Working with Amazon CloudFront for Dynamic Content Acceleration

**SPL-34 - Version 4.1.19**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

Amazon CloudFront is a web service that accelerates the delivery of your content to your users through Amazon CloudFront’s global network of edge locations. In this lab you learn how to distribute a dynamic website with Amazon CloudFront. You start the lab by running a dynamic forum website using the *Bitnami based phpBB (PHP Bulletin Board)* in Amazon EC2. Then you will create an Amazon CloudFront web distribution to deliver and receive your static and dynamic content from 70+ edge locations.

The majority focus of this lab is on creating an Amazon CloudFront distribution with the AWS management console. You investigate the features of Amazon CloudFront that enable different types of dynamic content to be accelerated to the end user and back to the origin web server. At the end of the lab, you will be able to accelerate your entire website, and improve your user’s experience in a fast, reliable, and scalable way.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should be familiar with basic navigation of the AWS Management Console and some basic networking concepts.

TOPICS COVERED

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

* Create an Amazon CloudFront distribution using the AWS Management Console.
* Customize your Amazon CloudFront distribution.
* Log into your Dynamic Application.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Additional information:** Where to find more information.
* **Caution:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **Consider:** **Consider:** A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand.
* **Copy edit:** A time when copying a command, script, or other text to a text editor (to edit specific variables within it) might be easier than editing directly in the command line or terminal.

DURATION

This lab requires *90* minutes to complete.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING:** **Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

WHAT IS AMAZON CLOUDFRONT?

Amazon CloudFront is a content delivery web service. It integrates with other Amazon Web Services to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no commitments.

Amazon CloudFront can be used to deliver your entire website, including dynamic, static, streaming, and interactive content using a global network of edge locations. Requests for your content are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

Amazon CloudFront is optimized to work with other Amazon Web Services, including Amazon Simple Storage Service (Amazon S3), Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Load Balancing, and Amazon Route 53. Amazon CloudFront also works seamlessly with any non-AWS origin server, which stores the original, definitive versions of your files.

Like other Amazon Web Services, there are no contracts or monthly commitments for using Amazon CloudFront – you pay only for as much or as little content as you actually deliver through the content delivery service.

**Task 1: Create an Amazon CloudFront web distribution**

In this task, you create an Amazon CloudFront web distribution.

An Amazon EC2 instance running the Bitnami phpBB Bulletin Board has been started as part of the lab setup. You need to wait for this instance to become available before creating a CloudFront distribution.

1. **Copy-edit:** Copy the **ForumURL** value displayed to the left of these instructions.

This *ForumURL* points to the Bulletin Board system running on the Amazon EC2 instance. You use it in an upcoming step.

1. At the top of the AWS Management Console, in the search bar, search for and choose

CloudFront

.

1. Choose **Create a CloudFront distribution**.

On the **Create distribution** page, configure the following options.

1. In the **Origin** section:

* For **Origin domain**, paste the value of **ForumURL** you copied earlier.
* For **Protocol**, choose **HTTP only**.
* For **HTTP port**, choose **80**.

1. In the **Default cache behavior** section:

* For **Viewer protocol policy:** Select **HTTP and HTTPS** - This will allow end users to use to access your objects.
* For **Allowed HTTP methods:** Select **GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE** - This will allow CloudFront to handle traffic both to and from the end user.

1. In the **Cache key and origin requests** section:

* Select **Legacy cache settings**.
* For **Query strings**, select **All** from the dropdown menu - The site uses query strings to help customize the dynamic content on the page. You need to forward these strings onto the origin server as well.
* For **Cookies**, select **All** from the dropdown menu - The site uses cookies to customize the dynamic content on the page. Thus, you need to forward the cookies to the origin server on Amazon EC2.
* For **Object caching**, select **Customize**.
* For **Minimum TTL**, enter

0

 if not already set - This will set the minimum time an object can stay cached to be 0 (not-cached), which is necessary for dynamic content. Otherwise, the default cache time is 24 hours.

**Note:** The default behavior applies to everything (\*), unless you set specific custom patterns, which you will do later in the lab.

1. In the **Web Application Firewall (WAF)**:

* Select **Do not enable security protections**.

1. In the **Settings** section:

* For **Price class**, select **Use all edge locations (best performance)**.

By default, Amazon CloudFront serves your objects from edge locations in all Amazon CloudFront regions.

