Lab 1: Introduction to Amazon Simple Storage Service (Amazon S3)

Source:
https://awseducate.instructure.com/courses/746/assignments/3071?module_item_id=13184

Requirement:

Participants using their account to login to the AWS Educate.

Lab overview and objectives

This lab teaches you the basic feature functionality of Amazon Simple Storage Service (Amazon S3) using the AWS Management Console.

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and from all industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, Internet of Things (IoT) devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999 percent (11 9's) of durability and stores data for millions of applications for companies all around the world.

After completing this lab, you will know how to:

- Create a bucket in Amazon S3
- Add an object to a bucket
- Manage access permissions on an object and a bucket
- Create a bucket policy
- Use bucket versioning

Duration

This lab requires approximately **60 minutes** to complete. You will have a total time of **240** minutes to complete this lab.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Accessing the AWS Management Console

1. At the top of these instructions, choose `Start Lab` to launch your lab.

   A **Start Lab** panel opens, and it displays the lab status.

   Tip: If you need more time to complete the lab, choose the **Start Lab** button again to restart the timer for the environment. Once you have consumed the total lab time that is allowed, you will no longer be able to extend the lab or start a new one. You can find the total time available for this lab in the Duration section of these instructions.

2. Wait until you see the message **Lab status: ready**, and then close the **Start Lab** panel by choosing the **X**.
3. At the top of these instructions, choose `AWS`

   This opens the AWS Management Console in a new browser tab. The system automatically logs you in.

   Tip: If a new browser tab does not open, a banner or icon is usually at the top of your browser with a message that your browser is preventing the website from opening pop-up windows. Choose the banner or icon, and then choose **Allow pop ups**.

4. 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 so that you can follow the lab steps more easily.

Task 1: Creating a bucket

You are new to Amazon S3 and want to test the features and security of Amazon S3 as you configure the environment to hold the Amazon Elastic Compute Cloud (Amazon EC2) report data. You know that every object in Amazon S3 is stored in a bucket, so creating a new bucket to hold the reports is the first thing on your task list.

In this task, you create a bucket to hold your Amazon EC2 report data and then examine the different bucket configuration options.

5. At the upper left of the AWS Management Console, on the **Services** menu, choose **S3**.
6. Choose **Create bucket**

Bucket names must be 3–63 characters long and consist of only lowercase letters, numbers, or hyphens. The bucket name must be globally unique across all of Amazon S3 regardless of account or Region, and you cannot change a bucket name after creating the bucket. As you enter a bucket name, a help box displays showing any violations of the naming rules. Refer to the Amazon S3 bucket naming rules in the **Additional resources** section at the end of the lab for more information.

7. In the **General configuration** section, enter the following as the **Bucket name**: `reportbucket(NUMBER)`

In the bucket name, replace **(NUMBER)** with a random number so that your bucket has a unique name.

- Example bucket name: `reportbucket987987`

Leave **Region** at its default value.

By selecting a particular Region, you can optimize latency, minimize costs, or address regulatory requirements. Objects stored in a Region never leave that Region unless you explicitly transfer them to another Region.

8. Choose **Create bucket**

Task 2: Uploading an object to the bucket

Now that you have created a bucket for your report data, you are ready to work with objects. An object can be any kind of file: a text file, a photo, a video, a .zip file, and so on. When you add an object to Amazon S3, you have the option to include metadata with the object and set permissions to control access to the object.

In this task, you test uploading objects to your reportbucket. You have a screen capture of a daily report and want to upload this image to your S3 bucket.

9. Right-click the following link: new-report.png. Choose **Save link as**, and save the file to your desktop.
10. In the **S3 Management Console**, find and select the bucket name that starts with **reportbucket**.
11. Choose **Upload**

This step launches an upload wizard. Use this wizard to upload files either by selecting them from a file chooser or by dragging them to the Amazon S3 window.

12. Choose **Add files**
13. Browse to and select the **new-report.png** file that you downloaded previously.
14. At the bottom of the page, choose **Upload**

Your file is successfully uploaded when the green bar indicating **Upload succeeded** appears.

