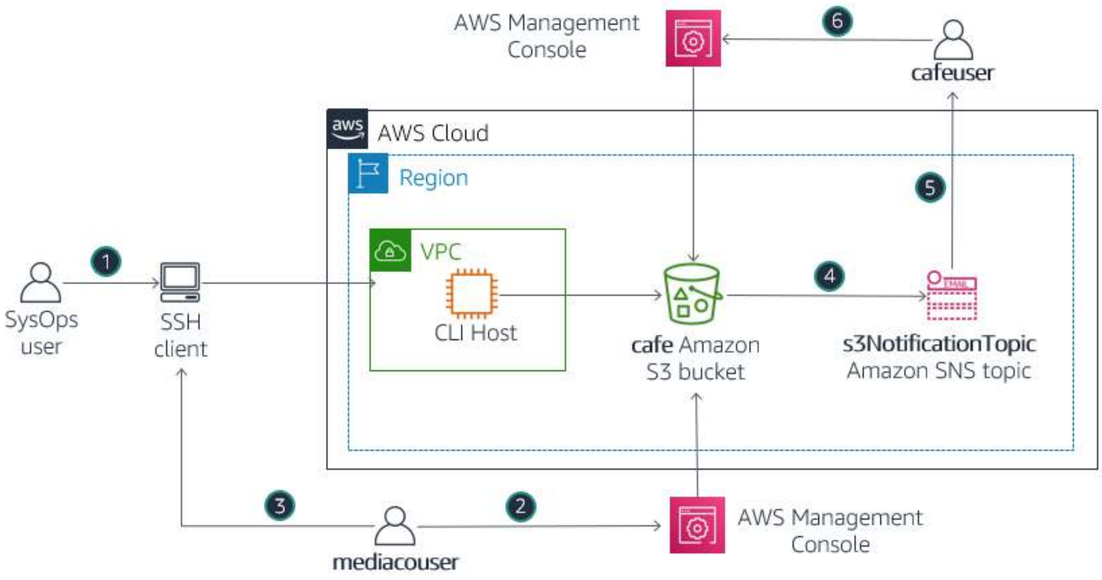
Activity - Work with Amazon S3

Activity overview

In this activity, you create and configure an Amazon S3 bucket to share images between a Café user (*cafeuser*) and an external media company user (*mediacouser*) that was hired to provide pictures of the products sold by the Café. You also configure the S3 bucket to automatically generate an email notification to the Café user when the bucket contents are modified.

This diagram shows the component architecture of the Amazon S3 file-sharing solution and illustrates its usage flow, which consists of the following steps:

Amazon S3 share bucket architecture diagram:



1. The systems operator at Café creates and configures an Amazon S3 bucket named *cafe* as a container for sharing images. An AWS Identity and Access Manager (IAM) user named *mediacouser* has been pre-created with appropriate S3 permissions to allow a user from the external media company to add, change, or delete images from the bucket. Another IAM user named *cafeuser* has also been pre-created to allow Martha or Frank to view the contents of the bucket when they receive a notification. The necessary S3 permissions are reviewed for each user to make sure that access to the bucket is secure and appropriate for each role.
2. When new product pictures are available or when existing pictures must be updated, a representative from the media company logs in to the AWS Management Console as *mediacouser* to upload, change, or delete the bucket contents.
3. As an alternative, the *mediacouser* can use the AWS Command Line Interface (AWS CLI) to manipulate the contents of the S3 bucket.
4. When Amazon S3 detects a change in the bucket's contents, it publishes a notification to the

s3NotificationTopic Amazon Simple Notification Service (Amazon SNS) topic.

5. The *cafeuser*, who is subscribed to the **s3NotificationTopic**, receives an email message that contains the details of the changes to the bucket’s contents.
6. The *cafeuser* then logs in to the AWS Management Console to view the newly uploaded images or the results of the changes to the bucket's contents.

