Hands-on with AWS SAM

**SPL-BE-200-DVHSAM-1 - Version 1.0.2**

© 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**

This lab demonstrates how to use AWS Serverless Application Model (AWS SAM) to create an AWS CloudFormation stack. The stack deploys the necessary resources and dependencies for the grid-maker application.

To start, you create a CloudFormation stack by using SAM, which will deploy the necessary resources and dependencies for the grid-maker application. These resources include an Amazon API Gateway endpoint, AWS Lambda functions, two Amazon Simple Storage Service (Amazon S3) buckets, and an Amazon DynamoDB table. AWS SAM automates the deployment process, making it easier to create and manage the necessary resources.

After the resources are deployed, you test the functionality of the application by invoking the API that was created in Amazon API Gateway. The API acts as a frontend for the Lambda functions, which are started by the API based on the request that’s received. These Lambda functions are responsible for performing various tasks, such as copying images to the S3 bucket, writing uniqueGridId data to the DynamoDB table, creating a new grid image, and generating a presigned URL to view the grid image.

This lab is designed to challenge developers who are looking to test their abilities. You are given high-level steps to perform, and you have the opportunity to identify the solution on your own. If you get stuck along the way, this lab includes both hints to help you and a solution.

OBJECTIVES

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

* Use AWS SAM to build a deployment package.
* Use AWS SAM to automate the creation of the resources used for the application.
* Run the API to create both the grid image and an S3 presigned URL.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab:

* Familiarity with the basic navigation of the AWS Management Console.
* Versed in editing and running scripts by using an AWS Cloud9 code editor and terminal.
* A basic understanding and familiarity with API Gateway, AWS SAM, Lambda, and CloudFormation.
* Previous experience with AWS services and serverless computing is helpful, but isn’t required.

DURATION

This lab requires *60* minutes to complete.

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:

* **Command:** A command that you must run.
* **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.
* **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.

**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.

**Task 1: Prepare the application for deployment**

In this task, you connect to the AWS Cloud9 environment that’s provisioned as part of this lab. Then, you issue the *aws sam build* command to process the AWS SAM template file, application code, and any applicable language-specific files and dependencies. The command also copies build artifacts in the format and location that are expected for subsequent steps in your workflow.

TASK 1.1: CONNECT TO THE AWS CLOUD9 IDE

1. From the **Lab Information** section to the left of these instructions, copy the **Cloud9Environment** URL link and into a new browser, tab paste the URL.

The browser takes you to the AWS Cloud9 environment that you use during this lab.

You don’t need the **Cloud9 Welcome screen** or any of the other default tabs that appear when you first launch **AWS Cloud9**.

1. To close each tab, choose the **X**.

This section of the IDE is where you update various files throughout this lab.

**Consider:** You are working in another AWS Cloud9 environment that’s similar to the previous lab. The only difference is the application files that you see in the file tree. If you need a refresher, take a moment to familiarize yourself with the **AWS Cloud9** IDE interface by expanding the *AWS Cloud9 review* section.

**AWS Cloud9 review**

TASK 1.2: PREPARE THE APPLICATION FOR DEPLOYMENT BY USING AWS SAM

For this particular task, you use the *sam build* command. With this command, you can use the *requirements.txt* file that outlines the dependencies required for the application. By running the *sam build* command, you can install all these dependencies and create a .zip file that includes both your code and the necessary dependencies. This .zip file can then be used with the Lambda functions.

1. Open the **/api-backend-sam/template.yaml** file.

Review the template’s contents to get an idea of what resources will be created when it’s used with the AWS SAM commands.

1. **Command:** To create the deployment package by using the

sam build

 command, run the following commands:

cd ~/environment/api-backend-sam

sam build

**Expected output:**

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

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

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

SAM CLI now collects telemetry to better understand customer needs.

You can OPT OUT and disable telemetry collection by setting the

environment variable SAM\_CLI\_TELEMETRY=0 in your shell.

Thanks for your help!