15. In the **Upload: status** section in the upper right, choose **Close**

Task 3: Making an object public

Security is a priority in Amazon S3. Before you configure your EC2 instance to connect to the reportbucket, you want to test the bucket and object settings for security.

In this task, you configure permissions on your bucket and your object to test accessibility.

First, you attempt to access the object to confirm that it is private by default.

16. In the **reportbucket** overview page, on the **Objects** tab, locate the **new-report.png** object, and choose the **new-report.png** file name.

The **new-report.png** overview page opens. The navigation in the upper left updates with a link to return to the bucket overview page.

17. In the **Object overview** section, locate and copy the **Object URL** link.

The link should look similar to the following: https://reportbucket987987.s3-us-west-2.amazonaws.com/new-report.png

18. Open a new browser tab and paste the object URL link into the address field, and then press **Enter**.

You receive an **Access Denied** error because objects in Amazon S3 are private by default.

Now that you've confirmed that the default security of Amazon S3 is private, you test how to make the object publicly accessible.

19. Keep the browser with the Access Denied error open, and return to the web browser tab with the **S3 Management Console**.
20. You should still be on the new-report.png **Object overview** tab.
21. In the upper right, choose the **Object actions** dropdown menu, you will notice that **Make public via ACL** is greyed out.
22. In the upper left of the page, choose the **reportbucket** name in the navigation to go back to the main **reportbucket** overview page.
23. Choose the **Permissions** tab.
24. We need to allow the use of ACLs first. Under **Object Ownership** choose **Edit**.
25. Choose **ACLs enabled**.
26. Choose **Bucket owner preferred**.
27. Choose the check box next to **I acknowledge that ACLs will be restored**.
28. Choose **Save Changes**
29. Under **Block public access (bucket settings)**, choose **Edit** to change the settings.
30. Clear the check box for the **Block *all* public access** option, and then leave all other options cleared.

Notice that all of the individual options remain cleared. When clearing the option for all public access, you must then select the individual options that apply to your situation and security objectives. You use access control lists (ACLs) and bucket policies later in the lab, so these options remain cleared in this task. In a production environment, it is recommended to use the least permissive settings possible. Refer to the Amazon S3 block public access link in the **Additional resources** section at the end of the lab for more information.

31. Choose **Save changes**
32. A dialogue box opens asking you to confirm your changes. Enter `confirm` in the field, and then choose **Confirm**

A message that says **Successfully edited Block Public Access settings for this bucket.** displays at the top of the window.

33. Choose the **Objects** tab.
34. Choose the **new-report.png** file name.
35. At the upper right on the **new-report.png** overview page, choose the **Object actions** dropdown menu, and select **Make public**.

Notice the warning: **When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.** This warning reminds you that if you make the object public, then everyone in the world will be able to read the object.

36. Choose **Make public** and you should see the green banner **Successfully edited public access** at the top of the window.
37. In the upper right, choose **Close** to return to the **new-report.png** object overview.
38. Return to the browser tab that displayed **Access Denied** for the new-report.png object, and refresh the page.

The new-report.png object now displays properly because it is publicly accessible.

39. Close the web browser tab that displays your new-report.png image, and return to the tab with the Amazon S3 Management Console.

In this example, you granted read access to just one specific object. If you would like to grant access to the entire bucket, you need to use a bucket policy, which this lab covers later.

In the next task, you work with your EC2 instance to confirm connectivity to the S3 bucket.

Task 4: Testing connectivity from the EC2 instance

In this task, you connect to your EC2 instance to test connectivity and security to the Amazon S3 reportbucket.

You should already be signed in to the AWS Management Console. If not, follow the steps in the Start Lab section to sign in to the AWS Management Console.

40. On the **Services** menu, choose **EC2**.
41. On the **EC2 Dashboard**, under the **Resources** section, choose **Instances (running)**.
42. Select the check box for **Bastion Host** and choose **Connect**
43. In the **Connect to instance** window, select the **Session Manager** tab for the connection method.

With AWS Systems Manager Session Manager, you can connect to the bastion host instance without the need for specific ports to be open on your firewall or Amazon Virtual Private Cloud (Amazon VPC) security group. Refer to **AWS Systems Manager Session Manager** in the **Additional resources** section at the end of this lab for more information.