Activity objectives

After completing this activity, you will be able to:

- **Use** the *s3api* and *s3* AWS CLI commands to create and configure an Amazon S3 bucket.
- **Configure** an Amazon S3 bucket for file sharing with an external user.
- **Secure** an Amazon S3 bucket for different access requirements by using Amazon S3 permissions.
- **Configure** event notification on an Amazon S3 bucket.

Business case relevance

A new business requirement for Café—Share files with an external partner

Frank has been experimenting with new recipes and has started to expand the Café's product list. He wants to add these new products to the online menu on the Café's website. He has hired an external media company to create a portfolio of pictures that showcase the new products.

Frank wants to receive the pictures from the media company electronically in one convenient and secure location. He also wants to be notified by email when pictures are uploaded so that he can approve them before they are deployed to the website. Frank asks Sofîa to consult with the AWS team for a recommended solution.

Faythe, an AWS developer, suggests using Amazon S3 to share files with the external partner.

In this activity, you will take on the role of Sofîa, and implement the Amazon S3 file-sharing solution. You will also take on the roles of Frank and the media company user to test and validate the Amazon S3 usage scenario.

Activity steps

Duration: This activity requires approximately **90 minutes** to complete.

Launching the activity environment

7. At the top of these instructions, click Start Lab to launch your lab.

   A Start Lab panel opens displaying the lab status.

8. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.

9. At the top of these instructions, click AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in. On the **New AWS Console Home** pop-up window, Click **Maybe later**.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

10. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

Task 1: Connect to the AWS CLI Host instance by using SSH

Begin by opening a Secure Shell (SSH) session to the **CLI Host** instance that is provided in your lab environment. You will be using the AWS CLI to create the Amazon S3 bucket and to perform most of the bucket configuration actions that are required in this activity.

If you are a Windows user, follow the steps described in Task 1.1. Otherwise, if you are a macOS or Linux user, follow the steps in Task 1.2.

Task 1.1: Windows SSH

💬 These instructions are for Windows users only.

If you are using macOS or Linux, skip to the next section.

11. Read through the three bullet points in this step before you start to complete the actions, because you will not be able see these instructions when the Details panel is open.

    - Click on the Details drop down menu above these instructions you are currently reading, and then click Show. A Credentials window will open.
    - Click on the **Download PPK** button and save the **labsuser.ppk** file. Typically your browser

   will save it to the Downloads directory.
   - Then exit the Details panel by clicking on the **X**.

12. Download needed software.

    - You will use **PuTTY** to SSH to Amazon EC2 instances. If you do not have PuTTY installed on your computer, download it here.

13. Open **putty.exe**

14. Configure PuTTY to not timeout:

    - Click **Connection**
    - Set **Seconds between keepalives** to `30`

    This allows you to keep the PuTTY session open for a longer period of time.

15. Configure your PuTTY session:

    - Click **Session**
    - **Host Name (or IP address):** Copy and paste the **IPv4 Public IP address** for the CLI Host instance. To find it, return to the EC2 Console and click on **Instances**. Check the box next to the CLI Host instance and in the *Description* tab copy the **IPv4 Public IP** value.
    - Back in PuTTy, in the **Connection** list, expand ⊞ **SSH**
    - Click **Auth** (don't expand it)
    - Click **Browse**
    - Browse to and select the lab#.ppk file that you downloaded
    - Click **Open** to select it
    - Click **Open**

16. Click **Yes**, to trust the host and connect to it.

17. When prompted **login as**, enter: `ec2-user`

    This will connect you to the EC2 instance.

18. Windows Users: Click here to skip ahead to the next task.

Task 1.2: macOS/Linux SSH

These instructions are for Mac/Linux users only. If you are a Windows user, skip ahead to the next task.

19. Read through the three bullet points in this step before you start to complete the actions, because you will not be able see these instructions when the Details panel is open.

    - Click on the Details drop down menu above these instructions you are currently reading, and then click Show . A Credentials window will open.
    - Click on the **Download PEM** button and save the **labsuser.pem** file.
    - Then exit the Details panel by clicking on the **X**.

