Moving to AWS Lambda

**SPL-BE-200-DVMTAW-1 - Version 1.0.3**

© 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 Lambda as the compute environment to create a grid image and store it in an Amazon Simple Storage Service (Amazon S3) bucket. You then view the image in a browser tab by using an S3 presigned URL. To create a Lambda function, you need two components: a deployment package and an execution role.

The deployment package is a .zip file archive or container image that contains the code and dependencies for your function. The execution role is a role that grants permissions to the function to use AWS services, such as Amazon CloudWatch Logs and Amazon S3, for log streaming and accessing objects in S3 buckets. In this lab, the execution role has already been created as part of the lab-creation process.

To create a function, you use the

aws lambda create-function

 command. After you create the function, you can run it by using the

aws lambda invoke

 command. The lab provides step-by-step guidance on how to create and run the function, and how to view the results of the function invocation.

Overall, this lab provides a practical example of how to create a Lambda function by using the AWS Command Line Interface (AWS CLI), and how to deploy and test a serverless application on AWS by running the Lambda function as part of the API, along with other AWS services.

OBJECTIVES

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

* Create a deployment package, including the application code and Python package requirements.
* Run an AWS CLI command to create a Lambda function.
* Run an AWS CLI command to invoke a Lambda function with an event.
* Inspect Amazon CloudWatch Logs after invoking a Lambda function.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab:

* You should be familiar with basic navigation of the AWS Management Console.
* You should be comfortable editing and running scripts by using an AWS Cloud9 code editor and terminal.
* You should have a basic understanding of and familiarity with AWS Lambda.

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: Create and run a Lambda function**

In this task, you connect to the AWS Cloud9 environment. You review the application and then create a Lambda function by using the AWS CLI.

TASK 1.1: CONNECT TO AWS CLOUD9

In this task, you connect to the AWS Cloud9 environment that’s provisioned as part of this lab.

1. From the **Lab Information** section to the left of these instructions, copy the **Cloud9Environment** URL link and in 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 file throughout this lab.

**Consider:** You’re 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: REVIEW THE APPLICATION

In this task, you familiarize yourself with the application.

A **terminal pane** is at the bottom of the IDE. You can expand it up halfway to have more visibility when running commands. You can also close it and open a new terminal session from the top menu. (To open a new terminal session, choose the  icon and select **New terminal**.)

1. From the file tree, expand the **lambda-function** folder.
2. Open the **lambda\_function.py** file and review the code.

This application performs the following actions:

* Imports modules for *file*, *math*, *temp storage*, and *AWS S3 functions*
* Sets up variables for the *S3 client*, *tile size*, and the *source* and *destination* buckets
* Defines the *lambda\_handler* function to process event and context parameters from the AWS Lambda service
* *Gets*, and *lists* a number of *images* from *source bucket*
* Calculates the *grid size* based on *image count* and *tile size*
* Creates new *image object* for destination grid
* Creates a presigned URL (which you use to access an Amazon S3 object) by using the *s3.generate\_presigned\_url* operation from the AWS SDK for Python (Boto3) library

TASK 1.3: CREATE A LAMBDA FUNCTION BY USING THE AWS CLI

In this task, you install the application dependencies (which are based on a *requirements.txt* file) into the *package* directory.

1. **Command:** To perform the actions in the following list, run the following command:

* Change directories into the **lambda-function** folder
* Pull down the Python packages based on the **requirements.txt** file
* Store the packages in the **package** directory

cd ~/environment/lambda-function ; pip install -r requirements.txt --target ./package

**Expected output:**

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

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

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

Collecting pillow

Downloading Pillow-9.4.0-cp39-cp39-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (3.3 MB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 MB 50.2 MB/s eta 0:00:00

Collecting boto3

Downloading boto3-1.26.86-py3-none-any.whl (134 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.7/134.7 KB 33.1 MB/s eta 0:00:00

