



AWS S3 Image Upload and Real-time File Processing with Rekognition on AmebaPro – User Guide



Realtek Semiconductor Corp.

No. 2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211. Fax: +886-3-577-6047

www.realtek.com

COPYRIGHT

©2019 Realtek Semiconductor Corp. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Realtek Semiconductor Corp.

DISCLAIMER

Please Read Carefully:

Realtek Semiconductor Corp., (Realtek) reserves the right to make corrections, enhancements, improvements and other changes to its products and services. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Reproduction of significant portions in Realtek data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Realtek is not responsible or liable for such reproduced documentation. Information of third parties may be subject to additional restrictions.

Buyers and others who are developing systems that incorporate Realtek products (collectively, “Customers”) understand and agree that Customers remain responsible for using their independent analysis, evaluation and judgment in designing their applications and that Customers have full and exclusive responsibility to assure the safety of Customers' applications and compliance of their applications (and of all Realtek products used in or for Customers' applications) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to their applications, Customer has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Customer agrees that prior to using or distributing any applications that include Realtek products, Customer will thoroughly test such applications and the functionality of such Realtek products as used in such applications.

Realtek's provision of technical, application or other design advice, quality characterization, reliability data or other services or information, including, but not limited to, reference designs and materials relating to evaluation kits, (collectively, “Resources”) are intended to assist designers who are developing applications that incorporate Realtek products; by downloading, accessing or using Realtek's Resources in any way, Customer (individually or, if Customer is acting on behalf of a company, Customer's company) agrees to use any particular Realtek Resources solely for this purpose and subject to the terms of this Notice.

Realtek's provision of Realtek Resources does not expand or otherwise alter Realtek's applicable published warranties or warranty disclaimers for Realtek's products, and no additional obligations or liabilities arise from Realtek providing such Realtek Resources. Realtek reserves the right to make corrections, enhancements, improvements and other changes to its Realtek Resources. Realtek has not conducted any testing other than that specifically described in the published documentation for a particular Realtek Resource.

Customer is authorized to use, copy and modify any individual Realtek Resource only in connection with the development of applications that include the Realtek product(s) identified in such Realtek Resource. No other license, express or implied, by estoppel or otherwise to any other Realtek intellectual property right, and no license to any technology or intellectual property right of Realtek or any third party is granted herein, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Realtek products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of Realtek Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from Realtek under the patents or other Realtek's intellectual property.

Realtek's Resources are provided “as is” and with all faults. Realtek disclaims all other warranties or representations, express or implied, regarding resources or use thereof, including but not limited to accuracy or completeness, title, any epidemic failure warranty and any implied warranties of merchantability, fitness for a particular purpose, and non-infringement of any third party intellectual property rights.

Realtek shall not be liable for and shall not defend or indemnify Customer against any claim, including but not limited to any infringement claim that related to or is based on any combination of products even if described in Realtek Resources or otherwise. In no event shall Realtek be liable for any actual, direct, special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of Realtek's Resources or use thereof, and regardless of whether Realtek has been advised of the possibility of such damages. Realtek is not responsible for any failure to meet such industry standard requirements.

Where Realtek specifically promotes products as facilitating functional safety or as compliant with industry functional safety standards, such products are intended to help enable customers to design and create their own applications that meet applicable functional safety standards and requirements. Using products in an application does not by itself establish any safety features in the application. Customers must ensure compliance with safety-related requirements and standards applicable to their applications. Designer may not use any Realtek products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death. Such equipment includes, without limitation, all medical devices identified by the U.S.FDA as Class III devices and equivalent classifications outside the U.S.

Customers agree that it has the necessary expertise to select the product with the appropriate qualification designation for their applications and that proper product selection is at Customers' own risk. Customers are solely responsible for compliance with all legal and regulatory requirements in connection with such selection.

Customer will fully indemnify Realtek and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's non-compliance with the terms and provisions of this Notice.

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.




USING THIS DOCUMENT

Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide.



1 What Can We Get in this Tutorial

AmebaPro can upload the image to AWS S3 cloud storage, then trigger a AI image processing function to do object detection! Here, we have some application scenario introduced: 1.1 wild animal, 1.2 vehicle, 1.3 food, 1.4 retail product.

