

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.

User name	Groups	Access key age	Password age	Last activity	MFA
awsstudent	None	Today	Today	None	Not enabled
root-qwkl		None	None		
user-1	None	None	Today	None	Not enabled
user-2	None	None	Today	None	Not enabled
user-3	None	None	Today	None	Not enabled

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.

The screenshot shows the AWS IAM User Summary page for a user named 'user-1'. The top navigation bar includes 'Resource Groups', a user icon, 'awsstudent @ 3502-9196-2203', 'Global', and 'Support'. The main content area has a 'Summary' tab selected, showing the User ARN (arn:aws:iam::350291962203:user/spl66/user-1), Path (/spl66/), and Creation time (2020-02-24 22:09 UTC+0530). Below this, there are tabs for 'Permissions', 'Groups', 'Tags (2)', 'Security credentials', and 'Access Advisor'. The 'Permissions' tab is active, displaying a message: 'Get started with permissions' (This user doesn't have any permissions yet. Get started by adding the user to a group, copying permissions from another user, or attaching a policy directly. Learn more) with 'Add permissions' and 'Add inline policy' buttons. A note below says 'Permissions boundary (not set)'. There are 'Delete user' and help buttons at the top right.

Click the Groups tab.

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

The screenshot shows the AWS IAM User Summary page for 'user-1', with the 'Groups' tab selected. The top navigation bar and summary details are identical to the previous screenshot. The 'Groups' tab is highlighted. Below it, there is a 'Add user to groups' button and a table with columns for 'Group name' and 'Attached permissions'. The table displays 'No results'.

Click the Security credentials tab.

user-1 is assigned a Console password

User ARN: arn:aws:iam::350291962203:user/spl66/user-1

Path: /spl66/

Creation time: 2020-02-24 22:09 UTC+0530

Sign-in credentials:

- Summary: Console sign-in link: https://350291962203.signin.aws.amazon.com/console

Console password: Enabled (never signed in) | Manage

Assigned MFA device: Not assigned | Manage

Signing certificates: None

Access keys:

No results

In the navigation pane on the left, click Groups.

The following groups have already been created for you:

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

Group Name	Users	Inline Policy	Creation Time
EC2-Admin	0	✓	2020-02-24 22:09 UTC+0530
EC2-Support	0		2020-02-24 22:09 UTC+0530
S3-Support	0		2020-02-24 22:09 UTC+0530

Click the EC2-Support group.

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

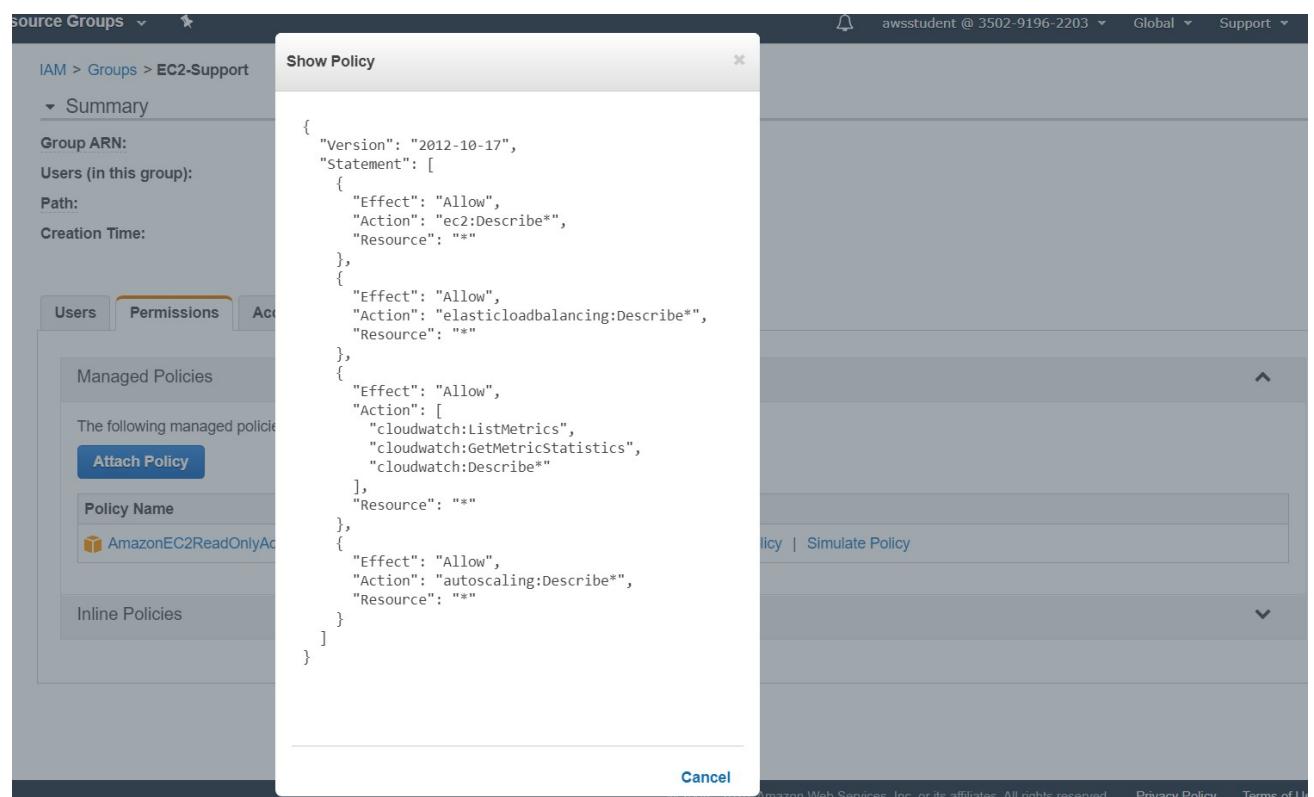
Click the Permissions tab.

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.

The screenshot shows the AWS IAM Groups page. The top navigation bar includes 'Resource Groups' and other global settings. The main navigation on the left is 'IAM > Groups > S3-Support'. The 'Summary' tab is selected, displaying the Group ARN (arn:aws:iam::350291962203:group/spl66/S3-Support), the number of users (1), the path (/spl66/), and the creation time (2020-02-24 22:09 UTC+0530). Below this, the 'Users' tab is selected, showing a table with one user: 'user-1'. There are 'Actions' buttons for each user: 'Remove User from Group' and 'Add Users to Group'. The status message at the bottom indicates there is 1 user in 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.

The screenshot shows the AWS IAM Groups page for the 'EC2-Support' group. The top navigation bar includes 'Resource Groups', a user icon, 'awsstudent @ 3502-9196-2203', 'Global', and 'Support'. The left sidebar has a 'Groups' section with 'EC2-Support' selected. The main content area shows the group's summary: Group ARN: arn:aws:iam::350291962203:group/spl66/EC2-Support, Users (in this group): 1, Path: /spl66/, Creation Time: 2020-02-24 22:09 UTC+0530. Below this, there are tabs for 'Users' (selected), 'Permissions', and 'Access Advisor'. A message states 'This view shows all users in this group: 1 User'. A table lists one user: 'user-2' with an 'Actions' column containing a 'Remove User from Group' link. Buttons for 'Remove Users from Group' and 'Add Users to Group' are at the bottom right.

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.

