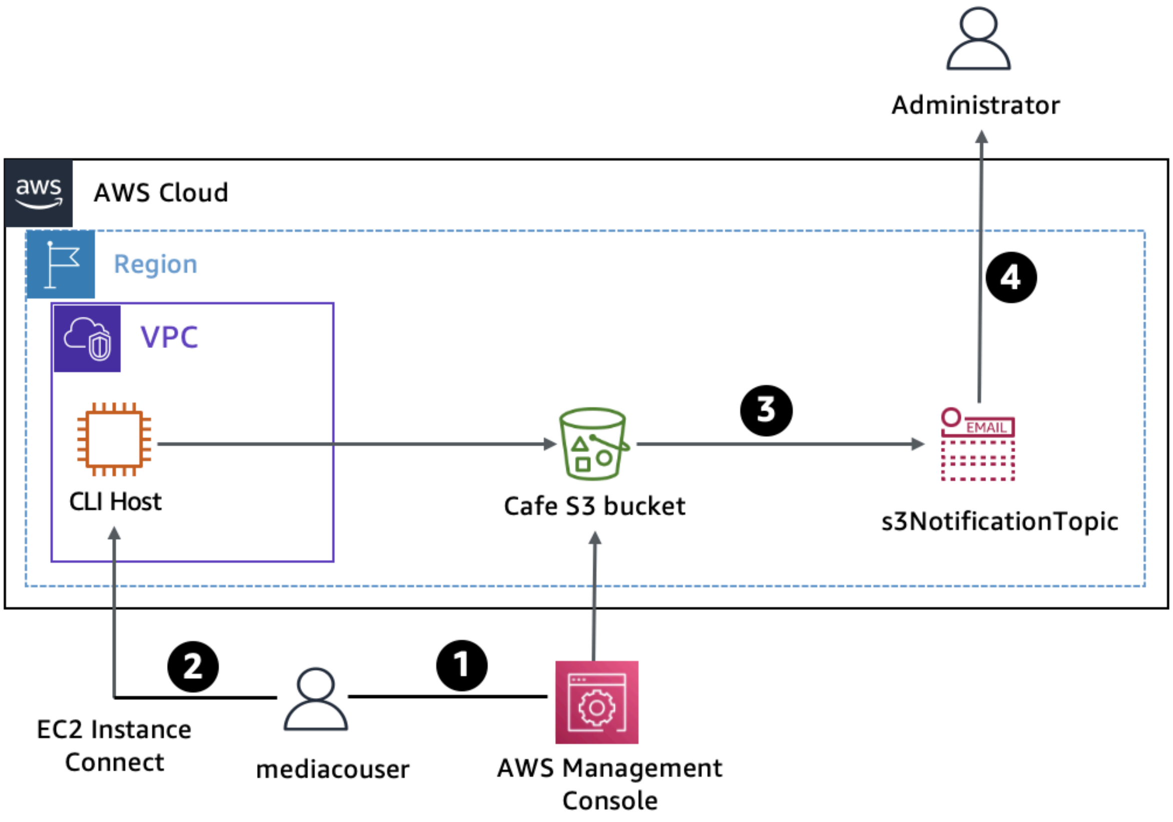
# working with Amazon S3W

# Lab overview

In this lab, you create and configure an Amazon Simple Storage Service (Amazon S3) bucket to share images with an external user at a media company (mediacouser) who has been hired to provide pictures of the products that the café sells. You also configure the S3 bucket to automatically send an email notification to the administrator when the bucket contents are modified.

The following diagram shows the component architecture of the Amazon S3 file-sharing solution and illustrates its usage flow.



An AWS Identity and Access Management (IAM) user named mediacouser, which represents an external user at a media company, has been pre-created with the appropriate Amazon S3 permissions to allow the user to add, change, or delete images from the bucket. The necessary Amazon S3 permissions are reviewed for each user to make sure that access to the bucket is secure and appropriate for each role.

The following steps describe the usage flow in the diagram:

1. When new product pictures are available or when existing pictures must be updated, a representative from the media company signs in to the AWS Management Console as **mediacouser** to upload, change, or delete the bucket contents.
2. As an alternative, **mediacouser** can use the AWS Command Line Interface (AWS CLI) to change the contents of the S3 bucket.
3. When Amazon S3 detects a change in the contents of the bucket, it publishes an email notification to the **s3NotificationTopic** Amazon Simple Notification Service (Amazon SNS) topic.
4. The administrator who is subscribed to the **s3NotificationTopic** SNS topic receives an email message that contains the details of the changes to the contents of the bucket.

**Note:** In real-world implementations, external users might not receive direct access to CLI Host as depicted in the diagram.