1.1 Wild Animal Detection

Uploaded image	Rekognition results (AWS AI)
	<p>Label: Lion, Confidence: 99.99998474121094</p> <p>Label: Wildlife, Confidence: 99.99998474121094</p> <p>Label: Mammal, Confidence: 99.99998474121094</p> <p>Label: Animal, Confidence: 99.99998474121094</p>
	<p>Label: Zebra, Confidence: 98.60205841064453</p> <p>Label: Animal, Confidence: 98.60205841064453</p> <p>Label: Mammal, Confidence: 98.60205841064453</p> <p>Label: Wildlife, Confidence: 98.60205841064453</p>
	<p>Label: Elephant, Confidence: 99.82683563232422</p> <p>Label: Mammal, Confidence: 99.82683563232422</p> <p>Label: Animal, Confidence: 99.82683563232422</p> <p>Label: Wildlife, Confidence: 99.82683563232422</p>




1.2 Vehicle Detection

Uploaded image	Rekognition results (AWS AI)
	<p>Label: Car, Confidence: 99.77043914794922</p> <p>Label: Vehicle, Confidence: 99.77043914794922</p> <p>Label: Transportation, Confidence: 99.77043914794922</p> <p>Label: Road, Confidence: 96.77059173583984</p> <p>Label: Traffic Jam, Confidence: 88.41131591796875</p> <p>Label: Truck, Confidence: 87.05320739746094</p> <p>Label: Person, Confidence: 83.31153869628906</p> <p>Label: License Plate, Confidence: 75.08149719238281</p> <p>Label: Freeway, Confidence: 66.02681732177734</p> <p>Label: Highway, Confidence: 59.900604248046875</p>
	<p>Label: Person, Confidence: 99.8815689086914</p> <p>Label: Car, Confidence: 99.66630554199219</p> <p>Label: Transportation, Confidence: 99.66630554199219</p> <p>Label: Vehicle, Confidence: 99.66630554199219</p> <p>Label: Traffic Light, Confidence: 93.42364501953125</p> <p>Label: Light, Confidence: 93.42364501953125</p> <p>Label: Taxi, Confidence: 87.03340148925781</p> <p>Label: Road, Confidence: 80.97848510742188</p> <p>Label: Pedestrian, Confidence: 71.65261840820312</p>

1.3 Food Detection

Uploaded image	Rekognition results (AWS AI)
	<p>Label: Cake, Confidence: 93.53020477294922</p> <p>Label: Dessert, Confidence: 93.53020477294922</p> <p>Label: Food, Confidence: 93.53020477294922</p> <p>Label: Sweets, Confidence: 92.21732330322266</p> <p>Label: Strawberry, Confidence: 92.0057601928711</p> <p>Label: Fruit, Confidence: 92.0057601928711</p>
	<p>Label: Plant, Confidence: 98.99000549316406</p> <p>Label: Food, Confidence: 91.17074584960938</p> <p>Label: Vegetable, Confidence: 88.99947357177734</p> <p>Label: Carrot, Confidence: 82.17688751220703</p> <p>Label: Produce, Confidence: 80.4115982055664</p>

1.4 Retail Product Detection

Uploaded image	Rekognition results (AWS AI)
	<p>Label: Grocery Store, Confidence: 95.96288299560547</p> <p>Label: Shop, Confidence: 95.96288299560547</p> <p>Label: Market, Confidence: 95.61573791503906</p> <p>Label: Supermarket, Confidence: 93.32327270507812</p> <p>Label: Food, Confidence: 75.57024383544922</p> <p>Label: Refrigerator, Confidence: 69.4200439453125</p> <p>Label: Candy, Confidence: 58.83950424194336</p>
	<p>Label: Market, Confidence: 97.97370147705078</p> <p>Label: Grocery Store, Confidence: 97.06415557861328</p> <p>Label: Shop, Confidence: 97.06415557861328</p> <p>Label: Shelf, Confidence: 96.08903503417969</p> <p>Label: Supermarket, Confidence: 93.59891510009766</p> <p>Label: Food, Confidence: 73.77120208740234</p> <p>Label: Candy, Confidence: 65.2990951538086</p>
	<p>Label: Soda, Confidence: 99.874267578125</p> <p>Label: Beverage, Confidence: 99.874267578125</p> <p>Label: Market, Confidence: 97.0456771850586</p> <p>Label: Supermarket, Confidence: 95.8358154296875</p> <p>Label: Shop, Confidence: 95.8358154296875</p> <p>Label: Grocery Store, Confidence: 95.8358154296875</p> <p>Label: Coke, Confidence: 66.05561828613281</p> <p>Label: Shelf, Confidence: 62.23150634765625</p> <p>Label: Juice, Confidence: 49.38665008544922</p>

2 Set Up AmebaPro to Upload the File to S3

Please follow the steps in the following to set up your device.