20. Open a terminal window, and change directory `cd` to the directory where the labsuser.pem file was downloaded.

    For example, run this command, if it was saved to your Downloads directory:

    ```
    cd ~/Downloads
    ```

21. Change the permissions on the key to be read only, by running this command:

    ```
    chmod 400 labsuser.pem
    ```

22. Return to the AWS Management Console, and in the EC2 service, click on **Instances**. Check the box next to the CLI Host instance.
23. In the *Description* tab, copy the **IPv4 Public IP** value.
24. Return to the terminal window and run this command (replace **<public-ip>** with the actual public IP address you copied):

    ```
    ssh -i labsuser.pem ec2-user@<public-ip>
    ```

25. Type `yes` when prompted to allow a first connection to this remote SSH server.

    Because you are using a key pair for authentication, you will not be prompted for a password.

Task 1.2: Configure the AWS CLI on the CLI Host EC2 Instance

26. Discover the region in which the CLI Host instance is running:

    ```
    curl http://169.254.169.254/latest/dynamic/instance-identity/document | grep region
    ```

    You will use this region information in a moment.

27. Update the AWS CLI software with the credentials.

    ```
    aws configure
    ```

28. At the prompts, enter the following information:
    - **AWS Access Key ID**: Click on the Details drop down menu above these instructions, and then click Show. Copy the **AccessKey** value and paste it into the terminal window.
    - **AWS Secret Access Key**: Copy and paste the **SecretKey** value from the same Credentials screen.
    - **Default region name**: Type in the name of the region where your EC2 instances are running, which you just discovered a moment ago. For example, `us-east-1` or `eu-west-2`.
    - **Default output format**: `json`

Task 2: Create and initialize the Amazon S3 share bucket

AWS provides two AWS CLI tools—the *s3* CLI and the *s3api* CLI—that you can use to interact with the Amazon S3 service through the command line interface. The *s3* CLI exposes a smaller number of commands than the *s3api* CLI, but the *s3* CLI supports higher-level operations that help simplify most commonly performed tasks.

In this task, you use the AWS *s3* CLI to create the Amazon S3 share bucket and initialize it with some images. You then list the bucket contents to verify the success of your actions.

Tips:

- Use the AWS CLI documentation for s3 to help you determine the correct syntax for the AWS s3 CLI commands that you must use in this task.
- Use your favorite text editor to make any required substitutions to a command before you run it. Specifically, copy the command to a text editor, make the necessary substitutions, and then paste the revised command to the SSH window.

29. Create the *<cafe-xxxnnn>* S3 bucket. Because an S3 bucket name must be unique across all existing bucket names in Amazon S3, you will add a suffix to the name with a format of *-xxxnnn*. For *xxx*, substitute your initials. For *nnn*, substitute a random number. In the SSH window for the **CLI Host** instance, enter:

    ```
    aws s3 mb s3://<cafe-xxxnnn> --region <region>
    ```

    In the command, substitute **<cafe-xxxnnn>** with your unique S3 bucket name. Also, substitute **<region>** with the region where your CLI Host instance is running.

    When the *make bucket (mb)* command completes successfully, it returns the name of the bucket.

30. Load some images in the S3 bucket under the **/images** prefix. Sample image files are provided in the **initial-images** folder on the CLI Host. In the SSH window for the **CLI Host** instance, enter:

    ```
    aws s3 sync ~/initial-images/ s3://<cafe-xxxnnn>/images
    ```

    In the command, substitute **<cafe-xxxnnn>** with your unique S3 bucket name.

    As the *synchronize (sync)* command runs, you will see the names of the image files being uploaded.

31. List the bucket contents by using the *s3 ls* command. Choose to display the list in human-readable form with summary totals for the number of objects and their total size at the bottom. In the SSH window for the **CLI Host** instance, enter:

    ```
    aws s3 ls s3://<cafe-xxxnnn>/images/ --human-readable --summarize
    ```

    In the command, substitute **<cafe-xxxnnn>** with your unique S3 bucket name.