# Objectives

By the end of this lab, you will be able to do the following:

- Use the s3api and s3 AWS CLI commands to create and configure an S3 bucket.
- Verify write permissions to a user on an S3 bucket.
- Configure event notification on an S3 bucket.

# Task 1: Connecting to the CLI Host EC2 instance and configuring the AWS CLI

In this task, you connect to the CLI Host EC2 instance by using EC2 Instance Connect and configure the AWS CLI so that you can run commands.

## Task 1.1: Connecting to the CLI Host EC2 instance

In this task, you use EC2 Instance Connect to connect to the CLI Host EC2 instance.

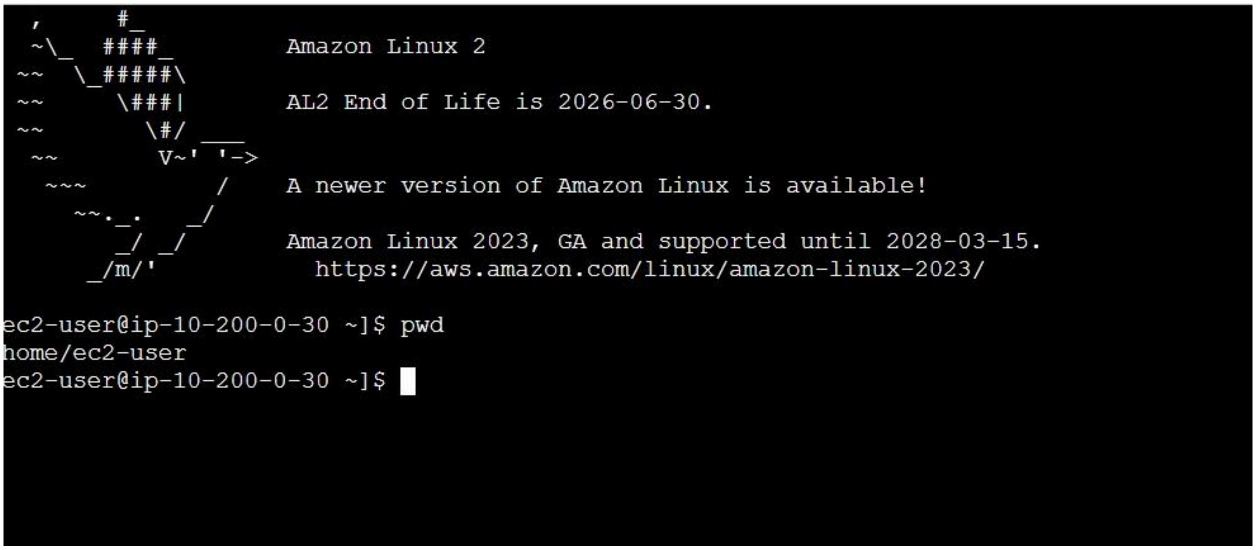
7. On the **AWS Management Console**, in the **Search** bar, enter and choose `EC2` to open the **EC2 Management Console**.
8. In the navigation pane, choose **Instances**.
9. From the list of instances, select the **CLI Host** instance.
10. Choose **Connect**.
11. On the **EC2 Instance Connect** tab, choose **Connect**.
    This option opens a new browser tab with the **EC2 Instance Connect** terminal window.
    You use this terminal window to complete the tasks throughout the lab. If the terminal becomes unresponsive, refresh the browser or use the steps in this task to connect again.