The screenshot shows the AWS IAM Groups page for the 'EC2-Admin' group. The top navigation bar and sidebar are identical to the previous screenshot. The main content area shows the group's summary: Group ARN: arn:aws:iam::350291962203:group/spl66/EC2-Admin, Users (in this group): 1, Path: /spl66/, Creation Time: 2020-02-24 22:09 UTC+0530. Below this, there are tabs for 'Users' (selected), 'Permissions', and 'Access Advisor'. A message states 'This view shows all users in this group: 1 User'. A table lists one user: 'user-3' with an 'Actions' column containing a 'Remove User from Group' link. Buttons for 'Remove Users from Group' and 'Add Users to Group' are at the bottom right.

In the navigation pane on the left, click Groups.

The screenshot shows the AWS IAM Groups list page. The top navigation bar includes 'Resource Groups', a user icon, 'awsstudent @ 3502-9196-2203', 'Global', and 'Support'. The left sidebar has a 'Groups' section with 'Create New Group' and 'Group Actions' dropdown. The main content area shows a table of groups: 'EC2-Admin' (1 user, inline policy checked, created 2020-02-24 22:09 UTC+0530), 'EC2-Support' (1 user, inline policy checked, created 2020-02-24 22:09 UTC+0530), and 'S3-Support' (1 user, inline policy checked, created 2020-02-24 22:09 UTC+0530). A search bar at the top right says 'Showing 3 results'.

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.

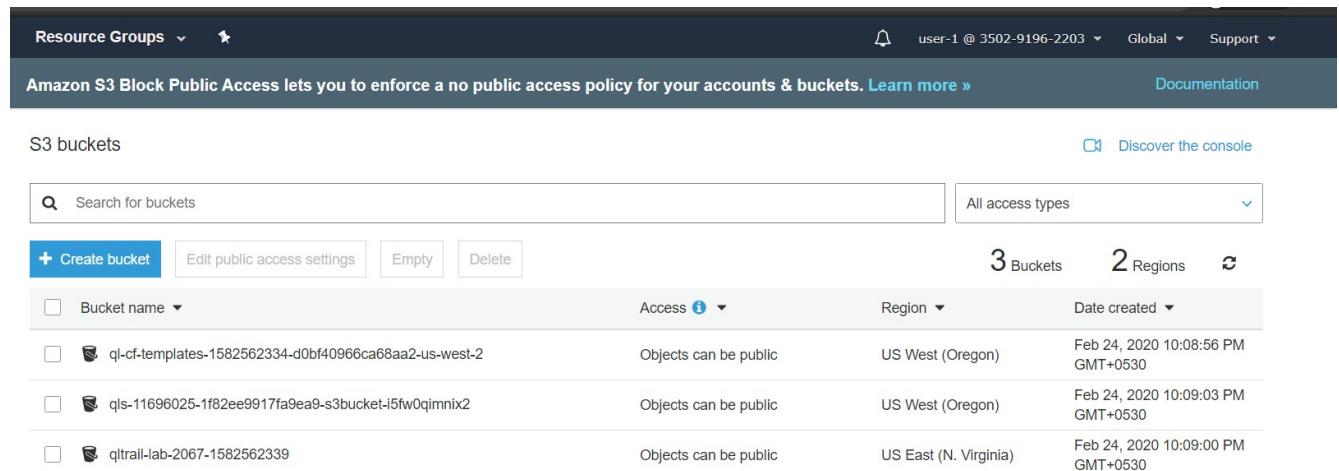
You will now sign-in as user-1, who has been hired as your Amazon S3 storage support staff.

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.



The screenshot shows the AWS S3 buckets page. At the top, there's a banner about Amazon S3 Block Public Access. Below it, a search bar and filter options are visible. A button to 'Create bucket' is highlighted in blue. The main table lists three buckets:

Bucket name	Access	Region	Date created
ql-cf-templates-1582562334-d0bf40966ca68aa2-us-west-2	Objects can be public	US West (Oregon)	Feb 24, 2020 10:08:56 PM GMT+0530
qls-11696025-1f82ee9917fa9ea9-s3bucket-i5fw0qimnix2	Objects can be public	US West (Oregon)	Feb 24, 2020 10:09:03 PM GMT+0530
qltrail-lab-2067-1582562339	Objects can be public	US East (N. Virginia)	Feb 24, 2020 10:09:00 PM GMT+0530

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.

A screenshot of the AWS Management Console showing the EC2 Instances page. The top navigation bar includes 'Resource Groups', a user icon, and 'Ohio'. Below the navigation is a toolbar with 'Launch Instance', 'Connect', and 'Actions' buttons. A search bar is present above the main table. The table has columns for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), IPv4 Public IP, and IPv6 Public IP. A message at the bottom of the table area reads: "An error occurred fetching instance data: You are not authorized to perform this operation."

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.

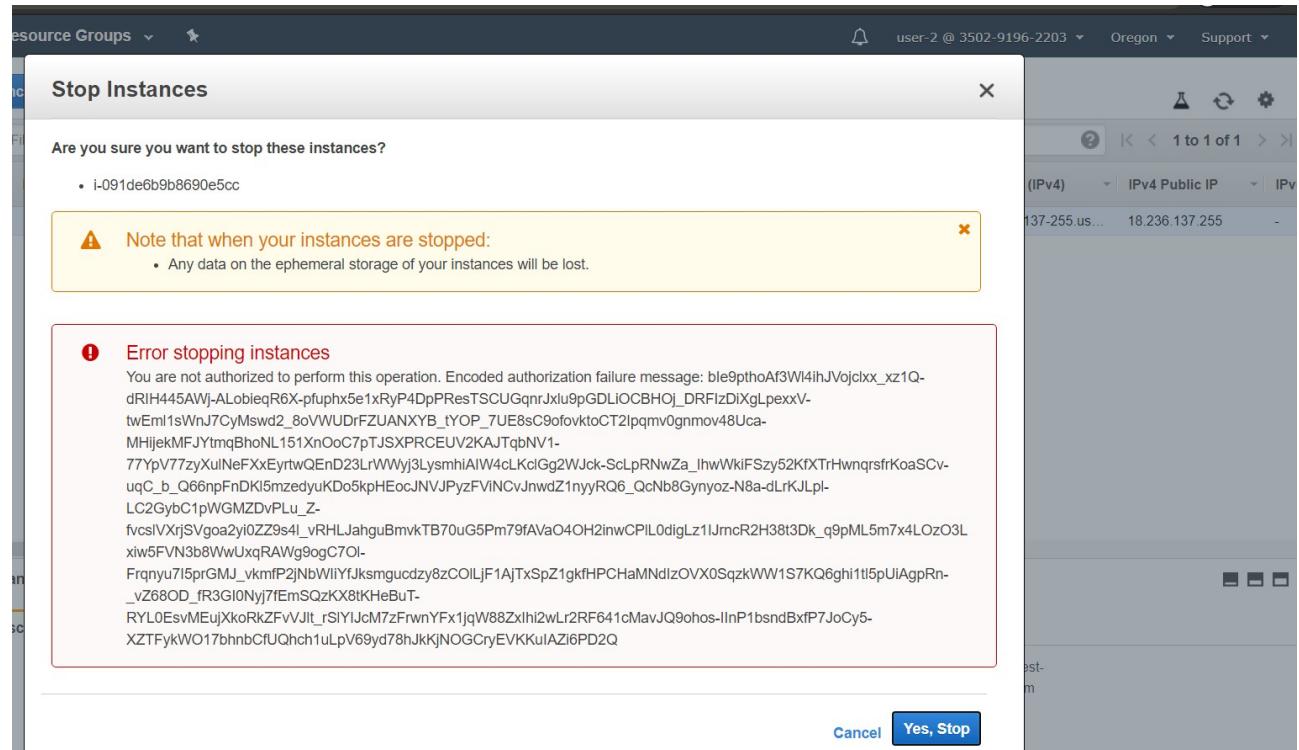
In the navigation pane on the left, click Instances.