    When the *list (ls)* command completes, you will see the details of the image files that were uploaded, and their total number and size.

Task 3: Review the media company user and permissions

Next, you review the permissions assigned to the *mediacouser* IAM user. This user was created for you. The user provides a way for the media company to use the AWS Management Console or the AWS CLI to upload and modify images in the S3 share bucket. You will also review the permissions that the *cafeuser* and *mediacouser* users have been granted to access the S3 bucket.

Task 3.1: Review the cafeuser IAM User

In this section you will review the properties of the *cafeuser* user.

32. In the AWS Management Console browser tab, select **Services > IAM**.
33. In the IAM console navigation pane, click **Users**.
34. Under the User name list, click **cafeuser**.
35. In the **Permissions** tab, click the arrow next to the *AmazonS3ReadOnlyAccess* policy name. This opens a box with a description of the policy and that also enables you to view its JavaScript Object Notation (JSON) definition.
36. Click **{} JSON** and examine the policy's permissions:
    - Which S3 actions does the policy allow?
    - On which S3 resources are the actions allowed?

    Check your answers with the instructor.

Task 3.2: Review the mediaco IAM Group

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

37. In the navigation pane on the left, click **User groups**.
38. In the group name list, click **mediaco**.

    The Summary page for the mediaco group is displayed.
39. In the **Permissions** tab, click **+** beside *IAMUserChangePassword* to expand the policy.
40. Review the AWS managed policy that permits users to change their own password, if needed.
41. Click **-** to colapse.
42. Similarly click **+** next to mediaCoPolicy.

    Note: You may have to scroll down to see the 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 as well as 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. Two additional operations are included for eventual version-related actions.

43. Click **-** to colapse.

Task 3.3: Review the mediacouser IAM User

In this section you will review the properties of the *mediacouser* user.

44. In the IAM console navigation pane, click **Users**.
45. Under the User name list, click **mediacouser**.
46. You should see two policies *IAMUserChangePassword* and *mediaCoPolicy* under the Permissions tab. These policies were assigned to the *mediaco* IAM group, you reviewed in the previous step.
47. Click the **Groups** tab to verify you see the *mediaco* IAM group. The mediaco user is a member of this group and therefore inherits the permissions assigned to the mediaco group.
48. Click the **Security credentials** tab and then click on **Create access key** under Access keys.
49. Download the file and make a note of the Access Key and Secret access key details. You will need this later in the lab.
50. Copy the AWS account number. To do this:
    - Click on the **voclabs/user...** drop down menu in the upper right of the screen.
    - Copy the account number that displays. It will be a 12 digit number with dashes in it.
    - **Important**: Do **not** sign out of the console. Instead, leave this browser tab open. You will return to it later.

Task 3.4: Test the mediacouser permissions

Test the permissions that you have reviewed by logging in to AWS Management Console as *mediacouser* and performing 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.

51. Log in to the AWS Management Console as the mediacouser user. To do this:

    Important: Do not sign out of the session where you are logged in as the voclabs... user. Instead, choose one of two options:

    - **Option 1**: use a different browser type. For example, if you started this lab using Chrome, and you have another browser such as Firefox or Safari or Edge, launch that other browser now.
        - If you are using Option 1, skip the Option 2 section below.