Learn More: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-telemetry.html

Building codeuri: /home/ec2-user/environment/api-backend-sam/add\_image runtime: python3.9 metadata: {} architecture: x86\_64 functions: AddImageFunction

requirements.txt file not found. Continuing the build without dependencies.

Running PythonPipBuilder:CopySource

Building codeuri: /home/ec2-user/environment/api-backend-sam/generate\_grid runtime: python3.9 metadata: {} architecture: x86\_64 functions: GenerateGridFunction

Running PythonPipBuilder:ResolveDependencies

Running PythonPipBuilder:CopySource

Build Succeeded

Built Artifacts : .aws-sam/build

Built Template : .aws-sam/build/template.yaml

Commands you can use next

=========================

[\*] Validate SAM template: sam validate

[\*] Invoke Function: sam local invoke

[\*] Test Function in the Cloud: sam sync --stack-name {stack-name} --watch

[\*] Deploy: sam deploy --guided

SAM CLI update available (1.76.0); (1.57.0 installed)

To download: https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-install.html

TASK 1.3: DEPLOY THE APPLICATION BY USING AWS SAM

To deploy your serverless application to AWS with CloudFormation, you can use the **sam deploy --guided** command in this task. This command initiates the creation of a CloudFormation stack that contains all the resources that your application needs. Moreover, it uploads your code and dependencies to an S3 bucket, which is managed by AWS SAM on your behalf. This streamlines the deployment process and helps get your application up and running quickly.

**Note:** The *sam deploy --guided* flag is an option that you can use to interactively configure your deployment parameters. It prompts you for information such as the stack name, Region, capabilities, tags, and parameters. It also creates a *samconfig.toml* file that stores your settings and helps speed up future deployments.

1. **Command:** To create the resources that make up your application, run the following **sam deploy --guided** command:

sam deploy --guided

You are prompted with a series of questions.

1. Respond to the question with the following options:

* **Command:** For the **Stack Name [sam-app]:**, enter

sam-lab

 and press **Enter**.

**WARNING:** The *stack name* needs to be set to *sam-lab* because the lab policies are created based on this name. If you use a different name, all lab steps won’t work properly.

* **Command:** For **AWS Region [us-west-2]:**, accept the default value by pressing **Enter**.
* **Command:** For **Confirm changes before deploy [y/N]:** accept the default answer by pressing **Enter**.
* **Command:** For **Allow SAM CLI IAM role creation [Y/n]:**, accept the default answer by pressing **Enter**.
* **Command:** For **Disable rollback [y/N]:**, accept the default answer by pressing **Enter**.
* **Command:** For **AddImageFunction has no authentication. Is this okay? [y/N]:**, enter

y

.

* **Command:** For **GenerateGridFunction has no authentication. Is this okay? [y/N]:**, enter

y

.

* **Command:** For **Save arguments to configuration file [Y/n]:**, accept the default answer by pressing **Enter**.
* **Command:** For **SAM configuration file [samconfig.toml]:**, accept the default value by pressing **Enter**.
* **Command:** For **SAM configuration environment [default]:**, accept the default value by pressing **Enter**.

**Expected output:**

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

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

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

Configuring SAM deploy

======================

Looking for config file [samconfig.toml] : Found

Reading default arguments : Success

Setting default arguments for 'sam deploy'

=========================================

Stack Name [sam-app]: sam-lab

AWS Region [us-west-2]:

#Shows you resources changes to be deployed and require a 'Y' to initiate deploy

Confirm changes before deploy [y/N]:

#SAM needs permission to be able to create roles to connect to the resources in your template

Allow SAM CLI IAM role creation [Y/n]:

#Preserves the state of previously provisioned resources when an operation fails

Disable rollback [y/N]:

AddImageFunction may not have authorization defined, Is this okay? [y/N]: y

GenerateGridFunction may not have authorization defined, Is this okay? [y/N]: y

Save arguments to configuration file [Y/n]:

SAM configuration file [samconfig.toml]:

SAM configuration environment [default]:

Looking for resources needed for deployment:

Managed S3 bucket: aws-sam-cli-managed-default-samclisourcebucket-gxetfraqaagc

A different default S3 bucket can be set in samconfig.toml

Saved arguments to config file

Running 'sam deploy' for future deployments will use the parameters saved above.