44. Choose **Connect**

A new browser tab or window opens with a connection to the bastion host instance.

You are now connected to the EC2 instance that holds the reporting application. Because Session Manager uses HTTPS port 443, it does not require you to open SSH port 22 to the outside world. You are satisfied with this security feature. Now you want to see how EC2 interacts with your S3 bucket.

45. In the bastion host session, enter the following command to change to the home directory (/home/ssm-user/):

```
cd ~
```

The output returns you to the command prompt.

46. Enter the following command to verify that you are in the home directory:

```
pwd
```

The output should be as follows:

```
/home/ssm-user
```

You are now in the ssm-user's home directory where you will run all of the commands in this lab.

47. Enter the following command to list all of your S3 buckets.

```
aws s3 ls
```

The output should look similar to the following:

```
2020-11-11 22:34:46 reportbucket987987
```

You see the reportbucket you created and lab auto-generated buckets.

Note: During the creation of the lab environment, both an *instance profile* (which defines who you are for authentication) and a *role* (which defines what you can do after you authenticate) have been automatically added for the EC2 instance to allow the EC2 instance to list the S3 buckets and objects.

48. In the following command, change *(NUMBER)* at the end of the reportbucket name to the name of the bucket you created. Enter your adjusted command to list all the objects in your reportbucket.

```
aws s3 ls s3://reportbucket(NUMBER)
```

The command looks similar to the following: **aws s3 ls s3://reportbucket987987**

The output should look like the following:

```
2020-11-11 15:46:34     86065 new-report.png
```

There is currently only one object in your bucket. The object is called new-report.png.

49. Enter the following command to change directories into the reports directory.

```
cd reports
```

The output returns you to the command prompt.

50. Enter the following command to list the contents of the directory.

```
ls
```

The output shows some files created in your reports directory to test the application.

```
dolphins.jpg files.zip report-test.txt  report-test1.txt report-test2.txt report-test3.txt  whale.jpg
```

51. In the following command, change *(NUMBER)* at the end of the reportbucket name to the name of the bucket you created. Enter your adjusted command to see if you can copy a file to the S3 bucket.

```
aws s3 cp report-test1.txt s3://reportbucket(NUMBER)
```

The command looks similar to this: **aws s3 cp report-test1.txt s3://reportbucket987987**

The output indicates an **upload failed** error. This error occurs because you have read-only rights to the bucket and do not have the permissions to perform the PutObject action.

52. Leave this window open. and go back to browser tab with the AWS console.

In the next task, you create a bucket policy to add the PutObject permission.

Task 5: Creating a bucket policy

A bucket policy is a set of permissions associated with an S3 bucket. It is used to control access to an entire bucket or to specific directories within a bucket.

In this task, you use the AWS Policy Generator to create a bucket policy to enable read and write access from the EC2 instance to the bucket so that your reporting application can successfully write to Amazon S3.

53. Right-click the following link: sample-file.txt. Choose **Save link as**, and save the file to your desktop.
54. Return to the AWS Management Console, go to the **Services** menu, and select **S3**.
55. In the **S3 Management Console** tab, select the name of your bucket.
56. To upload the **sample-file.txt** file, choose **Upload** and use the same upload process that you used in task 2.
57. On the **reportbucket** overview page, choose the **sample-file.txt** file name. The **sample-file.txt** overview page opens.
58. Under the **Object overview** section, locate and copy the **Object URL** link.
59. In a new browser tab, paste the link into the address field, and then press Enter.

Once again, your browser displays an **Access Denied** message. You need to configure a bucket policy to grant access to all objects in the bucket without having to specify permissions on each object individually.

60. Keep this browser tab open, but return to the tab with the **S3 Management Console**.
61. Select **Services** and select **IAM**. In the left navigation, choose **Roles**.
62. In the **Search** field, enter `EC2InstanceProfileRole`

This is the role that the EC2 instance uses to connect to Amazon S3.

63. Select **EC2InstanceProfileRole**. In the **Summary** section, copy the **Role ARN** to a text file to use in a later step.