2.1 Download the Project for FreeRTOS v202012-LTS

This tutorial is based on **Amazon-v202012-LTS** framework, we can clone the project from Github in specified branch, **AmebaPro-202012.00-LTS-dev**: https://github.com/HungTseLee/KVS_WebRTC_on_AmebaPro/tree/AmebaPro-202012.00-LTS-dev

Run the command to download the whole project:

```
$ git clone -b AmebaPro-202012.00-LTS-dev --recurse-submodules https://github.com/HungTseLee/KVS_WebRTC_on_AmebaPro.git
```

If you already have a checkout, run the following command to sync submodules:

```
$ git submodule update --init
```

2.2 Configure the AWS Credential File

Configure **aws_clientcredential.h** and **aws_clientcredential_keys.h**

Refer to <https://docs.aws.amazon.com/freertos/latest/userguide/freertos-configure.html>, which will have the instructions.

In **aws_clientcredential.h**(component/common/application/amazon/amazon-freertos-202012.00/demos/include), set network connection related info

```
#define clientcredentialWIFI_SSID          "SSID"
#define clientcredentialWIFI_PASSWORD    "PASSWORD"
```

In **aws_clientcredential_keys.h**(component/common/application/amazon/amazon-freertos-202012.00/demos/include), set Demo required credentials

```
#define keyCLIENT_CERTIFICATE_PEM        "CERTIFICATE"
#define keyCLIENT_PRIVATE_KEY_PEM      "PRIVATE_KEY"
```

2.3 Enable the Demo for S3 File Upload

define the **CONFIG_CORE_HTTP_S3_UPLOAD_DEMO_ENABLED** in **aws_demo_config.h**(component\common\application\amazon\amazon-freertos-202012.00-Pro\vendors\realtek\boards\amebaPro\aws_demos\config_files)

```
//#define CONFIG_CORE_MQTT_MUTUAL_AUTH_DEMO_ENABLED
//#define CONFIG_OTA_UPDATE_DEMO_ENABLED
//#define CONFIG_DEVICE_SHADOW_DEMO_ENABLED
#define CONFIG_CORE_HTTP_S3_UPLOAD_DEMO_ENABLED
//#define CONFIG_CORE_HTTP_MUTUAL_AUTH_DEMO_ENABLED
```

2.4 Add the Image Data Used to Upload to S3

For the image data used to upload, we can store the image raw data in DRAM or SD card for demo. Here, we introduce how to store an image data in DRAM and upload it to S3. For further application, user can use their image source buffer to replace it.

Now, we can add the image data to the **http_demo_s3_upload_config.h** (component\common\application\amazon\amazon-freertos-202012.00-Pro\vendors\realtek\boards\amebaPro\aws_demos\config_files), you can use the tool like "HxD" to generate the data array of image

Here is the example, the demo image contain the food, so we call the array as **food_data**:

```
#include "section_config.h"
SDRAM_DATA_SECTION uint8_t food_data[942346] = {
    0xFF, 0xD8, 0xFF, 0xE0, 0x00, 0x10, 0x4A, 0x46, 0x49, 0x46, 0x00, 0x01,
    0x01, 0x01, 0x01, 0x2C, 0x01, 0x2C, 0x00, 0x00, 0xFF, 0xDB, 0x00, 0x43,
    0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
    .....
    0x19, 0xAC, 0x90, 0x3A, 0xF3, 0x41, 0xDA, 0xFC, 0xB9, 0x00, 0xFF, 0x00,
    0x52, 0x3F, 0x04, 0xFA, 0xA9, 0x23, 0xB5, 0x57, 0xFF, 0xD9
};
```

You can replace the buffer data with your own image!