* For **Alternate domain name (CNAME)**, leave empty.

**Consider:** In the future you can specify one or more domain names that you want to use for URLs for your objects instead of the domain name that Amazon CloudFront assigns when you create your distribution. For example, if you want the URL for the **object/images/image.jpg** to look like <http://www.example.com/images/image.jpg> instead of <http://d111111abcdef8.cloudfront.net/images/image.jpg>, you would create a **CNAME** for [www.example.com](http://www.example.com/). You can create up to 10 CNAMEs per distribution, and with Amazon Route 53 you can configure your “zone apex” (root domain such as example.com without the www) to point to the same Amazon CloudFront distribution.

 For more information, see *Routing traffic to an Amazon CloudFront distribution by using your domain name* in the [Additional resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2Fspl-34%3A4.1.19-29648ae8/en-US#additional_resources) section.

**Consider:** If you add a CNAME for [www.example.com](http://www.example.com/) to your distribution, you also need to create (or update) a CNAME record with your DNS service to route queries for [www.example.com](http://www.example.com/) to the CloudFront domain name (eg *d123456789.cloudfront.net*). You must have permission to create a CNAME record with the DNS service provider for the domain. Typically, this means that you own the domain, but you may also be developing an application for the domain owner.

 For more information about CNAMEs, see *Using custom URLs by adding alternate domain names (CNAMEs)* in the [Additional resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2Fspl-34%3A4.1.19-29648ae8/en-US#additional_resources) section.

* For **Custom SSL certificate**, leave empty.

**Consider:** This certificate is shared across all of *cloudfront.net* and requires the use of the *cloudfront.net* domain. Amazon CloudFront also supports your own custom SSL Certificate with Vanity SSL. This allows you to get all of the benefits of Amazon CloudFront and SSL while using your own domain name.

 For more information, see *Amazon CloudFront Custom SSL* in the [Additional resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2Fspl-34%3A4.1.19-29648ae8/en-US#additional_resources) section.

* For **Default root object**, leave empty.

**Consider:** In the future you would set this to the object that you want Amazon CloudFront to request from your origin (for example, *index.html*) when a viewer requests the root URL of your distribution (<http://www.example.com/>) instead of an object in your distribution (<http://www.example.com/product-description.html>). Specifying a default root object avoids exposing the contents of your distribution.

* For **Standard logging**, select **Off**.

**Consider:** If you want Amazon CloudFront to log information about each request for an object and store the log files in an Amazon S3 bucket, select **On**, specify the bucket and an optional prefix for the names of the log files. There is no extra charge to enable logging, but you accrue the usual Amazon S3 charges for storing and accessing the files.

1. Choose **Create distribution**.

**Expected output:**

 Successfully created new distribution. message is displayed on top of the screen.

1. At the top of the page, in the navigation breadcrumbs, choose the **Distributions** link to return to the **CloudFront Distributions** page.

**Note:** The distribution takes approximately *15 minutes* to deploy across Amazon CloudFront’s 70+ global endpoint locations. When your CloudFront distribution is ready, the value of the *Status* column for your distribution will change to  Enabled.

1. While the CloudFront distribution is being deployed, proceed with the next task.

**Congratulations!** You have successfully created an Amazon CloudFront web distribution.

**Task 2: Create a custom error page**

In this task, you create a custom error page. Also known as a **404 Error Page**, this will be displayed if a guest tries to access a page that does not exist.

1. **Copy-edit:** Use a text editor on your local computer to create a file containing the following code:

<html><body>

<h2>This is not the page you are looking for!</h2>

</body></html>

1. Save it with the filename of

Custom404ErrorPage.html

.

1. At the top of the AWS Management Console, in the search bar, search for and choose

S3

.

1. Select the **text link** for the S3 bucket name that starts with **error404bucket**.
2. Choose **Upload**.
3. Choose **Add files** .
4. Browse to and select the **Custom404ErrorPage.html** file that you created previously.
5. Choose **Upload**.

**Expected output:**

**Upload succeeded** message is displayed.

1. At the top right-hand corner of the page, choose **Close**.
2. Choose the text link to the **Custom404ErrorPage.html** file.
3. Choose **Actions** and select **Make public using ACL** from the dropdown menu.
4. Choose **Make public**.

**Expected output:**

**Successfully edited public access** message is displayed on top of the screen.

1. At the top right-hand corner of the page, choose **Close**.

You use this bucket later in the lab.