## Task 1.2: Configuring the AWS CLI on the CLI Host instance

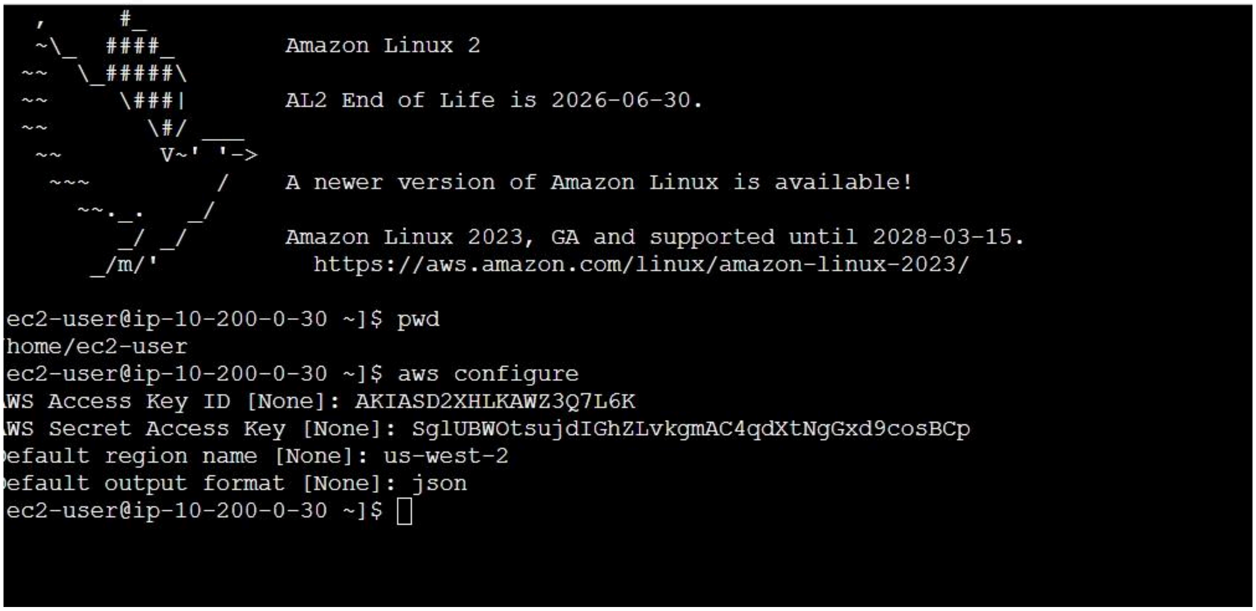
12. To set up the AWS CLI profile with credentials, run the following command in the EC2 Instance Connect terminal:
13. `aws configure`
14. At the prompts, copy the following values that you pasted into your text editor, and paste them into the terminal window as directed.
    - **AWS Access Key ID:** Enter the value for **AccessKey**.

- ○ **AWS Secret Access Key:** Enter the value for **SecretKey**.
- ○ **Default region name:** Enter `us-west-2`.
- ○ **Default output format:** Enter `json`.

15. You are ready to run AWS CLI commands to interact with AWS services.



```
 ,     #_
 ~\_  ####_        Amazon Linux 2
~~  \_#####\
~~     \###|       AL2 End of Life is 2026-06-30.
~~       \#/ ___
 ~~       V~' '->
  ~~~         /    A newer version of Amazon Linux is available!
    ~~._.   _/
       _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
     _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

ec2-user@ip-10-200-0-30 ~]$ pwd
home/ec2-user
ec2-user@ip-10-200-0-30 ~]$ █
```



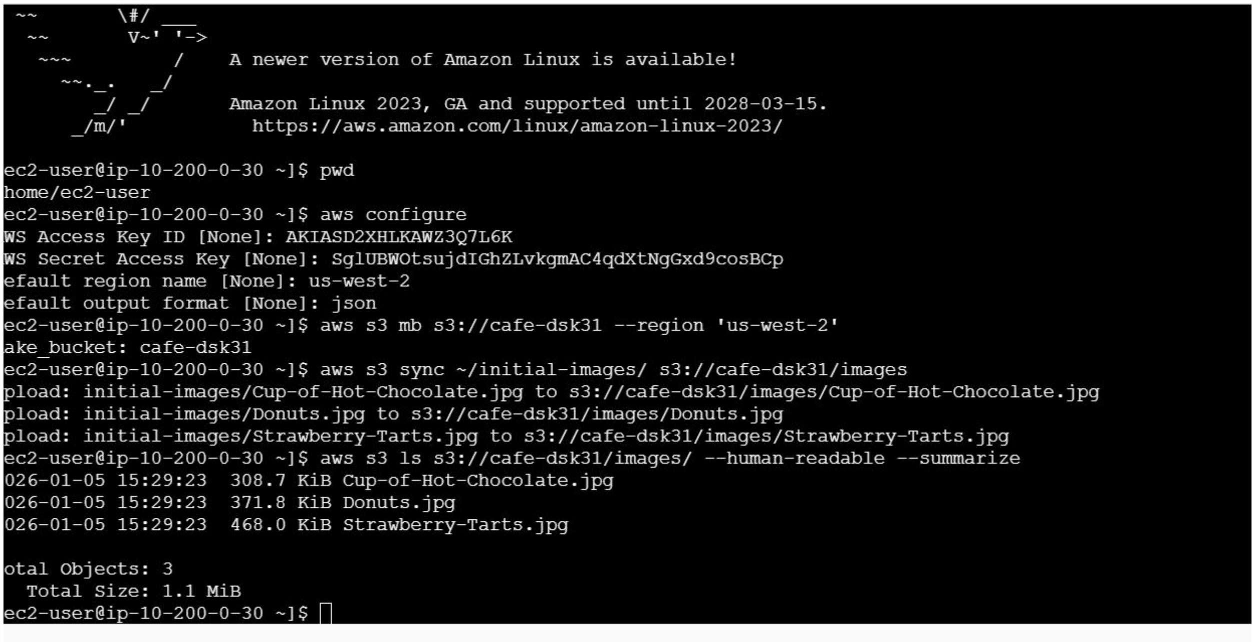
```
 ,     #_
 ~\_  ####_        Amazon Linux 2
~~  \_#####\
~~     \###|       AL2 End of Life is 2026-06-30.
~~       \#/ ___
 ~~       V~' '->
  ~~~         /    A newer version of Amazon Linux is available!
    ~~._.   _/
       _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
     _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

ec2-user@ip-10-200-0-30 ~]$ pwd
home/ec2-user
ec2-user@ip-10-200-0-30 ~]$ aws configure
WS Access Key ID [None]: AKIASD2XHLKAWZ3Q7L6K
WS Secret Access Key [None]: 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
efault region name [None]: us-west-2
efault output format [None]: json
ec2-user@ip-10-200-0-30 ~]$ □
```