Collecting s3transfer<0.7.0,>=0.6.0

Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.6/79.6 KB 16.8 MB/s eta 0:00:00

Collecting botocore<1.30.0,>=1.29.86

Downloading botocore-1.29.86-py3-none-any.whl (10.5 MB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.5/10.5 MB 95.2 MB/s eta 0:00:00

Collecting jmespath<2.0.0,>=0.7.1

Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)

Collecting urllib3<1.27,>=1.25.4

Downloading urllib3-1.26.14-py2.py3-none-any.whl (140 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 140.6/140.6 KB 29.1 MB/s eta 0:00:00

Collecting python-dateutil<3.0.0,>=2.1

Downloading python\_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)

━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 247.7/247.7 KB 43.6 MB/s eta 0:00:00

Collecting six>=1.5

Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)

Installing collected packages: urllib3, six, pillow, jmespath, python-dateutil, botocore, s3transfer, boto3

Successfully installed boto3-1.26.86 botocore-1.29.86 jmespath-1.0.1 pillow-9.4.0 python-dateutil-2.8.2 s3transfer-0.6.0 six-1.16.0 urllib3-1.26.14

1. **Command:** Change to the **package** directory and compress the **dependencies** into a .zip file named **lambda-function.zip** by running the following command:

cd ~/environment/lambda-function/package ; zip -r ~/environment/lambda-function.zip .

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

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

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

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

adding: boto3-1.26.86.dist-info/WHEEL (stored 0%)

adding: boto3-1.26.86.dist-info/top\_level.txt (stored 0%)

adding: boto3-1.26.86.dist-info/RECORD (deflated 60%)

adding: boto3-1.26.86.dist-info/INSTALLER (stored 0%)

adding: boto3-1.26.86.dist-info/REQUESTED (stored 0%)

adding: bin/ (stored 0%)

adding: bin/jp.py (deflated 63%)

adding: bin/\_\_pycache\_\_/ (stored 0%)

adding: bin/\_\_pycache\_\_/jp.cpython-39.pyc (deflated 39%)

1. **Command:** To change back to the **lambda-function** directory and add the **lambda\_function.py** file to the **lambda-function.zip** file, run the following command:

cd ~/environment/lambda-function ; zip ~/environment/lambda-function.zip lambda\_function.py

**Expected output:**

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

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

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

adding: lambda\_function.py (deflated 63%)

 Congratulations! You have successfully packaged your application so that you can use it to create a Lambda function.

**Task 2: Create the Lambda function**

In this task, you run an AWS CLI command, which is based on a set of parameters, to create the Lambda function. You provide a *function name*, a *runtime environment*, a *handler name*, a *role Amazon Resource Name (ARN)* (which has been created for your use), and a *.zip* file (which you created) that contains your function code.

**Note:** The *LambdaApplicationRole* has already been created and set as an environment variable. It has a policy attached to it that grants the following permissions.

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

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

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

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"s3:GetObject",

"s3:PutObject"

],

"Resource": ["arn:aws:s3:::source-images-region-random-number"

"arn:aws:s3:::destination-images-region-random-number"

]

}

]

}

1. **Command:** To create the **grid-maker** Lambda function, run the following command:

aws lambda create-function \

--function-name grid-maker \

--runtime python3.9 \

--timeout 30 \

--handler lambda\_function.lambda\_handler \

--role $LAMBDA\_ROLE \

--zip-file fileb://~/environment/lambda-function.zip

**Expected output:**

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

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

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

{

"FunctionName": "grid-maker",

"FunctionArn": "arn:aws:lambda:us-west-2:111111111111:function:grid-maker",

"Runtime": "python3.9",

"Role": "arn:aws:iam::111111111111:role/LambdaApplicationRole",

"Handler": "lambda\_function.lambda\_handler",

"CodeSize": 15827608,

"Description": "",

"Timeout": 30,

"MemorySize": 128,

"LastModified": "2023-03-07T21:22:23.176+0000",

"CodeSha256": "OvUsBpL4h2X+Yy38mrxa1YszPk2bnmTItSfUX8IhhcI=",

"Version": "$LATEST",

"TracingConfig": {

"Mode": "PassThrough"