**Congratulations!** You have successfully created a custom error page that will be displayed if a guest tries to access a page that does not exist.

**Task 3: Customize your Amazon CloudFront web distribution**

In this task, you customize the settings of your CloudFront distribution to further improve performance and customization of the site.

1. At the top of the AWS Management Console, in the search bar, search for and choose

CloudFront

.

1. Choose the **text link** for the **distribution ID** that was created earlier.

**Note:** You set the default cache behavior to have a minimum TTL (minimum time to cache an object) to 0 because of the dynamic components of the website. There are many static objects in the website, such as gif images, that do not change very often, if at all. These static objects can be cached for a very long time and the site would perform better if these objects were continually available in cache. With Amazon CloudFront you can override the default cache behavior for gif files by creating an additional cache behavior.

1. Choose the **Behaviors** tab.
2. Choose **Create behavior**.

On the **Create behavior** page, configure the following options.

1. In the **Settings** section:

* For **Path pattern** enter

\*.gif

.

This is the pattern of the objects you want to create an overriding cache behavior for.

* For **Origin and origin groups**, select the existing origin from the dropdown menu.

1. In the **Cache key and origin requests** section:

* Select **Legacy cache settings**.
* For **Object caching**, select **Customize**.
* In the **Minimum TTL** textbox, set the value to

86400

.

This is the number of seconds in 24 hours. You could set this to be shorter or longer, depending on the type of content. In this case the caches will check back with the origin at most once a day to refresh the gifs that are being requested.

When setting a longer TTL, the objects in cache will not be updated right away when an object is changed at the origin.

 For more information, see *Managing how long content stays in the cache (expiration)* in the [Additional resources](https://labs.skillbuilder.aws/sa/lab/arn%3Aaws%3Alearningcontent%3Aus-east-1%3A470679935125%3Ablueprintversion%2Fspl-34%3A4.1.19-29648ae8/en-US#additional_resources) section .

1. Navigate to the bottom of the page and choose **Create behavior**.

**Expected output:**

 Successfully created new cache behavior \*.gif. message is displayed on top of the screen.

**Consider:** When a user gets an error, you would like to provide a customized error page. You choose the 404 (page not found) error, but it is possible to serve custom pages for many different errors.

While still in the Distribution Settings window you first need to add an origin. You are going to add an Amazon S3 origin to serve the error page. This way if the Amazon EC2 Instance were to stop responding, you would still be able to serve the custom error page.

1. Choose the **Origins** tab.
2. Choose **Create origin**.

On the **Create origin** page, configure the following options.

1. In the **Settings** section:

* For **Origin domain**, select the **S3 bucket** with the name that starts with **error404bucket**.
* For **Origin access**, select **Legacy access identities**.
* For **Origin access identity**, choose **Create new OAI**.
* On the **Create new OAI** pop-up window, choose **Create**.

1. Choose **Create origin**.

**Expected output:**

 Successfully created origin. message is displayed at the top of the screen.

Next, create another **Behavior** that defines when to use this origin.

1. Choose the **Behaviors** tab.
2. Choose **Create behavior**.

On the **Create behavior** page, configure the following options.

1. In the **Settings** section:

* For **Path pattern** enter

/Custom\*ErrorPage.html

.

This tells CloudFront the **path** inside your **Origin bucket** to find the **custom error pages** you want to serve in response to **error conditions**.

* For **Origin and origin groups**, select the **bucket name** that starts with **error404bucket**.

1. Navigate to the bottom of the page and choose **Create behavior**.

**Expected output:**

 Successfully created new cache behavior /Custom\*ErrorPage.html. message is displayed at the top of the screen.

Finally, you add the **Error Page**.

1. Choose the **Error pages** tab.
2. Choose **Create custom error response**.

On the **Create custom error response** page, configure the following options.

1. In the **Error response** section:

* For **HTTP error code**, select **404: Not Found** from the dropdown menu.
* For **Customize error response**, select **Yes**.
* For **Response page path**, enter

/Custom404ErrorPage.html

 in the textbox (This is the file you created earlier and uploaded to the S3 bucket).

* For **HTTP Response code** select **404: Not Found**, if not already selected.

1. Choose **Create custom error response**.

**Expected output:**

 Successfully created 404 error page. message is displayed at the top of the screen.