# Task 2: Creating and initializing the S3 share bucket

In this task, you use the AWS CLI to create the S3 share bucket and upload a few images.

To do so, you run the following commands in the EC2 Instance Connect terminal window.

14. To create an S3 bucket, run the following command. In the command, replace *<cafe-xxxnnn>* with your bucket name. Your bucket name must begin with **cafe-** and should include a combination of letters and numbers to make your bucket name unique:
15. 
    ```
    aws s3 mb s3://<cafe-xxxnnn> --region 'us-west-2'
    ```
16. You should receive a message similar to the following: make_bucket: cafe-xxxx9999999  
    **Note:** Bucket names cannot contain uppercase letters. If you receive an error when you try to create your S3 bucket, make sure your bucket name doesn't include uppercase letters.  
    Next, you load some images into the S3 bucket under the /images prefix. Sample image files are provided in the initial-images folder on the CLI Host.
17. To load images into the bucket, run the following command. In the command, replace *<cafe-xxxnnn>* with your bucket name:
18. 
    ```
    aws s3 sync ~/initial-images/ s3://<cafe-xxxnnn>/images
    ```
19. The command output lists the image files that are being uploaded.
20. To verify that the files were synced to the S3 bucket, run the following command. In the command, replace *<cafe-xxxnnn>* with your bucket name:
21. 
    ```
    aws s3 ls s3://<cafe-xxxnnn>/images/ --human-readable --summarize
    ```
22. You see the details of the image files that were uploaded, including the number of files uploaded and the total size of the files.



```
 ~~       \#/ ___
  ~~       V~' '->
   ~~~         /    A newer version of Amazon Linux is available!
     ~~._.   _/
        _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
      _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

ec2-user@ip-10-200-0-30 ~]$ pwd
home/ec2-user
ec2-user@ip-10-200-0-30 ~]$ aws configure
WS Access Key ID [None]: AKIASD2XHLKAWZ3Q7L6K
WS Secret Access Key [None]: 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
efault region name [None]: us-west-2
efault output format [None]: json
ec2-user@ip-10-200-0-30 ~]$ aws s3 mb s3://cafe-dsk31 --region 'us-west-2'
ake_bucket: cafe-dsk31
ec2-user@ip-10-200-0-30 ~]$ aws s3 sync ~/initial-images/ s3://cafe-dsk31/images
pload: initial-images/Cup-of-Hot-Chocolate.jpg to s3://cafe-dsk31/images/Cup-of-Hot-Chocolate.jpg
pload: initial-images/Donuts.jpg to s3://cafe-dsk31/images/Donuts.jpg
pload: initial-images/Strawberry-Tarts.jpg to s3://cafe-dsk31/images/Strawberry-Tarts.jpg
ec2-user@ip-10-200-0-30 ~]$ aws s3 ls s3://cafe-dsk31/images/ --human-readable --summarize
026-01-05 15:29:23  308.7 KiB Cup-of-Hot-Chocolate.jpg
026-01-05 15:29:23  371.8 KiB Donuts.jpg
026-01-05 15:29:23  468.0 KiB Strawberry-Tarts.jpg