It should look similar to the following: **arn:aws:iam::596123517671:role/EC2InstanceProfileRole**

64. Choose **Services** and **S3**, and return to the **S3 Management Console**.
65. Choose the **reportbucket**.

You should see the two objects you uploaded. If not, navigate back to your bucket so that you see the list of objects you have uploaded.

66. Choose the **Permissions** tab.
67. In the **Permissions** tab, scroll to the **Bucket policy** section, and choose **Edit**

A blank **Bucket policy editor** displays. You can create bucket policies manually, or you can create them with the assistance of the **AWS Policy Generator**.

Amazon Resource Names (ARNs) uniquely identify AWS resources across all of AWS. A colon (:) separates each section of the ARN, and each section represents a specific piece of the path to the specified resource. The sections can vary slightly depending on the service being referenced but generally follow the format:

arn:*partition*:*service*:*region*:*account-id*:*resource*

Amazon S3 does not require Region or account-id parameters in ARNs, so those sections are left blank. However, the colon (:) to separate the sections is still used, so it looks similar to *arn:aws:s3:::reportbucket987987*

Refer to the **Amazon Resource Names (ARNs) and AWS Service Namespaces documentation** link in the **Additional resources** section at the end of the lab for more information.

68. Below the **Policy examples** and **Policy generator** buttons, find the **Bucket ARN**. Copy the Bucket ARN to a text file to use in a later step.

It looks like the following:

```
Bucket ARN
arn:aws:s3:::reportbucket987987
```

69. Choose Policy generator

A new web browser tab opens with the AWS Policy Generator.

AWS policies use the JSON format and are used to configure granular permissions for AWS services. You can manually write the policy in JSON, or you can use the AWS Policy Generator to create the policy with a user-friendly web interface.

In the AWS Policy Generator window, configure the following options:

- For **Select Type of Policy**, select **S3 Bucket Policy**.
- For **Effect**, select **Allow**.
- For **Principal**, paste the **EC2 Role ARN** that you copied to a text file in a previous step.
- For **AWS Service**, keep the default setting of **Amazon S3**.
- For **Actions**, select **GetObject** and **PutObject**.

The **GetObject** action grants permission for objects to be retrieved from Amazon S3. Refer to the **Additional resources** section at the end of the lab for links to more information about the actions available for use in Amazon S3 policies.

- For **Amazon Resource Name (ARN)**, enter `*`

70. Choose **Add Statement**. The details of the statement you configured are added to a table below the button. You can add multiple statements to a policy.
71. Choose **Generate Policy**.

A new window displays the generated policy in JSON format. It should look similar to the following:

```
{
  "Version": "2012-10-17",
  "Id": "Policy1604361694227",
  "Statement": [
    {
      "Sid": "Stmt1604361692117",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::416159072693:role/EC2InstanceProfileRole"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "*"
    }
  ]
}
```

72. Copy the policy that you created to your clipboard.
73. Close the web browser tab, and return to the S3 Management Console tab with the **Bucket policy** editor.
74. Paste the bucket policy that you created into the **Bucket policy** editor.
75. In the **Bucket policy** editor, update the **Resource** value replacing `*` with the Bucket ARN you saved earlier followed by `/*`:

The updated **Resource** line in the lab policy should be similar to the following example:

```
"Resource": "arn:aws:s3:::reportbucket987987/*"
```

Confirm that ***/**** appears after your bucket name as the Resource line in this sample shows.

76. Choose **Save changes**.
77. Return to the AWS Systems Manager (Systems Manager) window. If your session has timed out, reconnect to Systems Manager using the previous steps in the lab.
78. Enter the following command to verify that you are in the /home/ssm-user/reports directory.

```
pwd
```

The output should be as follows:

```
/home/ssm-user/reports
```

79. In the command below, replace *(NUMBER)* with the number you used to create your bucket. Enter your adjusted command to list all objects in your reportbucket.

```
aws s3 ls s3://reportbucket(NUMBER)
```

The command should look similar to the following: **aws s3 ls s3://reportbucket987987**

The output should look similar to the following:

```
sh-4.2$ aws s3 ls s3://reportbucket987987
2020-11-02 23:20:27      86065 new-report.png
2020-11-02 23:57:03         90 sample-file.txt
```

80. Enter the following command to list the contents of the reports directory.

```
ls
```

The output returns a list of files.