2.5 Modify the Pointer to Image Data

in **http_demo_s3_upload.c** (component\common\application\amazon\amazon-freertos-202012.00-Pro\demos\coreHTTP)

```
/* Pointer to the data to upload. */
#ifdef democonfigDEMO_HTTP_UPLOAD_DATA
    #define democonfigDEMO_HTTP_UPLOAD_DATA food_data
#endif
```

2.6 Make Pre-signed URL for upload file to S3

refer to the **README.md** in (component\common\application\amazon\amazon-freertos-202012.00\demos\common\http_demo_helpers\presigned_url_generator)

use **presigned_urls_gen.py** to generate the URL, then paste the URL to "**http_demo_s3_upload.c**"

```
/* Check that the pre-signed GET URL is defined. */
#ifdef democonfigS3_PRE_SIGNED_GET_URL
    #define democonfigS3_PRE_SIGNED_GET_URL "https://XXXXXX.s3.amazonaws.com/XXX.png?XXXXXX..."
#endif

/* Check that the pre-signed PUT URL is defined. */
#ifdef democonfigS3_PRE_SIGNED_PUT_URL
    #define democonfigS3_PRE_SIGNED_PUT_URL "https://XXXXXX.s3.amazonaws.com/XXX.png?XXXXXX..."
#endif
```


2.7 Build the Project and Download the Image to AmebaPro

Build the project and download the image to AmebaPro.

Build the project in GCC:

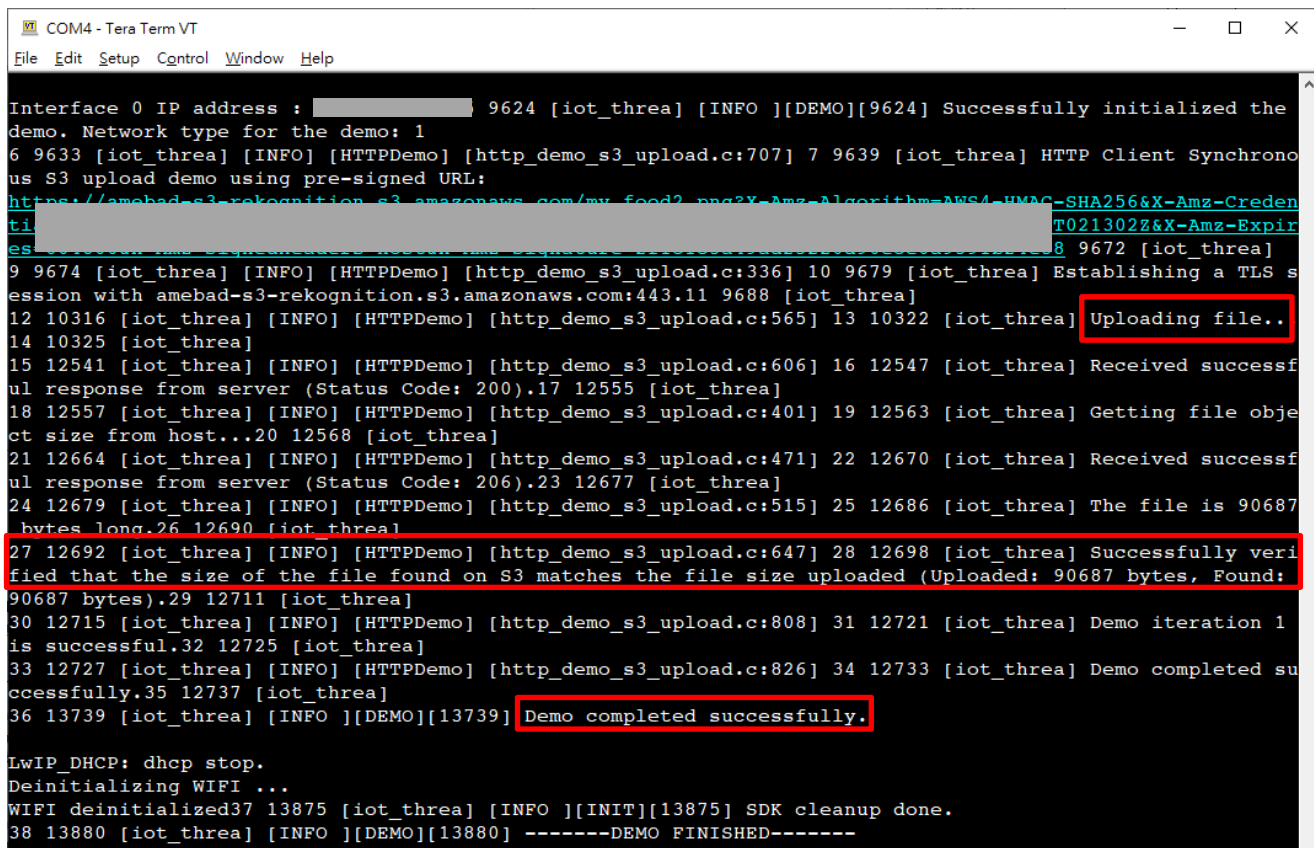
```
$ make -f Makefile_amazon_LTS all
```