The above parameters can be changed by modifying samconfig.toml

Learn more about samconfig.toml syntax at

https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-sam-cli-config.html

Uploading to sam-lab/b9f19ea8c53248d010c396d15857686e 675 / 675 (100.00%)

Uploading to sam-lab/7edc7d688bab221d9f30c2a89379e0c0 14391550 / 14391550 (100.00%)

Deploying with following values

===============================

Stack name : sam-lab

Region : us-west-2

Confirm changeset : False

Disable rollback : False

Deployment s3 bucket : aws-sam-cli-managed-default-samclisourcebucket-gxetfraqaagc

Capabilities : ["CAPABILITY\_IAM"]

Parameter overrides : {}

Signing Profiles : {}

Initiating deployment

=====================

Uploading to sam-lab/fdd459a9e33231559f8a3fcb7a29b6bb.template 2936 / 2936 (100.00%)

Waiting for changeset to be created..

CloudFormation stack changeset

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Operation LogicalResourceId ResourceType Replacement

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

+ Add AddImageFunctionAddImagePermission AWS::Lambda::Permission N/A

+ Add AddImageFunction AWS::Lambda::Function N/A

+ Add ApiGatewayApiProdStage AWS::ApiGatewayV2::Stage N/A

+ Add ApiGatewayApi AWS::ApiGatewayV2::Api N/A

+ Add DestinationBucket AWS::S3::Bucket N/A

+ Add GenerateGridFunctionAddImagePermission AWS::Lambda::Permission N/A

+ Add GenerateGridFunction AWS::Lambda::Function N/A

+ Add GridObjects AWS::DynamoDB::Table N/A

+ Add SourceBucket AWS::S3::Bucket N/A

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Changeset created successfully. arn:aws:cloudformation:us-west-2:111111111111:changeSet/samcli-deploy1679330596/ff5a5b11-af27-4658-b8cb-23c8a57da0b6

2023-03-20 16:43:27 - Waiting for stack create/update to complete

CloudFormation events from stack operations (refresh every 0.5 seconds)

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ResourceStatus ResourceType LogicalResourceId ResourceStatusReason

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CREATE\_IN\_PROGRESS AWS::DynamoDB::Table GridObjects -

CREATE\_IN\_PROGRESS AWS::S3::Bucket SourceBucket -

CREATE\_IN\_PROGRESS AWS::S3::Bucket DestinationBucket -

CREATE\_IN\_PROGRESS AWS::S3::Bucket SourceBucket Resource creation Initiated

CREATE\_IN\_PROGRESS AWS::DynamoDB::Table GridObjects Resource creation Initiated

CREATE\_IN\_PROGRESS AWS::S3::Bucket DestinationBucket Resource creation Initiated

CREATE\_COMPLETE AWS::DynamoDB::Table GridObjects -

CREATE\_COMPLETE AWS::S3::Bucket SourceBucket -

CREATE\_COMPLETE AWS::S3::Bucket DestinationBucket -

CREATE\_IN\_PROGRESS AWS::Lambda::Function AddImageFunction -

CREATE\_IN\_PROGRESS AWS::Lambda::Function GenerateGridFunction -

CREATE\_IN\_PROGRESS AWS::Lambda::Function AddImageFunction Resource creation Initiated

CREATE\_IN\_PROGRESS AWS::Lambda::Function GenerateGridFunction Resource creation Initiated

CREATE\_COMPLETE AWS::Lambda::Function AddImageFunction -

CREATE\_COMPLETE AWS::Lambda::Function GenerateGridFunction -

CREATE\_IN\_PROGRESS AWS::ApiGatewayV2::Api ApiGatewayApi -

CREATE\_IN\_PROGRESS AWS::ApiGatewayV2::Api ApiGatewayApi Resource creation Initiated