- **Option 2**: use the same browser type, but you must open a new **Incognito** or **private** browser session. For option 2, follow the step below, depending on the type of browser you are using:
- If you are using **Chrome**: Click on the three vertical dots icon in the top right corner of the browser tab in which you are reading these instructions, and choose **New incognito window**.
- If you are using **Firefox** or **Internet Explorer**: Click on the three horizontal bars icon in the top right corner of the browser tab in which you are reading these instructions, and choose **New Private Window**.
- If you are using **Safari**: From the Safari menu bar at the top of your Desktop screen, choose File > **New Private Window**
- If you are using **Edge**: Click on the three horizontal dots icon in the top right corner of the browser tab in which you are reading these instructions, and choose **New InPrivate window**.
- Now in the browser tab that you just opened—**regardless if you followed option 1 or option 2 above**—In the URL bar, browse to `https://aws.amazon.com/console/`
- If you are presented with a web page that does not already include IAM user name and Password fields to fill in, from the **My Account** menu at the top of the page, choose **AWS Management Console**.
- If you see a screen that only prompts for your email address or account ID, select **IAM user** and paste in the **Account ID** that you just copied. However, be sure to remove the dashes from it, then click **Next**.
- In the page that prompts for your Account ID, IAM user name, and Password, enter the following details:
  - For **Account ID or alias**, paste in the **Account ID** that you copied. However, be sure to remove the dashes from it. If you already pasted it into a prior screen, verify that the Account ID is the one you pasted in (update the Account ID if necessary).
  - For **IAM user name**, enter `mediacouser`
  - For **Password**, enter `Training1!`
  - Click **Sign In**

52. Select **Services > Storage > S3** or under **Recently visited** select **S3**.
53. In the list of bucket names, click **cafe-xxxnnn**, where *xxxnnn* corresponds to your bucket's unique name.
54. Click **images**. You see the list of images that were uploaded when you initialized the bucket in Task 2.
55. Test the *view* use case. Click **Donuts.jpg**.
56. Choose **Open**.

    A new browser tab should open and show a picture of various donuts.

    Tip: if a new browser tab did not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."
57. **Close** the browser tab that shows the Donuts.jpg image.

58. In the **Console** tab, in the breadcrumb trail at the top, click **images** to see the contents of the **images** folder again.
59. Test the *upload* use case. Click **Upload**.
60. In the **Upload** dialog box, click **Add files** followed by **Upload**.
61. Select **Close** to close the upload status page.
62. Navigate to the location of a file on your local computer that you can use to test the upload. Preferably, choose a picture file (with a .jpg extension) or a simple text file.
63. Select the file and click **Open**.
64. Click **Upload**. The file is successfully uploaded and displayed in the image list. You can optionally open it to view its content.
65. Test the *delete* use case. In the **Console** tab, in the image list, select the **Cup-of-Hot-Chocolate.jpg** check box.
66. Choose **Delete**.
67. In the **Delete objects** dialog box, under **Delete objects?** enter `delete`.
68. Choose **Delete objects**.
69. The object is deleted and no longer appears in the image list.
70. Choose **Close**.
71. Finally, test the *unauthorized* use case where the mediacouser attempts to change the bucket's permissions. In the breadcrumb trail at the top, click **cafe-xxxnnn** to return to the bucket content list.
72. Click the **Permissions** tab. This is where you can change a bucket's permissions. Notice that an error message is displayed: *Insufficient permissions* The 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.
73. **Sign out** of the Amazon S3 console as the mediacouser (but remain logged in as the voclabs... user in the other browser tab.)

    Great job! 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: Configure event notifications on the Amazon S3 share bucket

Task Overviews

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

a) Create the *s3NotificationTopic* SNS topic.

b) Grant Amazon S3 permission to publish to the topic.

c) Subscribe the *cafeuser* to the topic.

d) Add an event notification configuration to the S3 bucket.

Task 4.1: Create and configure the s3NotificationTopic

74. Return to the AWS Management Console window where you are logged in as the standard **voclabs...** lab user.
75. Select **Services > Application Integration > Simple Notification Service**.
76. If necessary, click the menu icon (☰) on the left to open the navigation pane.
77. In the navigation pane, select **Topics**.
78. Click **Create topic**.
79. Choose **Standard**.
80. In the **Name** box, enter `s3NotificationTopic`.
81. Click **Create topic**.
    A message is displayed indicating that the s3NotificationTopic was successfully created.
82. Copy and paste the value of the topic **ARN** field in a text editor to save it. You will need to supply it when you create the topic's access policy in the next steps and also later in this activity.
83. Configure the topic's access policy. In the **s3NotificationTopic** pane, click **Edit**.
84. Expand the **Access policy - optional** section.
85. Replace the contents of the **JSON editor** with the following policy:

```
{
  "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>"
        }
      }
    }
  ]
}
```