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.



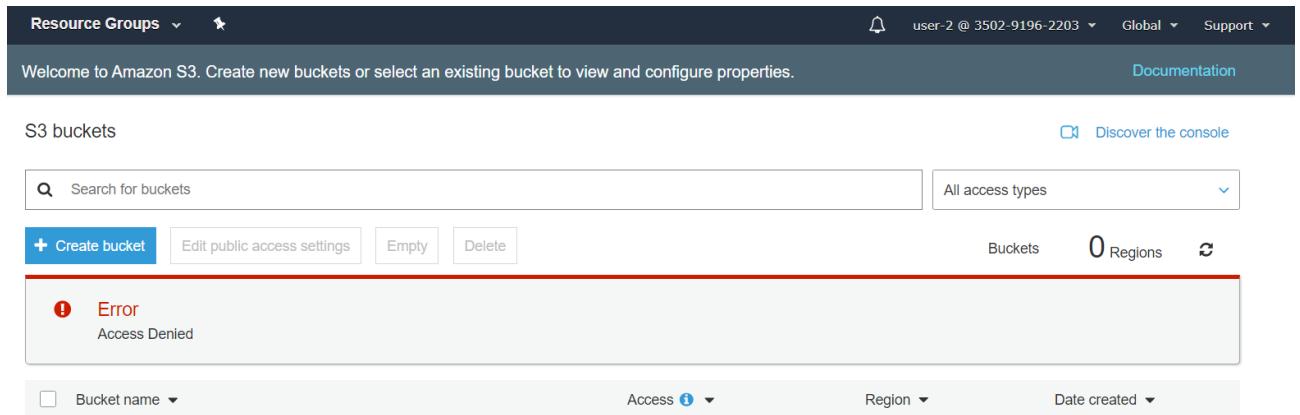
You will receive an error stating You are not authorized to perform this operation. This demonstrates that the policy only allows you to information, without making changes.

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.

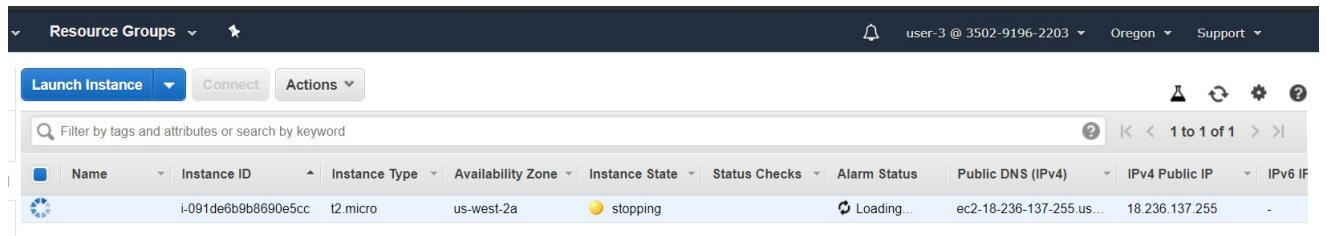
As an EC2 Administrator, you should now have permissions to Stop the Amazon EC2 instance.

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.

The screenshot shows the AWS S3 console with the following details:

- Header:** Services, Resource Groups, awsstudent @ 2013-0249-5971, Global, Support.
- Banner:** Prevent S3 objects from being deleted for a predefined retention period with S3 Object Lock. Learn more » Documentation
- Section:** S3 buckets
- Search Bar:** Search for buckets
- Filter:** All access types
- Actions:** + Create bucket, Edit public access settings, Empty, Delete
- Summary:** 3 Buckets, 1 Regions
- Table:** Lists three buckets with columns: Bucket name, Access, Region, Date created.
 - my-source1006: Bucket and objects not public, US East (N. Virginia), Feb 24, 2020 11:08:00 PM GMT+0530
 - ql-cf-templates-1582565602-d7f9e78792dcf28c-us-east-1: Objects can be public, US East (N. Virginia), Feb 24, 2020 11:03:24 PM GMT+0530
 - qltrail-lab-2274-1582565604: Objects can be public, US East (N. Virginia), Feb 24, 2020 11:03:25 PM GMT+0530

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

The screenshot shows the AWS S3 Properties page for the bucket "my-source1006".

- Header:** AWS, Services, Resource Groups, awsstudent @ 2013-0249-5971, Global, Support.
- Breadcrumbs:** Amazon S3 > my-source1006
- Section:** my-source1006
- Tabs:** Overview (selected), Properties, Permissions, Management, Access points
- Card:** Versioning
 - Description: Keep multiple versions of an object in the same bucket.
 - Status: Enabled
 - Link: Learn more
- Card:** Server access logging
 - Description: Set up access log records that provide details about access requests.
 - Status: Disabled
 - Link: Learn more
- Card:** Static website hosting
 - Description: Host a static website, which does not require server-side technologies.
 - Status: Disabled
 - Link: Learn more
- Card:** Object-level logging
 - Description: Record object-level API activity using the CloudTrail data events feature (additional cost).
 - Status: Disabled
 - Link: Learn more

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.

The screenshot shows the AWS S3 Bucket Properties page for the bucket 'my-destination1006'. The 'Properties' tab is active. In the 'Versioning' section, there is a note: 'Keep multiple versions of an object in the same bucket.' Below this is a 'Learn more' link and a status indicator: 'Enabled' with a checked checkbox. In the 'Server access logging' section, there is a note: 'Set up access log records that provide details about access requests.' Below this is a 'Learn more' link and a status indicator: 'Disabled' with an unchecked checkbox. In the 'Static website hosting' section, there is a note: 'Host a static website, which does not require server-side technologies.' Below this is a 'Learn more' link and a status indicator: 'Disabled' with an unchecked checkbox. In the 'Object-level logging' section, there is a note: 'Record object-level API activity using the CloudTrail data events feature (additional cost).' Below this is a 'Learn more' link and a status indicator: 'Disabled' with an unchecked checkbox.

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 tag. 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.

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and user information 'awsstudent @ 2013-0249-5971'. Below the navigation is the breadcrumb path 'Amazon S3 > my-source1006'. The main area displays the 'my-source1006' bucket. The 'Access points' tab is active. A search bar at the top says 'Type a prefix and press Enter to search. Press ESC to clear.' Below it, a toolbar includes 'Upload', 'Create folder', 'Download', 'Actions' (with dropdown), 'Versions', 'Hide', and 'Show'. The region is set to 'US East (N. Virginia)'. A table lists one object: 'pre-crr.txt'. The table has columns for Name (with a dropdown arrow), Last modified, Size (0 B), and Storage class (Standard). The table footer says 'Viewing 1 to 1'.

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

my-source1006

Overview Properties Permissions Management Access points

Lifecycle Replication Analytics Metrics Inventory

✓ Replication configuration updated successfully.

Source	Destination	Permissions
Bucket my-source1006	Bucket my-destination1006	IAM role S3-CRR-Role
Region US East (N. Virginia)	Region US West (Oregon)	Bucket policy Copy

Edit global settings

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.

Amazon S3 > my-destination1006

my-destination1006

Overview Properties Permissions Management Access points

Upload Create folder Download Actions Versions Hide Show

This bucket is empty. Upload new objects to get started.