**Congratulations!** You have successfully customized and configured the settings for your Amazon CloudFront distribution. Your content is ready to be accessed throughout the world on Amazon CloudFront’s edge network. To review, Amazon CloudFront knows where your Amazon EC2 origin server is, and will serve both static and dynamic content to and from the end user.

**Task 4: Obtain the password for your phpBB application**

To be able to configure the phpBB forum, you will need to obtain *user* password.

1. At the top of the AWS Management Console, in the search bar, search for and choose **EC2**.
2. In the left navigation pane, choose **Instances**.
3. Select the  **BitNami** instance.
4. Choose **Actions** > **Monitor and troubleshoot** > **Get system log**.
5. Search for

Setting Bitnami application password to

 in the system log.

1. **Copy-edit:** Copy the password to a text editor.

**Task 5: Configure the phpBB website**

Before you go and explore the full functionality of your newly created PHP Forum website, you need to make one configuration update to the website.

1. **Copy-edit:** Copy the value of **ForumURL** shown to the left of these instructions and paste it into a new browser tab.
2. Login using the following credentials:

* **Username:**

user

* **Password:** Paste the value of the password that you copied to your text editor earlier.

1. Once logged in, choose the  **ACP** (Administrator Control Panel) link at the top of the page.

You will be prompted to log in again.

1. Use the same credentials to log in to the admin section.

Once you login, the browser opens the **Administration Control Panel**.

**Note:** If presented, you can safely ignore the *Your phpBB installation is not up to date.* warning message since it is not relevant for this lab.

1. Towards the bottom of the left navigation pane, locate the **SERVER CONFIGURATION** section and choose the **Server settings** link.
2. From the **Server settings** page, navigate to the bottom of the page to the **Server URL settings** section.
3. Set the **Force server URL settings** option to **Yes**.
4. Set the **Server protocol:** to

http://

.

1. Without closing the current browser tab, return to the AWS Management Console browser tab opened to the **CloudFront Distributions** page.
2. Choose the **General** tab.
3. **Copy-edit:** Copy the value of **Distribution domain name** (it should look similar to: *d4b6v3lno3nv6.cloudfront.net*).

**Note:** If you copy the value from the  copy icon, it will have

https://

 prepended to the domain name. Do **NOT** use the

https://

 text at the front of the domain name. Cloudfront will use it’s own self-signed certificate and convert the URL from http to https.

1. Return to the **phpBB Forum** browser tab.
2. In the **Domain name** field, paste the **Amazon CloudFront domain name** that you just copied less the

https://

 in the beginning.

**Example:**

Domain name: d4b6v3lno3nv6.cloudfront.net

1. Choose **Submit.**.

**Expected output:**

Configuration updated successfully. message is displayed.

Since the IP that sends the request changes due to using CloudFront, you need to disable the security setting that performs **Session IP validation**.

1. From the navigation menu on the left, in the **SERVER CONFIGURATION** section, choose **Security settings**.
2. Update the setting for **Session IP validation** to **None**.
3. Choose **Submit** .

**Expected output:**

Configuration updated successfully. message is displayed.

**Congratulations!** You have successfully made the required configuration update to the phpBB website.

**Task 6: Test the website**

Once your CloudFront distribution is fully deployed, you can test the cached website.

1. Return to the AWS Management Console browser tab opened to the **CloudFront Distributions** page.
2. In the left navigation pane, choose **Distributions**.

**Note:** If Distributions is not visible, choose the **Navigation** icon  and then select **Distributions** from the list.

1. Wait until the **Status** for your CloudFront distribution is  Enabled.

**Note:** Your distribution might take up to 15 minutes to be deployed. Choose the refresh  option as required.

1. Choose the link to your CloudFront distribution to view its settings.
2. **Copy-edit:** Copy the **Distribution domain name** to your text editor (it should look similar to: *https://d4b6v3lno3nv6.cloudfront.net*) and Paste it into a new browser tab.

The browser returns the *phpBB website* and this time you are accessing the website via **CloudFront**. Notice how it switched the URL from http to https. It uses a self-signed certificate which is why you see the *Not secure* message at the start of the URL. You can ignore the *Not secure* notification. In a non-lab environment, you would use a signed-certificate.

1. Login again using the following credentials:

* Username:

user

* Password: Paste the value of the password that you copied to your text editor earlier.

Feel free to explore the site.