CREATE\_COMPLETE AWS::ApiGatewayV2::Api ApiGatewayApi -

CREATE\_IN\_PROGRESS AWS::Lambda::Permission GenerateGridFunctionAddImagePermission -

CREATE\_IN\_PROGRESS AWS::ApiGatewayV2::Stage ApiGatewayApiProdStage -

CREATE\_IN\_PROGRESS AWS::Lambda::Permission AddImageFunctionAddImagePermission -

CREATE\_IN\_PROGRESS AWS::Lambda::Permission GenerateGridFunctionAddImagePermission Resource creation Initiated

CREATE\_IN\_PROGRESS AWS::Lambda::Permission AddImageFunctionAddImagePermission Resource creation Initiated

CREATE\_IN\_PROGRESS AWS::ApiGatewayV2::Stage ApiGatewayApiProdStage Resource creation Initiated

CREATE\_COMPLETE AWS::ApiGatewayV2::Stage ApiGatewayApiProdStage -

CREATE\_COMPLETE AWS::Lambda::Permission AddImageFunctionAddImagePermission -

CREATE\_COMPLETE AWS::Lambda::Permission GenerateGridFunctionAddImagePermission -

CREATE\_COMPLETE AWS::CloudFormation::Stack sam-lab -

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CloudFormation outputs from deployed stack

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Outputs

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Key Api

Description API Gateway endpoint URL

Value https://ibkgxlpzo2.execute-api.us-west-2.amazonaws.com/Prod/

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Successfully created/updated stack - sam-lab in us-west-2

1. Save the **API Gateway endpoint URL**.

You use the endpoint URL in upcoming steps.

 Congratulations! You have successfully created the application deployment files based on the application code and the dependencies to Amazon S3. You have also successfully deployed the resources for your application to CloudFormation by using AWS SAM.

**Task 2: Create the grid image by using API Gateway**

As part of this task, you use the API to generate a grid image based on images that are stored locally. You also copy them to the *source-images* S3 bucket. After the API processes all the images, a final grid image is created and saved to the *destination-images* S3 bucket. You access the final grid image by using an S3 presigned URL.

1. Return to the **AWS Cloud9** environment browser tab.
2. **Command:** Create the **uniqueGridId** variable value (which is based on the **timestamp** when the command was run) by running the following command:

uniqueGridId=`date +%s` ; echo ${uniqueGridId}

**Expected output:** Your value will differ from what is shown in this example.

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

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

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

1683051249

You now create the *baseURL* variable by using the *invoke URL* value from API Gateway (you copied and saved this URL previously). Then, you update the placeholder text with the *invoke URL* value.

1. **Command:** To create the **baseURL** variable, run the following command:

baseUrl='placeholder-for-invoke-url'

**Expected output:**

*None, unless an error occurs.*

1. **Command:** Change directories to the source folder and list the images in this folder with the following command:

cd ~/environment/api-backend-sam/source ; ls

**Expected output:**

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

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

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

image01.jpg image02.jpg image03.jpg image04.jpg image05.jpg image06.jpg image07.jpg image08.jpg image09.jpg image10.jpg image11.jpg image12.jpg image13.jpg image14.jpg image15.jpg image16.jpg

Now, you invoke the API by using the curl -X POST method to send data to it. The command sends an image file named *image01.jpg* as binary data to the *source-images* S3 bucket by using the HTTP POST method. Next, it passes the parameter named *uniqueGridId* along with the *baseUrl* value. It then stores this data in the DynamoDB table. To avoid seeing the download details the curl command, it is redirected /dev/null, effectively suppressing it. Notice the use of the JQuery command to format the output.

1. **Command:** Invoke the API Gateway with the following command:

curl -X POST --data-binary @image01.jpg "${baseUrl}/add\_image?uniqueGridId=${uniqueGridId}" 2>/dev/null | jq -r '"\nMessage: \(.message)\n\nImage\_Size: \(.image\_size)\n"'

**Expected output:**

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

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

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

Message: image saved

Image\_Size: 196799

1. **Command:** Repeat the previous step three more times for the images named **image02.jpg**, **image03.jpg**, and **image04.jpg**.