otal Objects: 3
  Total Size: 1.1 MiB
ec2-user@ip-10-200-0-30 ~]$ 
```

# Task 3: Reviewing the IAM group and user permissions

Next, you review the permissions assigned to the mediaco IAM user group. This group was created to provide a way for the users of the media company to use the AWS Management Console or the AWS CLI to upload and modify images in the S3 share bucket. Creating the group makes it convenient to manage individual user permissions. You also review the permissions inherited by the mediacouser user that is part of the group.

## Task 3.1: Reviewing the mediaco IAM group

In this section, you review the permissions assigned to the mediaco group.

17. On the **AWS Management Console**, in the **Search** bar, enter and choose `IAM` to open the **IAM Management Console**.
18. In the navigation pane on the left, choose **User groups**.
19. From the **User groups** list, select **mediaco**.
    The **Summary** page for the **mediaco** group is displayed.
20. Choose the **Permissions** tab.

21. Next to **IAMUserChangePassword**, choose **+** to expand the policy.
    If needed, review the AWS managed policy that permits users to change their own password.
22. To collapse the policy, choose **-**.
23. Next to **mediaCoPolicy**, choose **+** to expand the policy.
    **Note**: You might have to scroll down to see the policy.
    Notice the following statements in this policy:
    - The first statement, identified by the **Sid** key name **AllowGroupToSeeBucketListInTheConsole**, defines permissions that allow the user to use the Amazon S3 console to view the list of S3 buckets in the account.
    - The second statement, identified by the **Sid** key name **AllowRootLevelListingOfTheBucket**, defines permissions that allow the user to use the Amazon S3 console to view the list of first-level objects in the **cafe** bucket and other objects in the bucket.
    - The third statement, identified by the **Sid** key name **AllowUserSpecificActionsOnlyInTheSpecificPrefix**, defines permissions that specify the actions that the user can perform on the objects in the **cafe-*/images/*** folder. The main operations are **GetObject**, **PutObject**, and **DeleteObject**, which correspond to the read, write, and delete permissions that you want to grant to the mediacouser user. Two additional operations are included for eventual version-related actions.
24. To collapse the policy, choose **-**.

# Task 3.2: Reviewing the mediacouser IAM user

In this section, you review the properties of the mediacouser user.

25. In the IAM console navigation pane, choose **Users**.
26. From the **Users** list, select **mediacouser**.
    On the **Permissions** tab, you should see two policies: **IAMUserChangePassword** and **mediaCoPolicy**. These policies are assigned to the mediaco IAM group that you reviewed in the previous task.
27. To verify that you see the mediaco IAM group, choose the **Groups** tab.
    The mediacouser user is a member of this group and therefore inherits the permissions assigned to the mediaco group.
28. Choose the **Security credentials** tab.
29. In the **Access keys** section, choose **Create access key**, and choose the following options:

  - Choose **Command Line Interface (CLI)**.
  - Select the check box for **I understand the above recommendation and want to proceed to create an access key.**

30. Choose **Next**.
31. Choose **Create access key**.
    The following message displays: *Access key created*
32. Choose **Download .csv file**.
33. Choose **Done**.
34. On the **mediacouser** page, from the **Security credentials** tab, copy the **Console sign-in link**.
    You use this link in the next task.

# Task 3.3: Testing the mediacouser permissions

In this task, you test the permissions that you have reviewed by signing in to AWS Management Console as mediacouser and performing the view, upload, and delete operations on the contents of the images folder in the S3 share bucket. These actions are the use cases that the external media company user is expected to perform on the bucket. In addition, you test the unauthorized use case, where the external user attempts to change the bucket permissions.

35. To sign in to the AWS Management Console as the mediacouser user, use one of the following options:
    **Important:** Do not sign out of the session where you are signed in as the **voclabs/user**. Instead, choose one of two options:
    - Option 1: Use a different browser.
    - Option 2: Use the same browser type, but open a new incognito or private browser session.
      For either option that you choose, enter the **Console sign-in link** that you copied from the previous step into your new browser tab. The AWS Management Console sign-in page opens and already has the **Account ID** populated.
36. On the sign-in page, enter the following credentials:
    - Enter the following credentials:
      - **IAM user name:** `mediacouser`.
      - **Password:** `Training1!`.
37. Choose **Sign in**.
38. On the new **AWS Management Console** page, in the **Search** bar, enter and choose `S3` to open the **S3 Management Console**.
39. From the list of buckets, select the bucket that you created earlier.

40. To display the list of images that were uploaded earlier, select **images/**.
41. To test the **view** use case, select **Donuts.jpg**, and choose **Open**.
    A new browser tab should open that shows a picture of various donuts.
    **Tip:** If a new browser tab does not open, a banner or icon at the top of your browser will indicate that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and choose **Allow pop-ups**.
42. Close the browser tab that shows the Donuts.jpg image.
43. In the **Console** tab, in the breadcrumb trail at the top, choose **images/** to see the contents of the images folder again.
44. To test the **upload** use case, choose **Upload**.
45. On the **Upload** page, choose **Add files**, and choose any image or picture from your local computer.
46. Choose **Upload**.
47. To close the **Upload: status** page, choose **Close**.
48. Select the file that you uploaded, and choose **Open**.
    A new browser tab should open that shows the file that you uploaded.
49. Close the browser tab that shows the file that you uploaded.
50. To test the **delete** use case, in the **Console** tab, in the image list, select the check box for **Cup-of-Hot-Chocolate.jpg**.
51. Choose **Delete**.
52. On the **Delete objects** page, in the **Delete objects?** box, enter `delete`.
53. Choose **Delete objects**.
    The object is deleted and no longer appears in the image list.
54. To close the **Delete objects: status** page, choose **Close**.
    Next, you test the **unauthorized** use case where mediacouser attempts to change the bucket's permissions.
55. In the breadcrumb trail at the top, choose your bucket to return to the bucket content list.
56. Choose the **Permissions** tab.
    This is where you can change a bucket's permissions.
    Notice that for **Permissions overview**, the following error message is displayed: "Insufficient permissions." mediacouser is prevented from changing the bucket permissions. You could also try to upload a file directly to the root of the bucket. This action should also fail.
57. Sign out of the Amazon S3 console as **mediacouser**.
    You have successfully created an Amazon S3 bucket, and you have confirmed that it is securely configured for file sharing with another user.

# Task 4: Configuring event notifications on the S3 share bucket

In this task, you configure the S3 share bucket to generate an event notification to an SNS topic whenever the contents of the bucket change. The SNS topic then sends an email message to its subscribed users with the notification message. Specifically, you perform the following steps:

- Create the s3NotificationTopic SNS topic.
- Grant Amazon S3 permission to publish to the topic.
- Subscribe to the topic.
- Add an event notification configuration to the S3 bucket.

## Task 4.1: Creating and configuring the s3NotificationTopic SNS topic

58. Return to the AWS Management Console window where you are signed in as **voclabs/user**.
59. On the **AWS Management Console**, in the **Search** bar, enter `SNS` and choose **Simple Notification Service** to open the **Simple Notification Service** console.
60. If necessary, to open the navigation pane, choose the menu icon () on the left.
61. In the navigation pane, choose **Topics**.
62. Choose **Create topic**.
63. Choose **Standard**.
64. For **Name**, enter `s3NotificationTopic`.
65. Choose **Create topic**.
    A message is displayed indicating that the s3NotificationTopic SNS topic has been successfully created.
66. From the **s3NotificationTopic** page in the **Details** section, copy and paste the **ARN** value to a text editor. You need this value later in this lab.
67. To configure the topic's access policy, choose **Edit**.
68. Expand the **Access policy - *optional*** section.
69. Replace the contents of the JSON editor with the following policy. In the JSON object, replace *<ARN of s3NotificationTopic>* with the ARM value that you copied

earlier, and replace <*cafe-xxxnnn*> with your S3 bucket name. Remember to remove the enclosing angle brackets (< >).

70.