81. In the command below, replace *(NUMBER)* with the number you used to create your bucket. Enter your adjusted command to try copying the report-test1.txt file to the S3 bucket.

```
aws s3 cp report-test1.txt s3://reportbucket(NUMBER)
```

The command should look like the following: **aws s3 cp report-test1.txt s3://reportbucket987987**

The output returns the following:

```
upload: ./report-test1.txt to s3://reportbucket987987/report-test1.txt
```

82. In the command below, replace *(NUMBER)* with the number you used to create your bucket. Enter your adjusted command to see if the file successfully uploaded to Amazon S3.

```
aws s3 ls s3://reportbucket(NUMBER)
```

The output should look similar to the following:

```
2020-11-11 18:20:23    86065 new-report.png
2020-11-11 18:32:18       31 report-test1.txt
2020-11-11 18:20:22       90 sample-file.txt
```

You have successfully uploaded (PutObject) a file from the EC2 instance to your S3 bucket.

83. In the command below, replace *(NUMBER)* with the number you used to create your bucket. Enter your adjusted command to retrieve (GetObject) a file from Amazon S3 to the EC2 instance.

```
aws s3 cp s3://reportbucket(NUMBER)/sample-file.txt sample-file.txt
```

The output should look similar to the following:

```
download: s3://reportbucket987987/sample-file.txt to ./sample-file.txt
```

84. Enter the following command to see if the file is now in the /reports directory.

```
ls
```

The output should look similar to the following:

```
dolphins.jpg  files.zip  report-test1.txt  report-test2.txt  report-test3.txt  sample-file.txt
```

You now see the sample-file.txt in your file list. Congratulations! You have successfully uploaded and retrieved a file from Amazon EC2 to the S3 bucket.

85. Return to the browser tab that displayed the **Access Denied** error for the **sample-file.txt**, and refresh the page.

The page still displays an error message because the bucket policy gave rights to only the principal called EC2InstanceProfileRole.

86. Go to the AWS Policy Generator, and add another statement to the bucket policy allowing everyone (*) read access (GetObject). Take a moment to generate this policy. This policy allows the EC2InstanceProfileRole to have access to the bucket while giving everyone access to read the objects via the browser.

Below is an expample of the above:

```
    {
      "Sid": "Stmt1604428842806",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::reportbucket987987/*"
    }
```

87. To test if your policy works, go to your browser with the **Access Denied** error and refresh it. If you can read the text, then congratulations! Your policy was successful.

If not, look at the following policy for help. The modified policy should look like the following policy. Notice that there are two statements: one with the EC2InstanceProfileRole and one where the principal is **"*"** for everyone.

If you had trouble generating the policy on your own, you can copy the policy below and paste it into the BucketPolicy Editor. Remember to replace the existing EC2InstanceProfileRole ARN in the policy below with the EC2InstanceProfileRole ARN you copied in a previous step. Ensure that you replace the reportbucket example ARN with the bucket you created and the **/*** appears at the end of the Bucket ARN. See the last line of the policy as an example.

```
{
  "Version": "2012-10-17",
  "Id": "Policy1604428844058",
  "Statement": [
    {
      "Sid": "Stmt1604428821481",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::285058481724:role/EC2InstanceProfileRole"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::reportbucket987987/*"
    },
    {
      "Sid": "Stmt1604428842806",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::reportbucket987987/*"
    }
  ]
}
```

88. Leave the tab open with the sample-file.txt displayed. You return to this tab in the next task.

In this task, you created a bucket policy to allow specific access rights to your bucket. In the next section, you explore how to keep copies of files to prevent against accidental deletion.

Task 6: Exploring versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

For auditing and compliance reasons, you need to enable versioning on your reportbucket. Versioning should protect the reports in the reportbucket against accidental deletion. You are curious to see if this works as advertised. In this task, you enable versioning and test the feature by uploading a modified version of the sample-file.txt file from the previous task.