},

"RevisionId": "08ad3b46-319f-4042-b432-da014f151cac",

"State": "Pending",

"StateReason": "The function is being created.",

"StateReasonCode": "Creating",

"PackageType": "Zip",

"Architectures": [

"x86\_64"

],

"EphemeralStorage": {

"Size": 512

},

"SnapStart": {

"ApplyOn": "None",

"OptimizationStatus": "Off"

},

"RuntimeVersionConfig": {

"RuntimeVersionArn": "arn:aws:lambda:us-west-2::runtime:07a48df201798d627f2b950f03bb227aab4a655a1d019c3296406f95937e2525"

}

}

Now that you created the *grid-maker* Lambda function, you create variables for the *source-images* and *destination-images* S3 buckets. These values are used to update the *event.json* file as inputs when you run the Lambda function.

1. **Command:** To create the two environment variables to store the **source-images** and **destination-images** S3 bucket names, and to update the **event.json** file with the bucket names, run the following commands:

destination\_bucket=$(aws s3api list-buckets --output text --query 'Buckets[?contains(Name, `destination-images`) == `true`] | [0].Name')

source\_bucket=$(aws s3api list-buckets --output text --query 'Buckets[?contains(Name, `source-images`) == `true`] | [0].Name')

printf "{\n \"SOURCE\_BUCKET\": \"$source\_bucket\",\n \"DESTINATION\_BUCKET\": \"$destination\_bucket\"\n}" > ~/environment/lambda-function/event.json

printf "\nThe event.json file has been updated with the following content:\n\n" ; cat event.json ; printf "\n\n"

**Note:** The *event.json* file passes two parameters for the *SOURCE\_BUCKET* and *DESTINATION\_BUCKET*. The parameters are pulled from the application in the *lambda\_handler* function.

**Expected output:**

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

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

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

The event.json file has been updated with the following content:

{

"SOURCE\_BUCKET": "source-images-us-west-2-0755207",

"DESTINATION\_BUCKET": "destination-images-us-west-2-4145392"

}

Invoke the grid-maker Lambda function to and copy the S3 presigned URL that is output to view the image in a new browser tab.

1. **Command:** To invoke the **grid-maker** Lambda function and output the S3 presigned URL, run the following grouping of commands:

aws lambda invoke --payload fileb://event.json --function-name grid-maker ~/environment/output.txt && cat ~/environment/output.txt | tr -d '"' > ~/environment/output\_without\_quotes.txt && mv ~/environment/output\_without\_quotes.txt ~/environment/output.txt && echo -e "\nS3 presigned URL to copy: \n" && cat ~/environment/output.txt && echo -e "\n"

**Expected output:**

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

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

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

{

"StatusCode": 200,

"ExecutedVersion": "$LATEST"

}

S3 presigned URL to copy:

https://destination-images-us-west-2-9675590.s3.amazonaws.com/b15296410ee69131fc4189b2167f1fbc.jpg?AWSAccessKeyId=ASIASSGTW44QLENTF4WC&Signature=VLpRpb0YX%2BHUUjOKETGtj5ISMEA%3D&x-amz-security-token=IQoJb3JpZ2luX2VjECUaCXVzLXdlc3QtMiJHMEUCIBIOq0yDKpFV34sfm5oWPMVILMLMzrxpIZnC94k3RC2OAiEAluT7bZ57YoNz7h3Bf8qHItYMa2GJD0wit94bbxdjyNcq6AIIThAEGgwxNzY1Mzc2NTkxNjgiDF6hZYdsyspoI%2FUD8yrFAuDNaOLoujcJvORT3fH63z2hvwWt8UaEYjIOHpdTUy9desv%2BLIkT2pwugx%2F2dMUT3So9ez8Mg%2BZYkvKqGRaqeOxOBgD%2F4eFGoGyWZ4FQsOb07mdvJQnbfceqSfLd6%2BuQMQtrk%2FQxSilrWVo77gDdCODgqk3d39WASvmRr7oCIKKEZJnZPgRjgagiuEQfcpPn5crKZM8UP4g6vjAJB4kcu1CdmhdRJkboV%2Bd6iPkSz%2BAFHe1YyJiC5nnrHdFIvnZgRTZuKJd%2FiRScAVtrE0vqTulk9FfT5TiCQtUO9zmPRIlVA%2FH7B0ZYeEVUUVTqvwqNg%2FqN%2FOJoJBvt5C9j%2F35A1%2FCm32XkPVGN7G9OO8FKx40GBGSB2H8szByTDxUL70DugTBAdheZtnTq2%2BzwRmUn%2B6ey5IExoDV88vu6Ifff9x29ubAG6C0w4%2FSUowY6ngE0%2BF8EPS7RoxNjQHDXrSA8U%2FECaH9mu9mxC5sFy9F6nAPbrLPiYQlM%2BEhqJr0T%2FmATDg5SdlBh%2BkJkkYLhvW789Ol8yZU6E9VU%2BE1Dopp1RD54zzDW%2BRqkCymA%2BMaJfEvYyg5q9rfGmH49SsRoFCUyItHgCpZQT5%2B6bcPf6sXl3n2ba%2FAUytJXzVBJvn6gy9G7gvSZtXcnMb3eg595%2FQ%3D%3D&Expires=1684357263

1. Copy the **S3 presigned URL** and paste it into a new browser tab.

You should now see the **grid-image.jpg** that you created from 16 individual images by using Lambda as the compute environment.



**Congratulations!** You successfully created the Lambda function based on the application package and initiated the Lambda function. The new *grid-image.jpg* file was created and 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*.

**Task 3: Review the logs in CloudWatch Logs after invoking the Lambda function**

In this task, you inspect the logs in CloudWatch Logs by using the

aws logs tail

 command in a new terminal session. Then, you invoke the Lambda function in the other terminal session one more time to see the logs update in real time.

1. Open a new terminal session and move it to a different location than the current terminal session.
2. **Command:** To tail the logs for the CloudWatch Logs log group identified as **/aws/lambda/grid-maker**, run the following command:

aws logs tail /aws/lambda/grid-maker --follow --format json

**Expected output:** Nothing at this time.

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

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

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

AWSLabsUser-9ew7ZDoHSuCpHxbvcK1gGU:~/environment/lambda-function $ aws logs tail /aws/lambda/grid-maker --follow --format json

1. Return to the other terminal session opened to the **~/environment/lambda-function** directory.
2. **Command:** To invoke the grid-maker Lambda function on more time, run the following command:

aws lambda invoke --payload fileb://event.json \

--function-name grid-maker ~/environment/output.txt

1. Switch back to the terminal that’s tailing the **/aws/lambda/grid-maker** CloudWatch Logs log group.

It can take a few moments, but you should see output that’s similar to the following.

**Expected output:**

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

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

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

2023-03-21T19:29:53.445000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 INIT\_START Runtime Version: python:3.9.v18 Runtime Version ARN: arn:aws:lambda:us-west-2::runtime:edb5a058bfa782cb9cedc6d534ac8b8c193bc28e9a9879d9f5ebaaf619cd0fc0

2023-03-21T19:29:53.929000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 START RequestId: 3f3438cf-fbe6-4edd-b140-06ce53b47535 Version: $LATEST

2023-03-21T19:29:54.230000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 Converting: 16 source images.

2023-03-21T19:29:54.230000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 Creating: 4 x 4 grid.