```
{
  "Version": "2008-10-17",
  "Id": "S3PublishPolicy",
  "Statement": [
    {
      "Sid": "AllowPublishFromS3",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "SNS:Publish",
      "Resource": "<ARN of s3NotificationTopic>",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:<cafe-xxxnnn>"
        }
      }
    }
  ]
}
```

71. Take a moment to review the intent of this policy. It grants the cafe S3 share bucket permission to publish messages to the s3NotificationTopic SNS topic.
72. Choose **Save changes**.

    Next, you subscribe to the topic to receive the event notifications from the S3 share bucket.
73. In the **s3NotificationTopic** pane, choose the **Subscriptions** tab.
74. Choose **Create subscription**.
75. Choose the **Topic ARN** box, and choose the **s3NotificationTopic** SNS topic that appears as an option.
76. From the **Protocol** dropdown list, choose **Email**.
77. In the **Endpoint** box, enter an email address that you can access.
78. Choose **Create subscription**.

    A message displays that confirms that the subscription was created successfully.
79. Check the inbox for the email address that you provided. You should see an email message with the subject *AWS Notification - Subscription Confirmation*.
80. Open the email message, and choose **Confirm subscription**. A new browser tab opens and displays a page with the message *Subscription confirmed!*

## Task 4.2: Adding an event notification configuration to the S3 bucket

In this task, you create an event notification configuration file that identifies the events that Amazon S3 will publish and the topic destination where Amazon S3 will send the event notifications. You then use the s3api CLI commands to associate this configuration file with the S3 share bucket.

79. In the terminal window for the CLI Host instance, enter the following command to edit a new file named s3EventNotification.json:
80. `vi s3EventNotification.json`
81. In the editor, to change to insert mode, press `i`.
82. In the following JSON object, replace *<ARN of s3NotificationTopic>* with the ARN value that you recorded earlier. Remember to remove the enclosing angle brackets (< >). Copy and paste your customized JSON configuration into the editor window.