89. You should be on the S3 bucket **Permissions** tab from the previous task. If you are not, choose the link to the bucket at the upper left of the screen to return to the bucket overview page.
90. On the **reportbucket** overview page, choose the **Properties** tab.
91. Under the **Bucket Versioning** section, click **Edit** select **Enable** then click **Save changes**.

Versioning is enabled for an entire bucket and all objects within the bucket. It cannot be enabled for individual objects.

There are also cost considerations when enabling versioning. Refer to the **Additional resources** section at the end of the lab for links to more information.

92. Right-click this link, and save the text file to your computer **using the same name as the text file in the previous task**: sample-file.txt

Although this file has the same name as the previous file, it contains new text.

93. In the Amazon S3 Management Console, on the reportbucket, choose the **Objects** tab.

Under the **Objects** section, find **Show versions**.

94. Choose **Upload** and use the same upload process that you used in tasks 2 and 5 to upload the new **sample-file.txt** file.
95. Go to the browser tab that has the contents of the sample-file.txt file.
96. Make a note of the contents on the page, and then refresh the page.

Notice that new lines of text appear.

If a version is not otherwise specified, Amazon S3 always returns the latest version of an object.

You can also obtain a list of available versions in the Amazon S3 Management Console.

97. Close the web browser tab with the contents of the text file.
98. In the Amazon S3 Management Console, choose the **sample-file.txt** file name. The **sample-file.txt** overview page opens.
99. Choose the **Versions** tab, and then select the check box for the bottom version, which reads **null**. (This is not the latest version.)
100. Click **Open**.

You should now see the original version of the file using the Amazon S3 Management Console.

However, if you try to access the older version of the sample-file.txt file using the object URL link, you will receive an access denied message. This message is expected because the bucket policy you created in the previous task allows permission to access only the latest version of the object. In order to access a previous version of the object, you need to update your bucket policy to include the **s3:GetObjectVersion** permission. The following bucket policy example includes the additional **s3:GetObjectVersion** action that allows you to access the older version using the link. You do not need to update your bucket policy with this example to complete this lab. You can try to do this on your own after you complete the task.

```
{
  "Id": "Policy1557511288767",
  "Version": "2012-10-17",
  "Statement": [
  {
    "Sid": "Stmt1557511286634",
    "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:s3:::reportbucket987987/*",
    "Principal": "*"
  }
  ]
}
```

101. Return to the **AWS Management Console** tab, and choose the link for the bucket name at the upper left to return to the bucket overview tab.
102. Locate the **Show versions** option, and toggle the button to on to show the versions.

Now you can view the available versions of each object and identify which version is the latest. Notice that the **new-report.png** object has only one version. The version ID is **null** because the object was uploaded before versioning was enabled on this bucket.

Also notice that you can now choose the version name link to navigate directly to that version of the object in the console.

103. Next to **Show versions**, toggle the button to off to return to the default object view.
104. Select the check box to the left of the **sample-file.txt**.
105. With the object selected, choose **Delete**
106. The **Delete objects** page appears.
107. At the bottom, in the **Delete objects?** section, enter `delete` and choose the **Delete objects** button to confirm deletion of the object.
108. In the upper right of the page, choose **Close** to return to the bucket overview.

The **sample-file.txt** object is no longer displayed in the bucket. However, if the object is deleted by mistake, you can use versioning to recover it.

109. Locate the **Show versions** option, and toggle the button to on to show the versions.

Notice that the **sample-file.txt** object is displayed again, but the most recent version is a **Delete marker**. The two previous versions are also listed. If versioning has been enabled on the bucket, objects are not immediately deleted. Instead, Amazon S3 inserts a delete marker, which becomes the current object version. The previous versions of the object are not removed. Refer to the **Additional resources** section at the end of the lab for links to more information about versioning.

110. Select the check box for the version of the **sample-file.txt** object with the **Delete marker**.
111. With the object selected, choose **Delete**
112. The **Delete objects** window appears.
113. At the bottom in the **Permanently delete objects?** section, enter `permanently delete` and choose the **Delete objects** button to confirm deletion of the object.
114. On the upper right of the page, choose **Close** to return to the bucket overview.

115. Next to **Show versions**, toggle the button to off to return to the default object view.

Notice that the **sample-file.txt** object has been restored to the bucket. Removing the delete marker has effectively restored the object to its previous state. Refer to the **Additional resources** section at the end of the lab for links to more information about undeleting S3 objects.