If success, the image file(**flash_is.bin**) will be generated in (project\realtek_amebapro_v0_example\GCC-RELEASE\application_is)
Then, we can use the **image tool**(tools\AmebaPro\Image_Tool\ ImageTool.exe) to download the image to AmebaPro.

Note:

For using the image tool, please see application note: https://github.com/HungTseLee/KVS_WebRTC_on_AmebaPro/blob/AmebaPro-202012.00-LTS-dev/doc/AN0300%20Realtek%20AmebaPro%20application%20note.en.pdf

After downloading the image to device, please reboot AmebaPro to run the demo. The stored image will then be uploaded to your S3 bucket!



```
COM4 - Tera Term VT
File Edit Setup Control Window Help

Interface 0 IP address : 9624 [iot_threa] [INFO ][DEMO][9624] Successfully initialized the
demo. Network type for the demo: 1
6 9633 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:707] 7 9639 [iot_threa] HTTP Client Synchrono
us S3 upload demo using pre-signed URL:
https://amebad-s3-rekognition-s3.amazonaws.com/my_food?x-amz-algorithm=AWS4-HMAC-SHA256&x-amz-creden
ti
9 9674 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:336] 10 9679 [iot_threa] Establishing a TLS s
ession with amebad-s3-rekognition.s3.amazonaws.com:443.11 9688 [iot_threa]
12 10316 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:565] 13 10322 [iot_threa] Uploading file..
14 10325 [iot_threa]
15 12541 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:606] 16 12547 [iot_threa] Received successf
ul response from server (Status Code: 200).17 12555 [iot_threa]
18 12557 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:401] 19 12563 [iot_threa] Getting file obje
ct size from host...20 12568 [iot_threa]
21 12664 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:471] 22 12670 [iot_threa] Received successf
ul response from server (Status Code: 206).23 12677 [iot_threa]
24 12679 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:515] 25 12686 [iot_threa] The file is 90687
bytes long.26 12690 [iot_threa]
27 12692 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:647] 28 12698 [iot_threa] Successfully veri
fied that the size of the file found on S3 matches the file size uploaded (Uploaded: 90687 bytes, Found:
90687 bytes).29 12711 [iot_threa]
30 12715 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:808] 31 12721 [iot_threa] Demo iteration 1
is successful.32 12725 [iot_threa]
33 12727 [iot_threa] [INFO ][HTTPDemo] [http_demo_s3_upload.c:826] 34 12733 [iot_threa] Demo completed su
ccessfully.35 12737 [iot_threa]
36 13739 [iot_threa] [INFO ][DEMO][13739] Demo completed successfully.

LwIP_DHCP: dhcp stop.
Deinitializing WIFI ...
WIFI deinitialized37 13875 [iot_threa] [INFO ][INIT][13875] SDK cleanup done.
38 13880 [iot_threa] [INFO ][DEMO][13880] -----DEMO FINISHED-----
```