```
{

  "TopicConfigurations": [

    {

      "TopicArn": "<ARN of s3NotificationTopic>",

      "Events": ["s3:ObjectCreated:*","s3:ObjectRemoved:*"],

      "Filter": {

        "Key": {

          "FilterRules": [

            {

              "Name": "prefix",

              "Value": "images/"

            }

          ]

        }

      }
      }
    ]
  }
```

Take a moment to review the intent of this configuration. It requests that Amazon S3 publish an event notification to the s3NotificationTopic SNS topic whenever an ObjectCreated or ObjectRemoved event is performed on objects inside an Amazon S3 resource with a prefix of **images/**.

82. Press ESC to exit insert mode.
83. To save the file and exit the editor, enter `:wq` and press Enter.
84. To associate the event configuration file with the S3 share bucket, run the following command. In the command, replace <*cafe-xxxnnn*> with your S3 bucket name:
85. 
    ```
    aws s3api put-bucket-notification-configuration --bucket <cafe-xxxnnn> --notification-configuration file://s3EventNotification.json
    ```
86. Wait a few moments, and then check the inbox for the email address that you used to subscribe to the topic. You should see an email message with the subject *Amazon S3 Notification*.
87. Open the email message, and examine the notification message. It should be similar to the following:
    {"Service":"Amazon S3","Event":"s3:TestEvent","Time":"2019-04-26T06:04:27.405Z","Bucket":"","RequestId":"7A87C25E0323B2F4","HostId":"fB3Z...SD////PWubF3E7RYtVupg="}
    Notice that the value of the **"Event"** key is **"s3:TestEvent"**. Amazon S3 sent this notification as a test of the event notifications configuration that you set up.

# Task 5: Testing the S3 share bucket event notifications

In this task, you test the configuration of the S3 share bucket event notification by performing the use cases that mediacouser expects to perform on the bucket. These actions include putting objects into and deleting objects from the bucket, which send email notifications. You also test an unauthorized operation to verify that it is rejected. You use the AWS s3api CLI command to perform these operations on the S3 share bucket.

87. To configure the CLI Host's AWS CLI client software to use the mediacouser credentials, in the SSH window for the CLI Host instance, enter the following command:
88. 
    ```
    aws configure
    ```
89. At the prompts, enter the following:
    - **AWS Access Key ID**: Copy and paste the value of the **Access key ID** of mediacouser, which is in the mediacouser_accessKeys.csv file that you downloaded in Task 3.
    - **AWS Secret Access Key**: Copy and paste the value of the **Secret Access Key** of mediacouser from the same file that you downloaded in Task 3.
    - **Default region name**: Press Enter at the prompt to keep the same Region that you selected earlier in this lab.
    - **Default output format**: Enter `json`.
90. Next, you test the **put** use case by uploading the Caramel-Delight.jpg image file from the new-images folder on the CLI Host.
91. To upload this file, run the following command. In the command, replace *<cafe-xxxnnn>* with your S3 bucket name:
92. 
    ```
    aws s3api put-object --bucket <cafe-xxxnnn> --key images/Caramel-Delight.jpg --body ~/new-images/Caramel-Delight.jpg
    ```
93. After the command completes, it returns the **ETag** (Entity tag) of the uploaded object.
94. Check the inbox for the email address that you used to subscribe to the s3NotificationTopic SNS topic. You should see a new email message with the subject *Amazon S3 Notification*.
95. Open the email message, and examine the notification message. Notice the following information:
    - The value of the **eventName** key is **ObjectCreated:Put**.
    - The value of the **key** object is **images/Caramel-Delight.jpg**, which is the image file key that you specified in the command.
96. This notification indicates that a new object with a key of **images/Caramel-Delight.jpg** was added (put) into the S3 share bucket.  
    Next, you test the **get** use case by getting the object with a key of **images/Donuts.jpg** from the bucket.
97. To get this object, run the following command. In the command, replace *<cafe-xxxnnn>* with your S3 bucket name:
98. 
    ```
    aws s3api get-object --bucket <cafe-xxxnnn> --key images/Donuts.jpg Donuts.jpg
    ```
99. Notice that an email notification was not generated for this operation. This operation does not generate an email notification because the share bucket is

configured to send notifications only when objects are created or deleted. Next, you test the **delete** use case by deleting the object with a key of **images/Strawberry-Tarts.jpg** from the bucket.

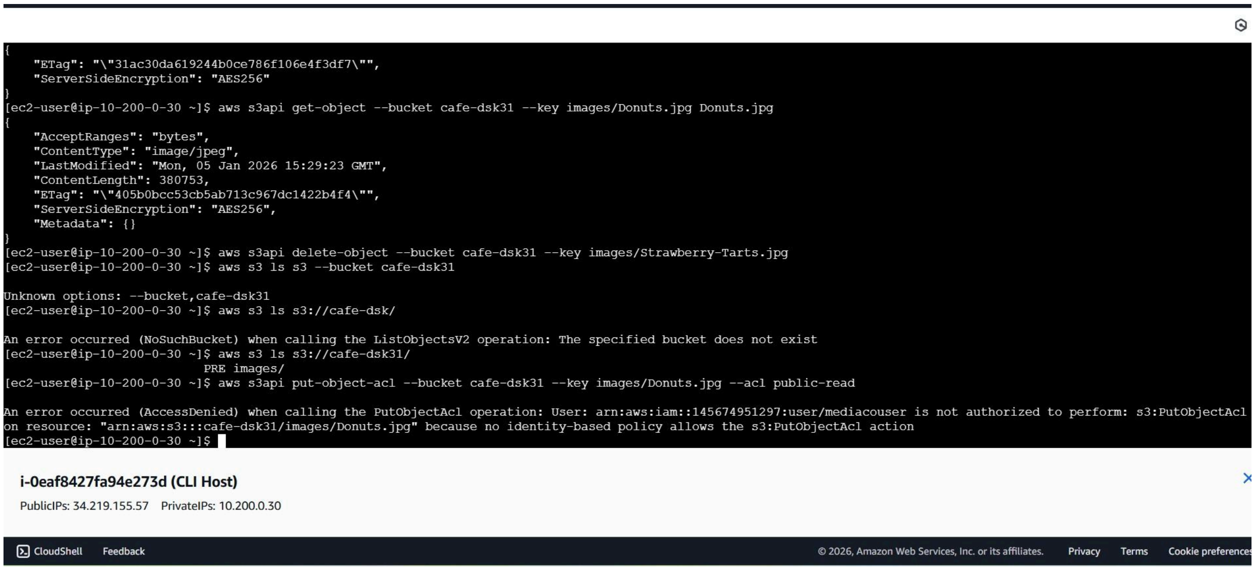
100. To delete this object, run the following command. In the command, replace *<cafe-xxxnnn>* with your S3 bucket name:
101. 
```
aws s3api delete-object --bucket <cafe-xxxnnn> --key
images/Strawberry-Tarts.jpg
```
102. Check the inbox for the email address that you used to subscribe to the s3NotificationTopic SNS topic. You should see a new email message with the subject *Amazon S3 Notification*.
103. Open the email message, and examine the notification message. Notice the following information:
    - The value of the **eventName** key is **ObjectRemoved:Delete**.
    - The value of the object **key** is **images/Strawberry-Tarts.jpg**, which is the image file key that you specified in the command.
104. This notification indicates that the object with a key of **images/Strawberry-Tarts.jpg** was deleted from the S3 share bucket. Finally, you test an unauthorized use case.
105. To try to change the permission of the Donuts.jpg object so that it can be read publicly, run the following command. In the command, replace <*cafe-xxxnnn*> with your S3 bucket name:
106. 
```
aws s3api put-object-acl --bucket <cafe-xxxnnn> --key
images/Donuts.jpg --acl public-read
```
107. The command fails and displays the following error message as expected: "An error occurred (AccessDenied) when calling the PutObjectAcl operation: Access Denied"

# Conclusion

Congratulations! You now have successfully done the following:

- Used the s3api and s3 AWS CLI commands to create and configure an S3 bucket
- Verified write permissions to a user on an S3 bucket
- Configured event notification on an S3 bucket

88.