**Hint**

1. **Command:** To engage the API to create the grid image and an S3 presigned URL while formatting the output using JQuery, run the following command:

curl -X POST "${baseUrl}/generate\_grid?uniqueGridId=${uniqueGridId}" 2>/dev/null | jq -r '"\nMessage: " + .message, "\nPresigned\_URL: " + .presigned\_url, "\n"'

**Expected output:** Output has been truncated.

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

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

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

Message: built grid

Presigned\_URL: https://sam-lab-destinationbucket-1mbvr192cs50g.s3.amazonaws.com/1d2de2acad4be9b7821a6436b911d5c7.jpg?AWSAccessKeyId=ASIAVZP34UDT2M44Y7I2&Signature=4ub6pgWsAXEkU5A4tqNLAmS3SQg%3D&x-amz-security-token=IQoJb3JpZ2luX2VjEDkaCXVzLXdlc3QtMiJHMEUCIQDGdjL27laVbYUTte3o2kvDIRAk30qUNX0XWTy4mb7BqgIgGTdIaLVGSzbjm%2FFiAAHEzXU8z%2Fv%2FgPIHFJxohMUfugAqpwMIYhAEGgwzOTgzNDk2MDcxNDMiDClwWGHhCowYurS%2B%2BiqEA6XHa8byd8iNqreH7yB3N0qwUgEMhF1L4%2F1CWjqMUP5Jtz710rsPcDAKy5PQr5ejswtU%2BcbQ%2FklbjJzZL58BmNca2Di1ra2jF3a5djy8IySAoDrXmjm%2FsWlm5zEcFzYgVTsQiZuVT6EWswcsZEn3WPVkOotaOpWeKVin70VcuXqv6xoEJnWiSepX%2B%2FMczLdfociFQpA0rgyg%2BlLKcN0BMBEFWueyhni7PaVY1b3aFs3SLfbdp5c8CbWZTqSKOe4EIKfgiRCE49Chvr5VYdzHaOyUP28jhMBv8IxV3FGQqG7Z%2BPYVB9troOQG7kFz0S9NUrYA74e4miLqkLRtxkiqLiEPO4k73xqw4i08k0f8nd0Fm2w2M%2FLJT1pWZgd1dIor06%2FRmNv%2Bq2mNegs%2FPwfAIgvrkAlmyLykrFMUELDc87KxSrxNdSsXuQO4DY1%2BtSBBLkbxAYMgkUGY1ZBB7iup%2BqpMHivflZjsr0ns0RgcDj5EQJViO1LVn2Ri97SPRz41HaCR4Zow2qGZowY6nQH%2FqXKlzI3e9s2fWtm0NI%2B%2BsQlIYu%2FwcYRLDRQR3S2kIhogKjUUBd4GvAcswk5owkTsVgpnUh770GYZL5HGWoS6xRaFM%2BAgcq%2F459fOW%2Fr51bp3QI04QVtTjZZbVK0nGkvVMMEXcp7YdfKQHgR%2FyGOGyPN2DRgjxSrHYmGO9t7V9a8HSqiEBouU59rFZ3tJ9DjUPS6mp123tZzQcUa2&Expires=1684427471

1. To see the **grid-image.jpg** created from four individual images, in a new browser tab, open the **S3 presigned URL**.



**Note:** To see additional photos added to the grid image, you can continue to run the command to add more images and then run the command to generate the grid image.

**Congratulations!** You successfully engaged the API to create the grid image based on images stored in the GridBuilder DynamoDB table. Additionally, the API generated an S3 presigned URL that points to where the image is stored in the *destination-images* S3 bucket. You then viewed the *grid-image.jpg* file in a browser tab by using the *S3 presigned URL*.

**Conclusion**

 Congratulations! You now have successfully:

* Used AWS SAM to build a deployment package.
* Used AWS SAM to automate the creation of the resources used for the application.
* Run the API to create the grid image and an S3 presigned URL.

**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.

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).