In the JSON object, substitute **<ARN of s3NotificationTopic>** with the value of the topic ARN that you recorded earlier, and **<cafe-xxxnnn>** with your unique S3 bucket name. Also remember to remove the enclosing angle brackets (< >) during the substitution.

86. Take a moment to review the intent of this policy. It grants the *cafe* S3 share bucket the permission to publish messages to the *s3NotificationTopic*.
87. Click **Save changes**.
88. Lastly, subscribe Frank to the topic as the cafeuser who will receive the event notifications from the S3 share bucket. In the **s3NotificationTopic** pane, select the **Subscriptions** tab.
89. Click **Create subscription**.
90. Click in the **topic ARN** box, and select the **s3NotificationTopic** that appears as an option.
91. In the **Protocol** menu, select **Email**.
92. In the **Endpoint** box, enter an email address that you can access.

    Note: For the purposes of this activity, you are going to pretend that you are Frank so you receive the S3 event notifications.
93. Click **Create subscription**. A message is displayed confirming that the subscription was created successfully.
94. Check the inbox for the email address that you provided. You should see an email message with the subject *AWS Notification - Subscription Confirmation*.
95. Open the email message and click **Confirm subscription**. A new browser tab opens and displays a page with the message *Subscription confirmed!*

Task 4.2: Add 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 to associate this configuration file with the Amazon S3 share bucket.

96. In the SSH window for the **CLI Host** instance, edit a new file named **s3EventNotification.json** by entering:

    ```
    vi s3EventNotification.json
    ```
97. In the editor, change to *insert* mode by entering `i`.
98. Customize the following JSON configuration, and then copy and paste it into the editor window.

    ```
    {
      "TopicConfigurations": [
        {
          "TopicArn": "<ARN of s3NotificationTopic>",
          "Events": ["s3:ObjectCreated:*","s3:ObjectRemoved:*"],
    ```

```
      "Filter": {
        "Key": {
          "FilterRules": [
            {
              "Name": "prefix",
              "Value": "images/"
            }
          ]
        }
      }
    }
  ]
}
```

In the JSON object, substitute **<ARN of s3NotificationTopic>** with the value of the topic ARN that you recorded earlier. It will be in the format arn:aws:sns:::s3NotificationTopic. Also remember to remove the enclosing angle brackets (< >) during the substitution.

99. Take a moment to review the intent of this configuration. It requests that Amazon S3 publish an event notification to the *s3NotificationTopic* whenever an *ObjectCreated* or *ObjectRemoved* event is performed on objects inside an S3 resource with a prefix of *images/*.
100. Press ESC to exit *insert* mode.
101. To save the file and exit the editor, enter `:wq`.
102. Associate the event configuration file with the S3 share bucket. In the SSH window for the **CLI Host** instance, enter:

```
 aws s3api put-bucket-notification-configuration --bucket <cafe-xxxnnn> --notification-configuration file://s3EventNotification.json
```

In the command, substitute **<cafe-xxxnnn>** with your unique S3 bucket name.