```
{
    "ETag": "\"31ac30da619244b0ce786f106e4f3df7\"",
    "ServerSideEncryption": "AES256"
}
[ec2-user@ip-10-200-0-30 ~]$ aws s3api get-object --bucket cafe-dsk31 --key images/Donuts.jpg Donuts.jpg
{
    "AcceptRanges": "bytes",
    "ContentType": "image/jpeg",
    "LastModified": "Mon, 05 Jan 2026 15:29:23 GMT",
    "ContentLength": 380753,
    "ETag": "\"405b0bcc53cb5ab713c967dc1422b4f4\"",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
[ec2-user@ip-10-200-0-30 ~]$ aws s3api delete-object --bucket cafe-dsk31 --key images/Strawberry-Tarts.jpg
[ec2-user@ip-10-200-0-30 ~]$ aws s3 ls s3 --bucket cafe-dsk31

Unknown options: --bucket,cafe-dsk31
[ec2-user@ip-10-200-0-30 ~]$ aws s3 ls s3://cafe-dsk/

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
[ec2-user@ip-10-200-0-30 ~]$ aws s3 ls s3://cafe-dsk31/
                           PRE images/
[ec2-user@ip-10-200-0-30 ~]$ aws s3api put-object-acl --bucket cafe-dsk31 --key images/Donuts.jpg --acl public-read

An error occurred (AccessDenied) when calling the PutObjectAcl operation: User: arn:aws:iam::145674951297:user/mediacouser is not authorized to perform: s3:PutObjectAcl
on resource: "arn:aws:s3:::cafe-dsk31/images/Donuts.jpg" because no identity-based policy allows the s3:PutObjectAcl action
[ec2-user@ip-10-200-0-30 ~]$
```

**i-0eaf8427fa94e273d (CLI Host)**

PublicIPs: 34.219.155.57    PrivateIPs: 10.200.0.30