US West (Oregon)

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.

my-source1006

Overview Properties Permissions Management Access points

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions Versions Hide Show US East (N. Virginia)

Name	Last modified	Size	Storage class
crr-bucket.txt	Feb 24, 2020 11:19:42 PM GMT+0530	0 B	Standard
pre-crr.txt	Feb 24, 2020 11:16:14 PM GMT+0530	0 B	Standard

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.

my-destination1006

Overview Properties Permissions Management Access points

Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions Versions Hide Show US West (Oregon)

Name	Last modified	Size	Storage class
crr-bucket.txt	Feb 24, 2020 11:19:42 PM GMT+0530	0 B	Standard

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

The screenshot shows the AWS S3 Replication configuration page. At the top, there are tabs for Overview, Properties, Permissions, Management, and Access points. The Management tab is selected. Below the tabs are buttons for Lifecycle, Replication (which is highlighted in blue), Analytics, Metrics, and Inventory. A success message box displays: "Replication configuration updated successfully." The main content area shows a table for defining replication rules. The table has columns for Source, Destination, and Permissions. The Source row shows: Bucket my-source1006, Region US East (N. Virginia). The Destination row shows: Bucket my-destination1006, Region US West (Oregon). The Permissions row shows: IAM role S3-CRR-Role, Bucket policy Copy. There is also a link to "Edit global settings". Below the table are buttons for "+ Add rule", "Edit priorities", "Edit", "Delete", and "Actions". A table below shows two existing replication rules:

Viewing 1 to 2 of 2							
Rule name	Scope	Storage class	Replicated object owner	KMS-encrypted objects	Status	Replication time control	Priority
crr-folder-only	prefix: crr-test/	Same as source	Same as source bucket	Do not replicate	Enabled	Disabled	2
crr-full-bucket	Entire bucket	Same as source	Same as source bucket	Do not replicate	Disabled	Disabled	1

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

The screenshot shows the AWS S3 bucket overview page for my-destination1006. The top navigation bar includes links for Services, Resource Groups, and a user profile. The main content area shows the bucket name "my-destination1006". Below the bucket name are tabs for Overview (selected) and Actions. A search bar contains the placeholder "Type a prefix and press Enter to search. Press ESC to clear.". Below the search bar are buttons for Upload, Create folder, Download, Actions, Versions, Hide, and Show. The location bar indicates the path: Amazon S3 > my-destination1006 > crr-test. On the right, it shows the region "US West (Oregon)". A table lists the contents of the crr-test folder:

Name	Last modified	Size	Storage class
crr-folder.txt	Feb 24, 2020 11:26:04 PM GMT+0530	0 B	Standard

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

The screenshot shows the AWS Replication configuration interface. At the top, there are tabs for Lifecycle, Replication (which is selected), Analytics, Metrics, and Inventory. A success message 'Replication configuration updated successfully.' is displayed. Below this, a table shows the configuration for a rule named 'crr-tag-only'. The table has columns for Source (Bucket: my-source1006, Region: US East (N. Virginia)), Destination (Bucket: my-destination1006, Region: US West (Oregon)), and Permissions (IAM role: S3-CRR-Role, Bucket policy: Copy). There is also a link to 'Edit global settings'. Below the table, a list of replication rules is shown:

Rule name	Scope	Storage class	Replicated object owner	KMS-encrypted objects	Status	Replication time control	Priority
crr-tag-only	tag: replicate yes	Same as source	Same as source bucket	Do not replicate	Enabled	Disabled	3
crr-folder-only	prefix: crr-test/	Same as source	Same as source bucket	Do not replicate	Enabled	Disabled	2
crr-full-bucket	Entire bucket	Same as source	Same as source bucket	Do not replicate	Disabled	Disabled	1

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.

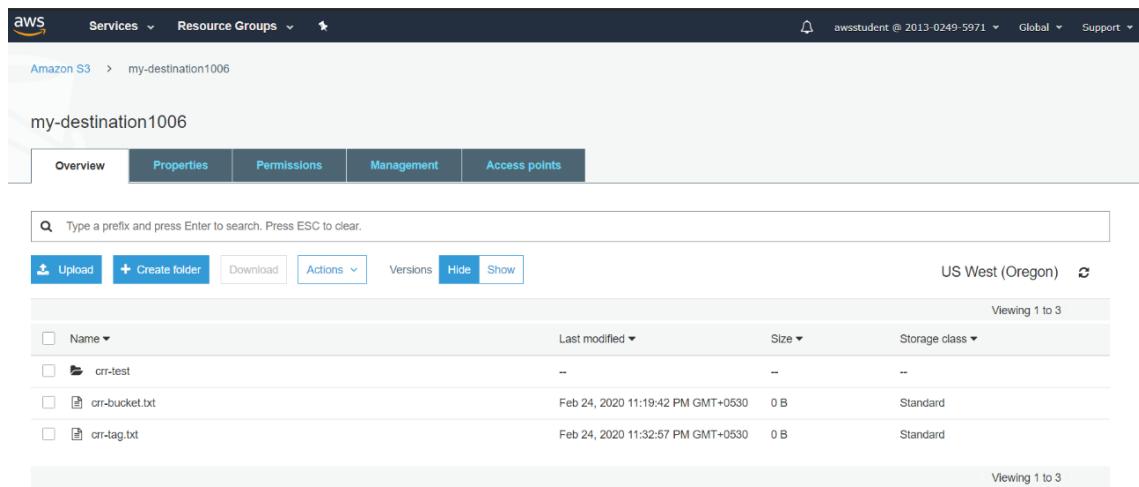
Click Save and then click Next

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.



Name	Last modified	Size	Storage class
crr-test	--	--	--
crr-bucket.txt	Feb 24, 2020 11:19:42 PM GMT+0530	0 B	Standard
crr-tag.txt	Feb 24, 2020 11:32:57 PM GMT+0530	0 B	Standard

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.

The screenshot shows the AWS S3 console with the 'my-source1006' bucket selected. The 'Actions' dropdown is open, and the 'Delete' option is highlighted. The table below lists three files: 'crr-test', 'crr-bucket.txt', and 'pre-crr.txt'. The 'crr-tag.txt' file is the one being deleted.

Name	Last modified	Size	Storage class
crr-test	--	--	--
crr-bucket.txt	Feb 24, 2020 11:19:42 PM GMT+0530	0 B	Standard
pre-crr.txt	Feb 24, 2020 11:16:14 PM GMT+0530	0 B	Standard

Navigate to your destination bucket.

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

The screenshot shows the AWS S3 console with the 'my-destination1006' bucket selected. The table below lists four files: 'crr-test', 'crr-bucket.txt', 'crr-tag.txt', and 'pre-crr.txt'. The 'crr-tag.txt' file is present in the list.

