created earlier.

Replace example-bucket with the name of the images bucket (e.g. images-123) that was copied to the text editor.

Be sure to replace example-bucket in both locations.

Replace test/key with the name of the picture uploaded. This should be HappyFace.jpg

Click Create

Click Test

AWS Lambda will now trigger the function, using HappyFace.jpg as the input image.

Towards the top of the page see the message: Execution result: succeeded

Click Details to expand it (towards the top of the screen).

Information will be shown including:

- Execution duration
- Resources configured
- Maximum memory used
- Log output

Now view the resized image that was stored in Amazon S3.

On the Services menu, click S3.

Click the name of the -resized bucket, then:

Click HappyFace.jpg

Click Open (If the image does not open, disable pop-up blocker.)

The image should now be a smaller thumbnail of the original image.

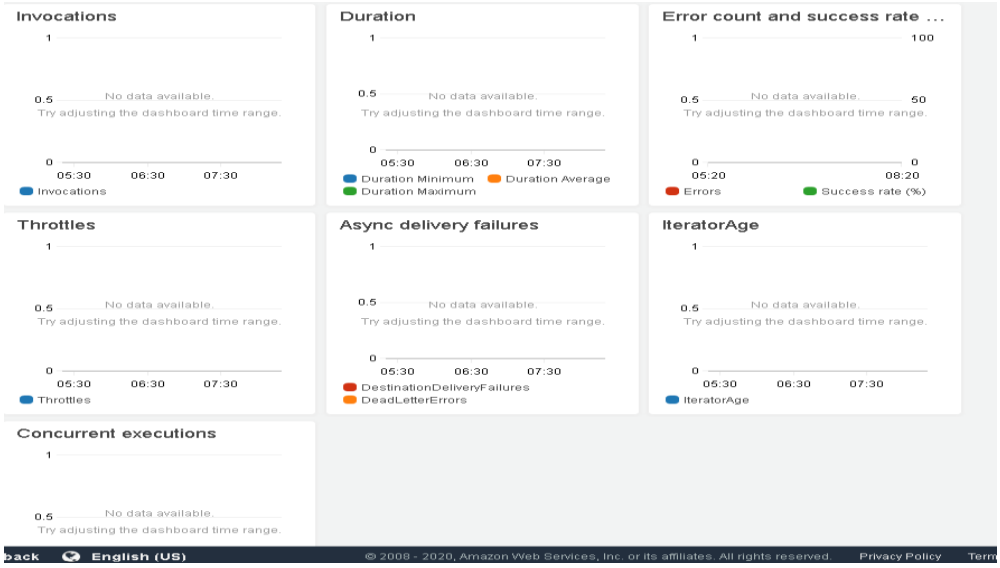
Task 4: Monitoring and Logging

Monitor AWS Lambda functions to identify problems and view log files to assist in debugging.

On the Services menu, click Lambda.

Click the Create-Thumbnail function.

Click the Monitoring tab.



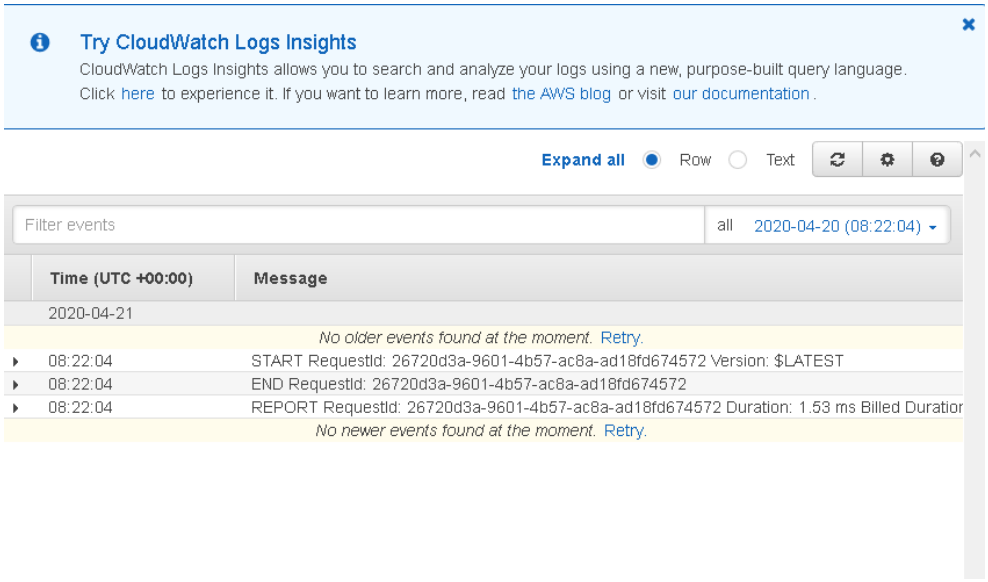
The console displays graphs showing:

- Invocations: The number of times the function has been invoked.
- Duration: How long the function took to execute (in milliseconds).
- Errors: How many times the function failed.
- Throttles: When too many functions are invoked simultaneously, they will be throttled. The default is 1000 concurrent executions.
- Iterator Age: Measures the age of the last record processed from streaming triggers (Amazon Kinesis and Amazon DynamoDB Streams).
- Dead Letter Errors: Failures when sending messages to the Dead Letter Queue.
- Log messages from Lambda functions are retained in Amazon CloudWatch Logs.

Click View logs in CloudWatch

Click the Log Stream that appears.

Expand each message to view the log message details.


Try CloudWatch Logs Insights
CloudWatch Logs Insights allows you to search and analyze your logs using a new, purpose-built query language.
Click here to experience it. If you want to learn more, read the AWS blog or visit our documentation.
Expand all
Row
Text
Filter events
all
2020-04-20 (08:22:04)
Time (UTC +00:00)
Message
2020-04-21
No older events found at the moment. Retry.
08:22:04
START RequestId: 26720d3a-9601-4b57-ac8a-ad18fd674572 Version: $LATEST
08:22:04
END RequestId: 26720d3a-9601-4b57-ac8a-ad18fd674572
08:22:04
REPORT RequestId: 26720d3a-9601-4b57-ac8a-ad18fd674572 Duration: 1.53 ms Billed Duration
No newer events found at the moment. Retry.