103. 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*.
104. 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":"<cafe-xxxnnn>","RequestId":"7A87C25E0323B2F4","HostId":"fB3Z...SD////PWubF3E7RYtVupg="}
```

Notice that the value of the **"Event"** key is **"s3:TestEvent"**. This notification was sent by Amazon S3 as a test of the event notifications configuration that you just set up.

Task 5: Test the Amazon 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 the *mediacouser* expects to perform on the bucket. These actions include putting and deleting objects in the bucket, which should generate email notifications to Frank. You also test an unauthorized operation to verify that it is rejected. You use the AWS *s3api* CLI to perform these operations on the S3 share bucket.

Tip: Use the AWS CLI documentation for s3api to help you determine the correct syntax for the AWS s3api CLI commands that you must use in this task.

105. Configure the CLI Host's AWS CLI client software to use the *mediacouser* credentials. In the SSH window for the **CLI Host** instance, enter:

```
aws configure
```

106. At the prompts, enter the following:
    - **AWS Access Key ID**: Copy and paste the value of the **Access Key ID** of the **mediacouser**, which is in the accessKeys.csv file you downloaded in Task 3.
    - **AWS Secret Access Key**: Copy and paste the value of the **Secret Access Key** of the **mediacouser**, from the same file downloaded in Task 3.
    - **Default region name**: keep the same region you set earlier in this activity, by clicking ENTER at the prompt.
    - **Default output format**: `json`
107. Test the *put* use case by uploading the **Caramel-Delight.jpg** image file from the **new-images** folder on the CLI Host. In the SSH window, enter:

```
aws s3api put-object --bucket <cafe-xxxnnn> --key images/Caramel-Delight.jpg
--body ~/new-images/Caramel-Delight.jpg
```

In the command, make sure to substitute <*cafe-xxxnnn*> with your unique S3 bucket name. After the command completes, it returns the *ETag* (Entity tag) of the uploaded object.

108. Check the inbox for the email address that you used to subscribe to the s3NotificationTopic. You should see a new email message with the subject *Amazon S3 Notification*.

109. Open the email message and examine the notification message. Notice that:
    - The value of the **"eventName"** key is **"ObjectCreated:Put"**.
    - The value of the object **"key"** is **"images/Caramel-Delight.jpg"**, which is the image file key that you specified in the command.

This notification indicates that a new object with a key of **images/Caramel-Delight.jpg** was added (put) to the S3 share bucket.

110. Test the *get* use case. Get the object with a key of **images/Donuts.jpg** from the bucket. In the SSH window, enter:

```
 aws s3api get-object --bucket <cafe-xxxnnn> --key images/Donuts.jpg
Donuts.jpg
```

In the command, substitute **<cafe-xxxnnn>** with your unique S3 bucket name. After the command completes, it returns some metadata about the retrieved object, including its *ContentLength*.

Notice that an email notification was not generated for this operation. This is because the share bucket is configured to send notifications for only the object create and object delete actions.

111. Test the *delete* use case. Delete the object with a key of **images/Strawberry-Tarts.jpg** from the bucket. In the SSH window, enter:

```
 aws s3api delete-object --bucket <cafe-xxxnnn> --key images/Strawberry-
Tarts.jpg
```

   In the command, substitute **<cafe-xxxnnn>** with your unique S3 bucket name.

112. Check the inbox for the email address that you used to subscribe to the s3NotificationTopic. You should see a new email message with the subject *Amazon S3 Notification*.
113. Open the email message and examine the notification message. Notice that:
   - 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.

   This notification indicates that the object with a key of **images/Strawberry-Tarts.jpg** was deleted from the S3 share bucket.

114. Finally, test an unauthorized use case. Try to change the permission of the **Donuts.jpg** object so that it can be read publicly. In the SSH window, enter:

```
 aws s3api put-object-acl --bucket <cafe-xxxnnn> --key images/Donuts.jpg
--acl public-read
```

   In the command, substitute **<cafe-xxxnnn>** with your unique S3 bucket name.

   The command fails and displays the following error message:

```
 An error occurred (AccessDenied) when calling the PutObjectAcl operation:
Access Denied
```

   The permissions and event notifications configuration of the S3 share bucket work as intended. Good work!

Update from Café

The Amazon S3 file-sharing solution has streamlined how Café exchanges images with the external media company. Frank is pleased that the solution is secure and that he receives notifications automatically when new pictures are uploaded. He is thinking of using the same approach to exchange documents electronically with his suppliers!