Name	Last modified	Size	Storage class
crr-test	--	--	--
crr-bucket.txt	Feb 24, 2020 11:19:42 PM GMT+0530	0 B	Standard
crr-tag.txt	Feb 24, 2020 11:32:57 PM GMT+0530	0 B	Standard
pre-crr.txt	Feb 24, 2020 11:16:14 PM GMT+0530	0 B	Standard

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

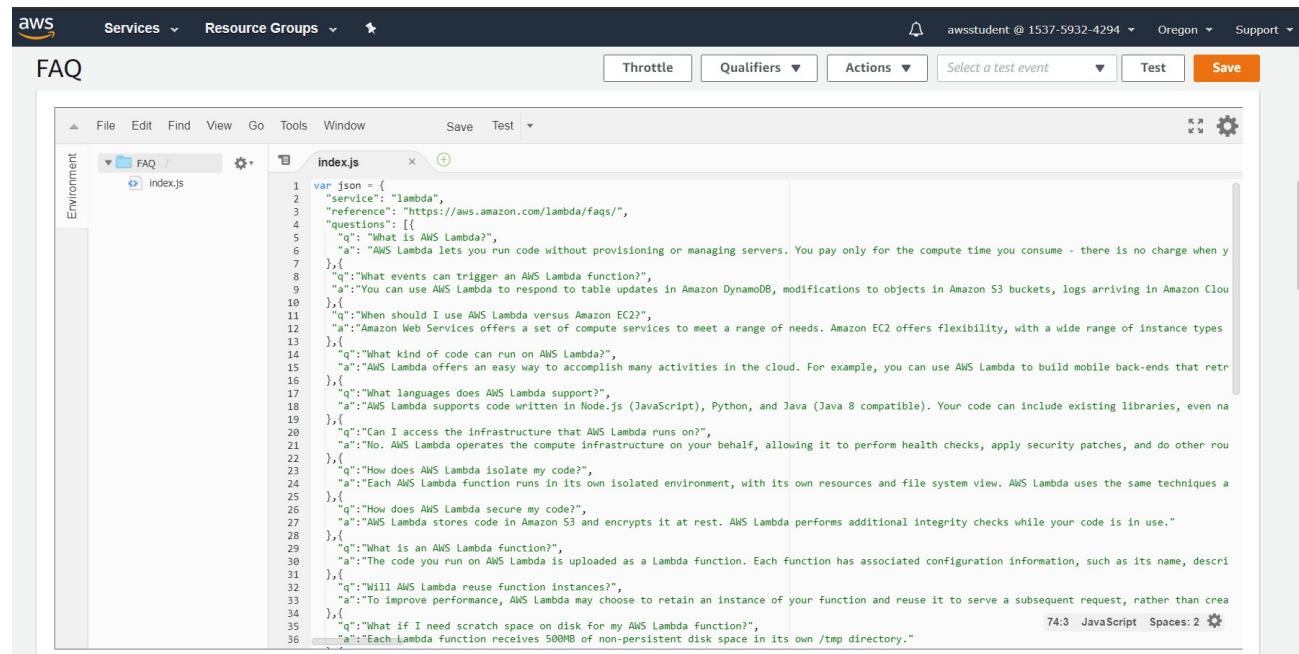
Click Create function

The screenshot shows the 'Basic information' section of the AWS Lambda 'Create function' wizard. The 'Function name' field contains 'FAQ'. The 'Runtime' dropdown is set to 'Node.js 12.x'. Under the 'Permissions' section, the 'Choose or create an execution role' dropdown is expanded, showing the 'lambda-basic-execution' role selected. At the bottom right, there are 'Cancel' and 'Create function' buttons.

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.



```
var json = {
  "ref": "Lambda",
  "reference": "https://aws.amazon.com/lambda/faqs/",
  "questions": [
    {"q": "What is AWS Lambda?", "a": "AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code isn't running."},
    {"q": "What events can trigger an AWS Lambda function?", "a": "You can use AWS Lambda to respond to table updates in Amazon DynamoDB, modifications to objects in Amazon S3 buckets, logs arriving in Amazon CloudWatch Logs, and more."},
    {"q": "When should I use AWS Lambda versus Amazon EC2?", "a": "Amazon Web Services offers a set of compute services to meet a range of needs. Amazon EC2 offers flexibility, with a wide range of instance types and configurations, while AWS Lambda is designed for serverless computing where you don't manage infrastructure."},
    {"q": "What kind of code can run on AWS Lambda?", "a": "AWS Lambda supports code written in Node.js (JavaScript), Python, and Java (Java 8 compatible). Your code can include existing libraries, even native ones."},
    {"q": "Can I access the infrastructure that AWS Lambda runs on?", "a": "No. AWS Lambda operates the compute infrastructure on your behalf, allowing it to perform health checks, apply security patches, and do other routine maintenance."},
    {"q": "How does AWS Lambda isolate my code?", "a": "Each AWS Lambda function runs in its own isolated environment, with its own resources and file system view. AWS Lambda uses the same techniques as Amazon VPC to provide network isolation."},
    {"q": "How does AWS Lambda secure my code?", "a": "AWS Lambda stores code in Amazon S3 and encrypts it at rest. AWS Lambda performs additional integrity checks while your code is in use."},
    {"q": "What is an AWS Lambda Function?", "a": "The code you run on AWS Lambda is uploaded as a Lambda function. Each function has associated configuration information, such as its name, description, and runtime environment."},
    {"q": "Will AWS Lambda reuse function instances?", "a": "To improve performance, AWS Lambda may choose to retain an instance of your function and reuse it to serve a subsequent request, rather than creating a new one."},
    {"q": "What if I need scratch space on disk for my AWS Lambda function?", "a": "Each Lambda function receives 500MB of non-persistent disk space in its own /tmp directory."}
]
```

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

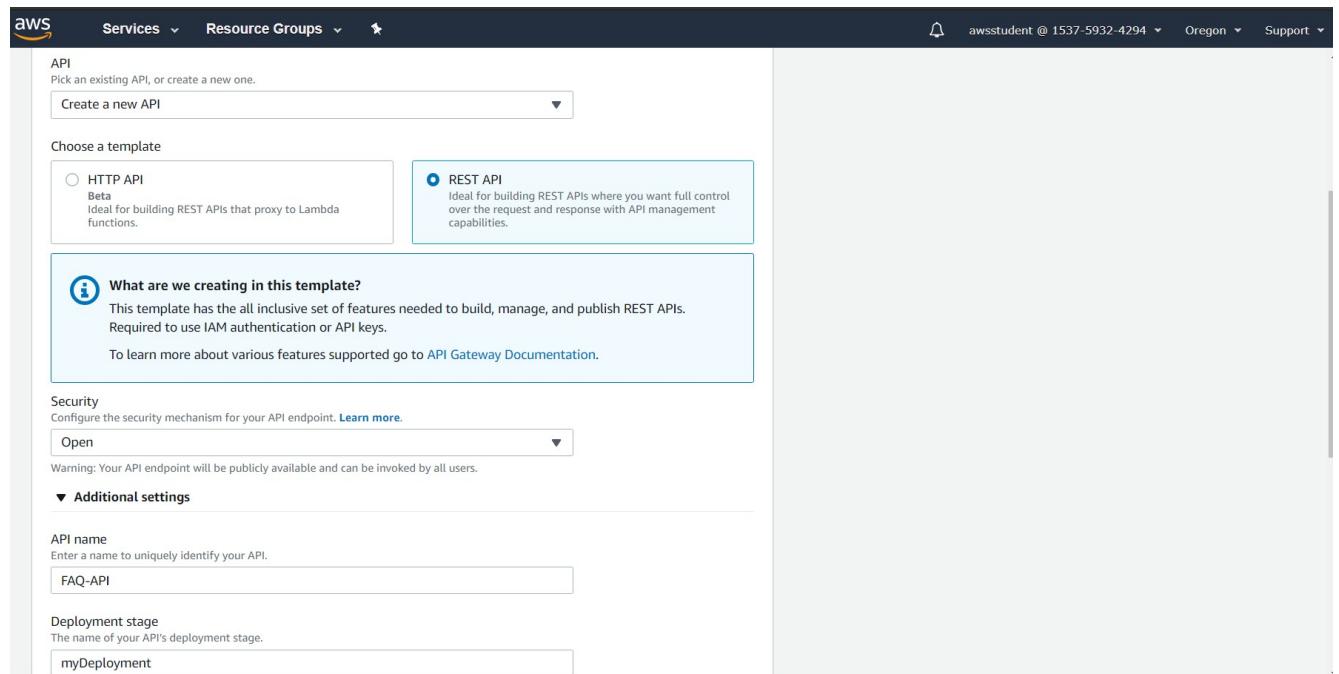
Security: Open

Expand Additional settings

API name: FAQ-API

Deployment stage: myDeployment

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



Pick an existing API, or create a new one.