You can create topics, post on topics, etc. All of the content you generate is going through Amazon CloudFront and being accelerated to and from the Amazon EC2 origin with optimized routing and persistent connections. And remember, you set up a caching rule for \*gifs. Thus, all of the small gif images that you see across the site are being served from the Amazon CloudFront edge servers without the request having to travel back to the Amazon EC2 origin.

1. Also, test out your custom error page by navigating to a location that doesn’t exist, such as **https://d4b6v3lno3nv6.cloudfront.net/no\_page.html**.

At this point, you should see the following message for the custom error page.

**Expected output:**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*

**\*\*\*\*** This is OUTPUT ONLY. **\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*

This is not the page you are looking for!

**Note:** If you receive a temporary redirect error, please wait a few minutes and then try again.

**Consider:** To review, Amazon CloudFront is using all of the following features to accelerate and customize your website:

**Multiple Cache Behaviors:** A cache behavior is the set of rules you configure for a given URL pattern based on file extensions, file names, or any portion of a URL path on your website (e.g. \*.gif). You can configure multiple cache behaviors for your web distribution. Amazon CloudFront will match incoming viewer requests with your list of URL patterns, and if there is a match, the service will honor the cache behavior you configure for that URL pattern. Each cache behavior can include the following Amazon CloudFront configuration values: origin server name, viewer connection protocol, minimum expiration period, query string parameters, cookies, and trusted signers for private content.

**Post & Put Support:** You can accelerate your user input (HTTP PUT,POST, etc) to your origin servers through your Amazon CloudFront web distribution. Origin servers can be an AWS resource, such as Amazon EC2, Elastic Load Balancing, or *a custom origin server outside of AWS*. Amazon CloudFront will quickly deliver content to the origin using the same optimized routing and persistent connections that the download content uses.

**Zero TTL:** Amazon CloudFront uses the expiration period you set on your files (through cache control headers) to determine whether it needs to check the origin for an updated version of the file. If you expect that your files will change frequently, you can set a short expiration period on the file. Amazon CloudFront accepts expiration periods as short as 0 seconds (in which case Amazon CloudFront will revalidate each viewer request with the origin). Amazon CloudFront also honors special cache control directives such as private, no-store, etc.; these are often useful when delivering dynamic content that may not be cached at the edge.

**Query String Parameter Forwarding:** Query string parameters are often used to return customized content generated by a script running on the origin server. You can optionally configure query strings to be forwarded to the origin servers and be included in the unique identity of the cached object. This feature can be enabled separately for each unique cache behavior you configure. Query string parameters can thus help you customize your web pages for each viewer while still taking advantage of the performance and scale benefits offered by caching content at Amazon CloudFront edge locations.

**HTTP Cookie Forwarding:** Amazon CloudFront supports delivery of dynamic content that is customized or personalized using HTTP cookies. To use this feature, you specify whether you want Amazon CloudFront to forward some or all of your cookies to your custom origin server. Amazon CloudFront then considers the forwarded cookie values when identifying a unique object in its cache. This way, your end users get both the benefit of content that is personalized just for them with a cookie and the performance benefits of Amazon CloudFront.

**Custom Error Pages:** Amazon CloudFront allows you to serve error pages with your own branding and content. For example, you can now help your visitors navigate to other sections of your website when they request an invalid URL, or you can configure a static page to serve as a fallback for failure of an origin web server.

**Congratulations!** You have successfully completed Amazon CloudFront Dynamic Content Acceleration. You’ve learned how to create and configure an Amazon CloudFront web distribution to deliver the static and dynamic components of your website to your global audience in a fast, reliable, and scalable way.

**Conclusion**

**Congratulations!** You now have successfully:

* Created an Amazon CloudFront distribution using the AWS Management Console.
* Customized your Amazon CloudFront distribution.
* Logged in to your Dynamic Application.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [Amazon CloudFront](http://aws.amazon.com/cloudfront)
* [Bitnami phpBB](http://bitnami.com/stack/phpbb)
* [Choosing the price class for a CloudFront distribution](http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/PriceClass.html)
* [Routing traffic to an Amazon CloudFront distribution by using your domain name](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-cloudfront-distribution.html)
* [Using custom URLs by adding alternate domain names (CNAMEs)](http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/CNAMEs.html)
* [Amazon CloudFront Custom SSL](http://aws.amazon.com/cloudfront/custom-ssl-domains/)
* [Managing how long content stays in the cache (expiration)](http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Expiration.html)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).