2023-03-21T19:29:57.874000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 Creating temp file /tmp/tmp992gzc9a.jpg

2023-03-21T19:29:58.225000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 Saved file to s3 bucket: destination-images-us-west-2-4145392, key: b2151bda58ab3d0818c9b1dca2a6b9af.jpg

2023-03-21T19:29:58.226000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 Presigned url: https://destination-images-us-west-2-4145392.s3.amazonaws.com/b2151bda58ab3d0818c9b1dca2a6b9af.jpg?AWSAccessKeyId=ASIAXRRUMSOOFUKYPNOR&Signature=dSKhXPxkCF5ScBue5euyZq4SFYk%3D&x-amz-security-token=IQoJb3JpZ2luX2VjEMz%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLXdlc3QtMiJHMEUCIH2cMdxTzX306GX0Cjo6r1Ti2DhUr18BiUOZH5U5WhDeAiEA6o8AseykpVN9PnXd9mXXCmidiHREFEVSR0hsmnmaixQq8QIIlf%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FARACGgw1MTg3MjcxNzcxMTYiDFHIBg2Nbhpb7q8GtyrFAvVUZu7ukAYIXSSNnWiNYOWdJIqIjfb5mpUwHIR3HyqGR%2BfIqY%2B33Lr0qSiS%2BGDHgJyqdjc0u2kh0NeIDq%2B5vvsek6SJMGdsnhVC6YxUYbwIggGfFAobjJN1cTkKSKmSKLvINDi3rYym%2BpX4%2FV8hA7PkPP7ffmVpH2cOACZqmNrcO7XZPF6MrjtuJKBTh7cMtAt4BrB1oVkQRlAh6rlfcODdePkaYUwBxcAIfmLRZCaDUbogZTjksVMbwvQkB2zvl8xKVN%2Fi2js0bGAY4kMLbfaoRX9%2BiplXFr2oJxpnJ2WWePyCr5%2FTcSUiSrACv%2FcCboBwkp0Wm7b6v1K3Lq%2BQVBbXNzj668My7S7IAxOkZqg5g5DhoouojblVyeqMGo2k6r%2FLB7eOqac5Yh8aBWG1gk9KRWsmxFC%2FHqtyfrqgif0Atxsm63UwsYvooAY6ngHFek8GxQbLNYHHAd2jpZErFkdJDj1nr7ODXX3lcgWWug7MoPGBazYBHXuwCycPawWO%2Fq8tkAcJ3epN%2FPNc0xxtpBwESmqFsCALKxDEawac%2BnL06pFSbtrh63oI3RvUtC5ncoNH8WRfhtTkOaj5w2s%2Bq4I4jOBgSU5z4KdFtHbvaiIbHjGk4s9OgE7WqSKfAT17utueQOW8a1KhRwlI6Q%3D%3D&Expires=1679427298

2023-03-21T19:29:58.232000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 END RequestId: 3f3438cf-fbe6-4edd-b140-06ce53b47535

2023-03-21T19:29:58.232000+00:00 2023/03/21/[$LATEST]07c143616e91462ab6ac6359162457b2 REPORT RequestId: 3f3438cf-fbe6-4edd-b140-06ce53b47535 Duration: 4302.91 ms Billed Duration: 4303 ms Memory Size: 128 MB Max Memory Used: 84 MB Init Duration: 483.76 ms

You can see log entries for the following actions:

* *Converting: 16 source images*
* *Creating: 4 x 4 grid*
* *Creating temp file*
* *Saved file to s3 bucket: destination-images-us-west-2-…*
* *Presigned url: https://destination-images-us-west-2-*

 Congratulations! You have reviewed the logs for the Create-Grid Lambda function in CloudWatch Logs. You were able to see all the actions that the Lambda function needed to take to create the grid image and the S3 presigned URL.

**Conclusion**

 Congratulations! You now have successfully:

* Created a deployment package, including the application code and Python package requirements.
* Run an AWS CLI command to create a Lambda function.
* Run a Lambda function to create the grid image.
* Inspected Amazon CloudWatch Logs after invoking a Lambda function.

**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).