Create a new API

Choose a template

HTTP API
Beta
Ideal for building REST APIs that proxy to Lambda functions.

REST API
Ideal for building REST APIs where you want full control over the request and response with API management capabilities.

i What are we creating in this template?
This template has the all inclusive set of features needed to build, manage, and publish REST APIs.
Required to use IAM authentication or API keys.
To learn more about various features supported go to [API Gateway Documentation](#).

Security
Configure the security mechanism for your API endpoint. [Learn more](#).

Open

Warning: Your API endpoint will be publicly available and can be invoked by all users.

▼ Additional settings

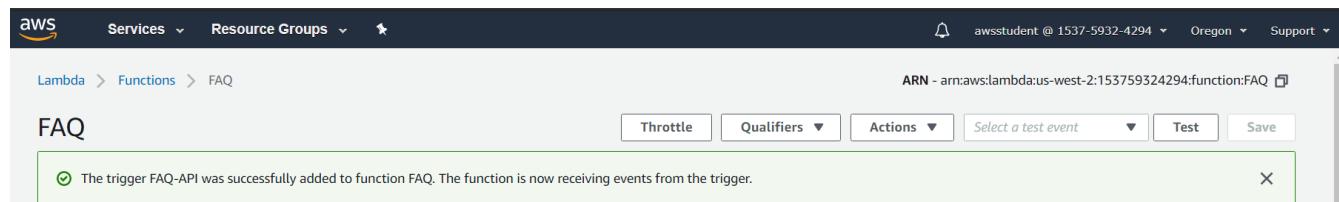
API name
Enter a name to uniquely identify your API.

FAQ-API

Deployment stage
The name of your API's deployment stage.

myDeployment

Click Save at the top right corner.



Lambda > Functions > FAQ

FAQ

ARN - arn:aws:lambda:us-west-2:153759324294:function:FAQ

The trigger FAQ-API was successfully added to function FAQ. The function is now receiving events from the trigger.

Throttle Qualifiers Actions Select a test event Test Save

Task 2: Test the Lambda function

On the FAQ Lambda function page,

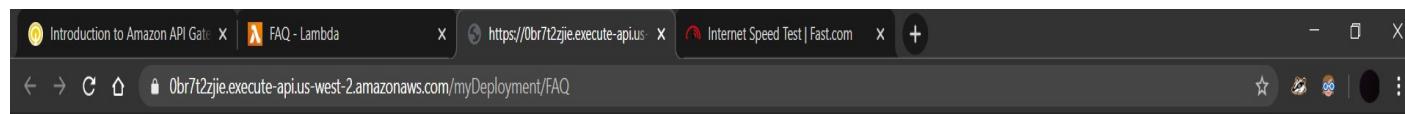
Under API Gateway, click the right arrow to view the details of the API.

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: {}

A screenshot of the AWS Lambda Test event configuration dialog. It shows a radio button for "Create new test event" selected, while "Edit saved test events" is unselected. Under "Event template", "Hello World" is chosen from a dropdown menu. In the "Event name" field, "BasicTest" is entered. Below the event name, there is a JSON editor area with a single line of code: "1 {}". At the bottom right of the dialog are "Cancel" and "Create" buttons, with "Create" being highlighted.

At the bottom of the screen, click Create

At the top of screen, Click Test

In the Execution result: succeeded window, expand Details.

The screenshot shows the AWS Lambda Functions interface. In the center, there's a card for the 'FAQ' function. At the top of the card, it says 'Execution result: succeeded (logs)'. Below this, there's a 'Details' link. The bottom of the card has tabs for 'Configuration', 'Permissions', and 'Monitoring', with 'Configuration' being the active tab.

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

The screenshot shows the AWS CloudWatch Logs Insights interface. On the left, there's a sidebar with various monitoring options like CloudWatch, Dashboards, Alarms, Metrics, Events, Rules, Event Buses, ServiceLens, Service Map, Traces, Synthetics, Canaries, Contributor Insights, and Settings. The main area shows a log stream for the 'CloudWatch > Log Groups > /aws/lambda/FAQ > 2020/02/25[\$LATEST]2212a16f1ede46c89d6839472690c2c0' log group. A modal window titled 'Try CloudWatch Logs Insights' provides information about the service. The log entries are listed in a table with columns for 'Time (UTC +00:00)' and 'Message'. The first few entries are:

Time (UTC +00:00)	Message
2020-02-25	No older events found at the moment. Retry.
04:44:13	START RequestId: 6f157851-0870-4dc2-a616-555d619f4f98 Version: \$LATEST
04:44:13	2020-02-25T04:44:13.300Z 6f157851-0870-4dc2-a616-555d619f4f98 INFO Quote selected: 8
04:44:13	2020-02-25T04:44:13.313Z 6f157851-0870-4dc2-a616-555d619f4f98 INFO { body: " ["q": "What is an AWS Lambda function?", "a": "The code you run on AWS Lambda is uploaded" }
04:44:13	END RequestId: 6f157851-0870-4dc2-a616-555d619f4f98
04:44:13	REPORT RequestId: 6f157851-0870-4dc2-a616-555d619f4f98 Duration: 79.65 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 70 MB Init Duration: 129.9
04:48:12	START RequestId: 288b9421-66d5-4ccf-9b20-95daa1ea66df Version: \$LATEST
04:48:12	2020-02-25T04:48:12.833Z 288b9421-66d5-4ccf-9b20-95daa1ea66df INFO Quote selected: 11
04:48:12	2020-02-25T04:48:12.833Z 288b9421-66d5-4ccf-9b20-95daa1ea66df INFO { body: " ["q": "Why must AWS Lambda functions be stateless?", "a": "Keeping functions stateless enables" }
04:48:12	END RequestId: 288b9421-66d5-4ccf-9b20-95daa1ea66df
04:48:12	REPORT RequestId: 288b9421-66d5-4ccf-9b20-95daa1ea66df Duration: 55.75 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 70 MB

Click on one of the log streams.