Next, you delete a specific version of the object.

116. To delete a specific version of the object, locate the **Show versions** option, and toggle the button to on to show the versions.

You should see two versions of the **sample-file.txt** object.

117. Select the check box for the latest version of the **sample-file.txt** object.
118. With the object selected, choose **Delete**
119. The **Delete objects** window appears.
120. At the bottom in the **Permanently delete objects?** section, enter `permanently delete` and choose the **Delete objects** button.
121. On the upper right of the page, choose **Close** to return to the bucket overview.

Notice that there is now only one version of the **sample-file.txt** file. When deleting a specific version of an object, no delete marker is created. The object is permanently deleted. Refer to the **Additional resources** section at the end of the lab for links to more information about deleting object versions in Amazon S3.

122. Next to **Show versions**, toggle the button to off to return to the default object view.
123. Choose the **sample-file.txt** file name. The sample-file.txt overview page opens.
124. Copy the **Object URL** link displayed at the bottom of the window.
125. In a new browser tab, paste the link into the address field, and then press Enter.

The browser page displays the text of the original version of the **sample-file.txt** object.

Summary

You have successfully created an S3 bucket for your company to use to store report data from your EC2 instance. You created a bucket policy so that the EC2 instance can PutObjects and GetObject from the reportbucket, and you successfully tested uploading and downloading files from the EC2 instance to test the bucket policy. You have enabled versioning on the S3 bucket to protect against accidental object deletion. You have successfully completed the configuration for your EC2 reportbucket. Congratulations!

Submitting your work

126. At the top of these instructions, choose **Submit** to record your progress and when prompted, choose **Yes**.

Tip: If you previously hid the terminal in the browser panel, expose it again by checking the Terminal checkbox in the top right. This will ensure that the lab instructions remain visible after you choose Submit.

127. If the results don't display after a couple of minutes, return to the top of these instructions and choose Grades

Tip: You can submit your work multiple times. After you change your work, choose **Submit** again. Your last submission is what will be recorded for this lab.

Lab complete

Congratulations! You have completed the lab.

128. Choose End Lab at the top of this page, and then select **Yes** to confirm that you want to end the lab.

A panel indicates that *DELETE has been initiated... You may close this message box now.*

129. Select the **X** in the top right corner to close the panel.

Additional resources

- Amazon S3 at http://aws.amazon.com/s3
- Amazon S3 training at https://www.aws.training/LearningLibrary?&search=Amazon%20Simple%20Storage%20Service&tab=view_all
- Editing Object Permissions at http://docs.aws.amazon.com/AmazonS3/latest/UG/EditingPermissionsonanObject.html
- Amazon S3 bucket naming rules at https://docs.aws.amazon.com/AmazonS3/latest/dev//BucketRestrictions.html#bucketnamingrules
- Amazon S3 block public access at https://docs.aws.amazon.com/AmazonS3/latest/user-guide/block-public-access.html
- Amazon Resource Names (ARNs) and AWS Service Namespaces documentation at https://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html
- AWS JSON Policy Elements documentation at https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_elements_principal.html
- Actions, Resources, and Condition Keys for Amazon S3 at https://docs.aws.amazon.com/IAM/latest/UserGuide/list_amazons3.html
- Amazon S3 Versioning at https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html

- Undelete objects in Amazon S3 at https://docs.aws.amazon.com/AmazonS3/latest/user-guide/undelete-objects.html
- Deleting object versions in Amazon S3 at https://docs.aws.amazon.com/AmazonS3/latest/dev/DeletingObjectVersions.html
- Amazon S3 Versioning cost considerations at https://aws.amazon.com/s3/faqs/
- AWS Systems Manager Session Manager at https://docs.aws.amazon.com/systems-manager/latest/userguide/session-manager.html

For more information about AWS Training and Certification, see https://aws.amazon.com/training/.

Your feedback is welcome and appreciated.

If you would like to share any suggestions or corrections, please provide the details in our AWS Training and Certification Contact Form at https://support.aws.amazon.com/#/contacts/aws-training.