EXP 4: Introduction to AWS Lambda

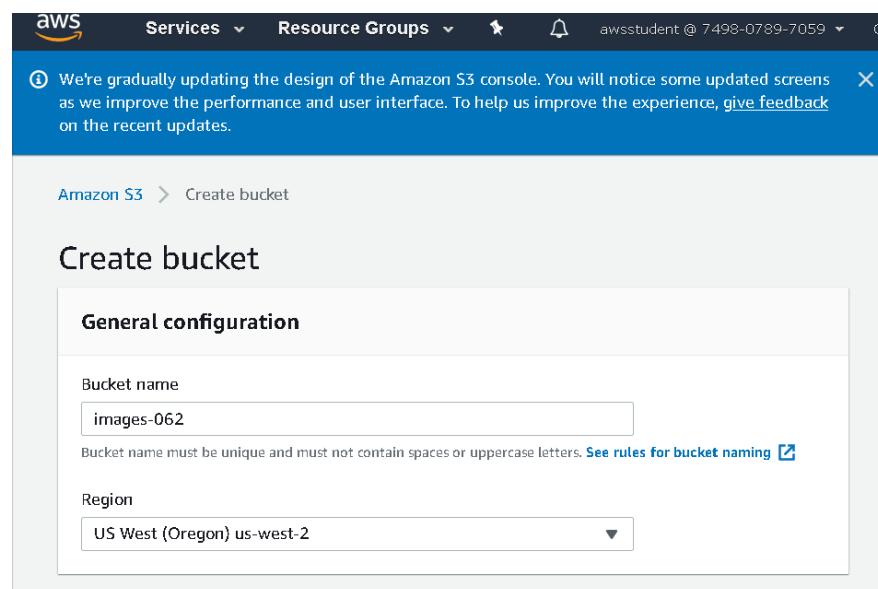
Task 1: Create the Amazon S3 Buckets

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.

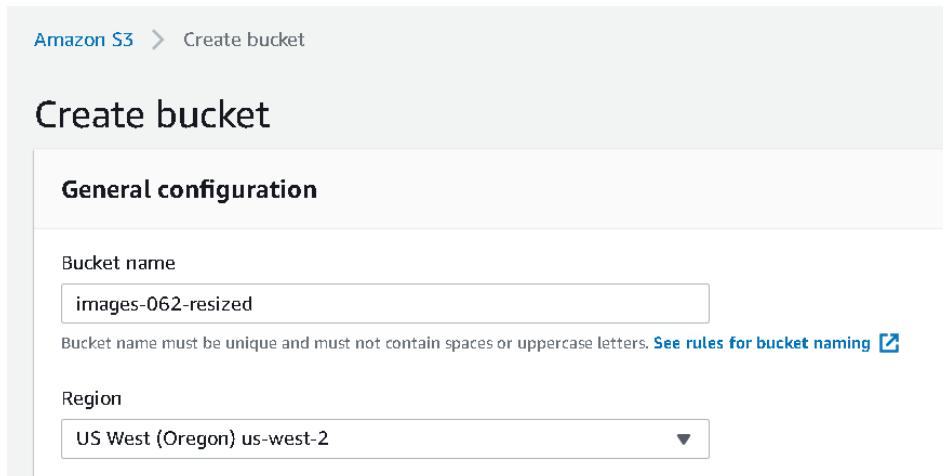
Click Create bucket and then configure:

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.

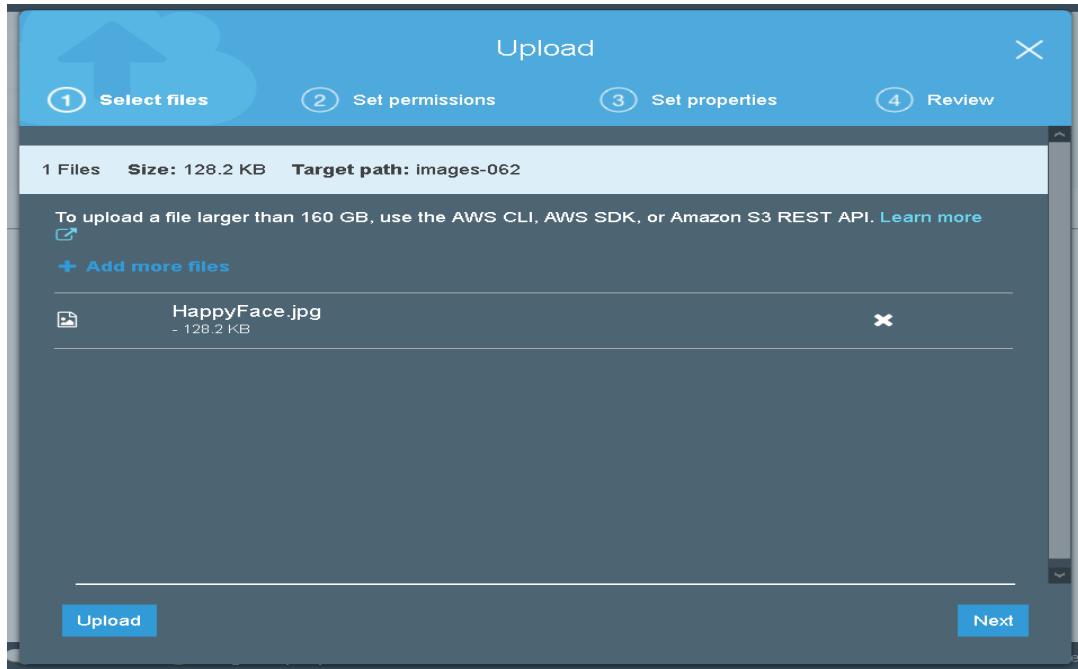
Open the image on the computer. It is a large picture, with dimensions of 1280 x 853.

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

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function.

Permissions [Info](#)
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
 [View the lambda-execution-role role on the IAM console.](#) 

[Cancel](#) [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:

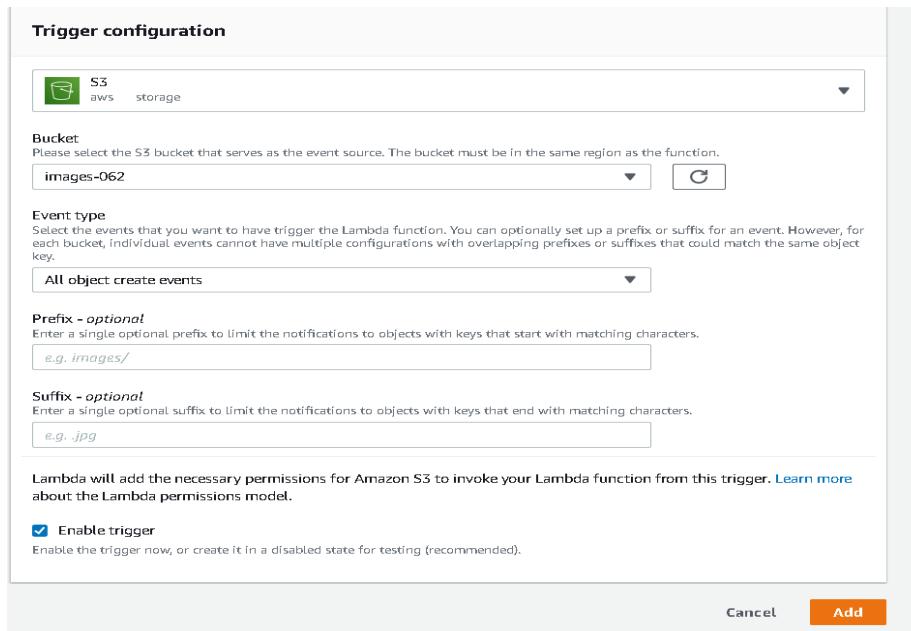
Select a trigger: S3

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.

Function code [Info](#)

Code entry type Upload a file from ...	Runtime Python 3.7	Handler Info CreateThumbnail.lambda
---	-----------------------	--

Amazon S3 link URL
Paste an S3 link URL to your function code .zip.

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

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:

Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

Create new test event
 Edit saved test events

Event template
[Amazon S3 Put](#)

Event name
[Upload](#)

```
8     "eventName": "ObjectCreated:Put",
9     "userIdentity": {
10       "principalId": "EXAMPLE"
11     },
12     "requestParameters": {
13       "sourceIPAddress": "127.0.0.1"
14     },
15     "responseElements": {
16       "x-amz-request-id": "EXAMPLE123456789",
17       "x-amz-ie-2": "EXAMPLE123/5678abcdefghijklmabdaisawesome/mnopqrstuvwxyzABCDEFGH"
18     },
19     "s3": {
20       "s3SchemaVersion": "1.0",
21       "configurationId": "testConfigRule",
22       "bucket": {
23         "name": "images-062",
24         "ownerIdentity": {
25           "principalId": "EXAMPLE"
26         },
27         "arn": "arn:aws:s3:::images-062"
28       },
29       "object": {
30         "key": "HappyFace.jpg",
31         "size": 1024,
32         "eTag": "0123456789abcdef0123456789abcdef",
33         "sequencer": "0A1B2C3D4E5F678901"
34       }
35     }
36   }
37 }
```

[Cancel](#) [Create](#)

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

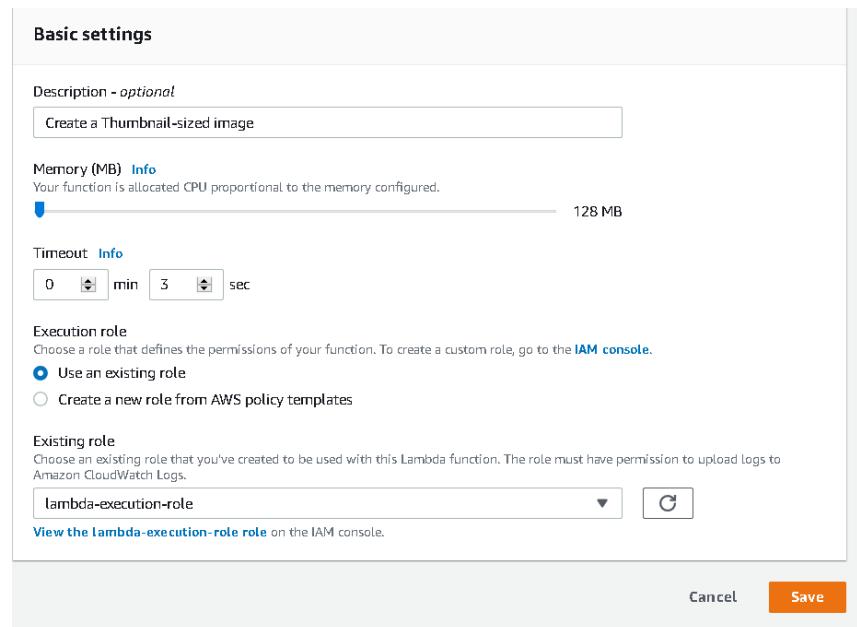
In the Basic settings section towards the bottom of the page, click Edit

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.

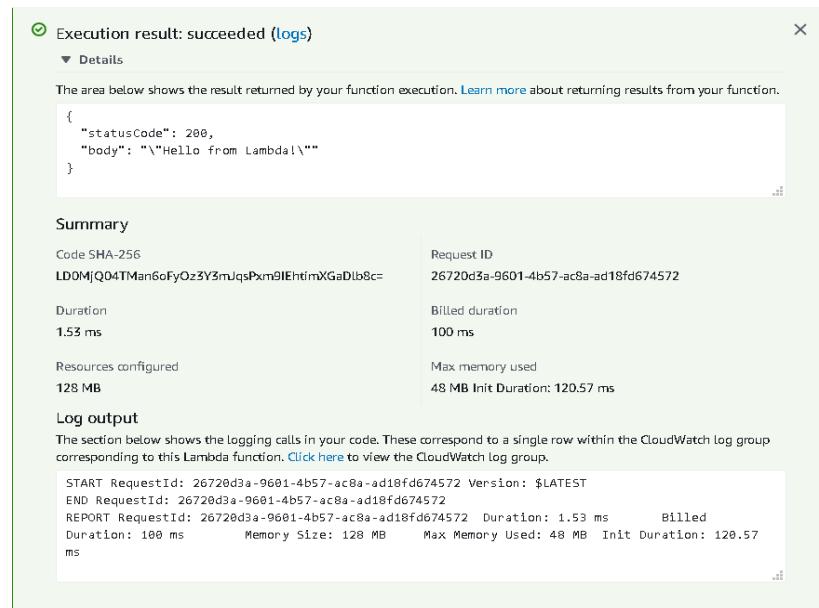
Task 3: Test Your Function

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



The screenshot shows the AWS Lambda Test configuration interface. At the top, there's a success message: "Execution result: succeeded (logs)". Below it, a "Details" section shows the response body: {"statusCode": 200, "body": "\"Hello from Lambda!\""}.

The "Summary" section provides performance metrics: Request ID (26720d3a-9601-4b57-ac8a-ad18fd674572), Duration (1.53 ms), and Resources configured (128 MB).

The "Log output" section displays CloudWatch logs corresponding to the Lambda function's execution:

```
START RequestId: 26720d3a-9601-4b57-ac8a-ad18fd674572 Version: $LATEST
END RequestId: 26720d3a-9601-4b57-ac8a-ad18fd674572
REPORT RequestId: 26720d3a-9601-4b57-ac8a-ad18fd674572 Duration: 1.53 ms Billed Duration: 1.53 ms Memory Size: 128 MB Max Memory Used: 48 MB Init Duration: 120.57 ms
```

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.

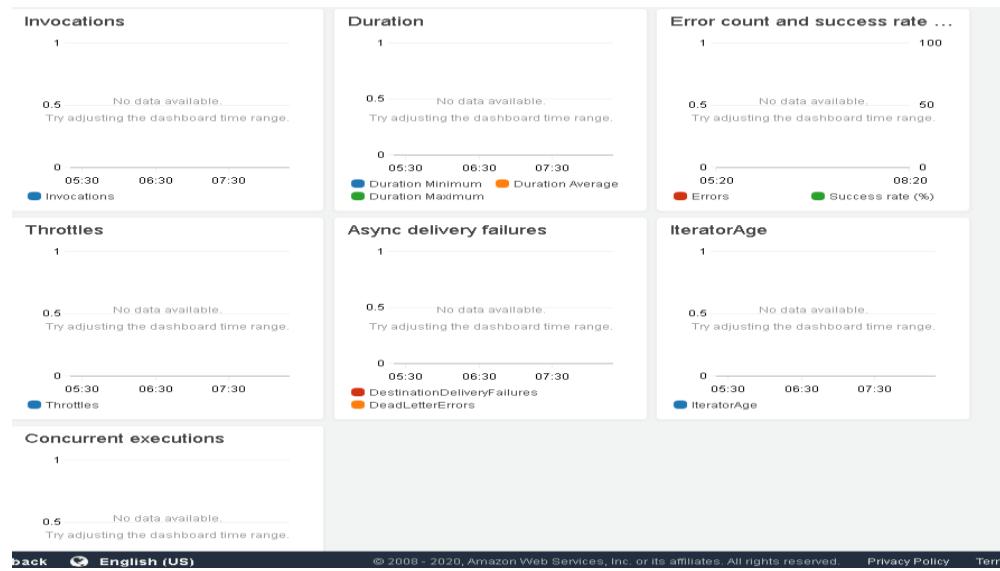
Task 4: Monitoring and Logging

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   

	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 .