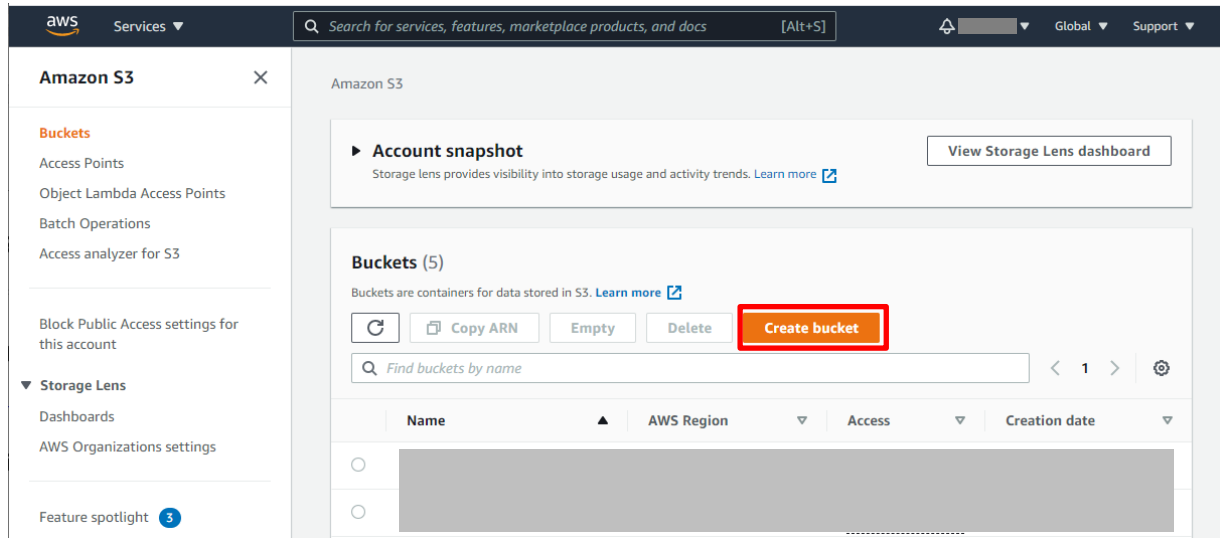
3 Real-time File Processing with Rekognition

After setting up the AmebaPro, we can do the real-time image processing on AWS website.

3.1 Create an Amazon S3 bucket Using the Console

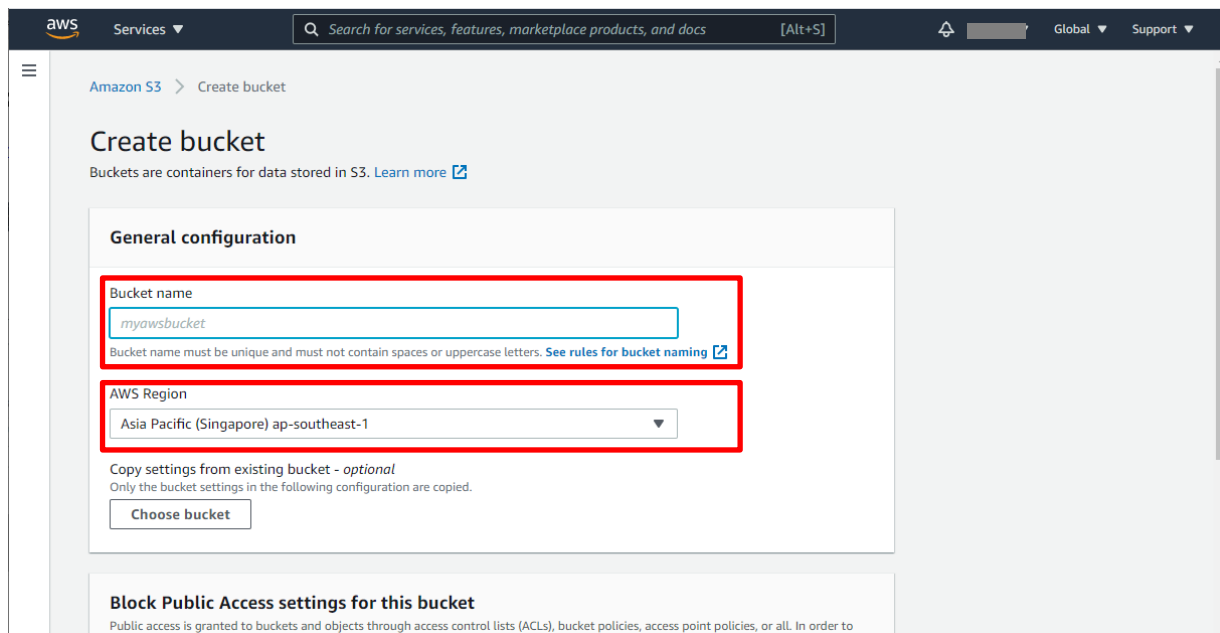
Go to Amazon S3 console: <https://console.aws.amazon.com/s3/home>

Create a bucket by choosing **Create bucket**

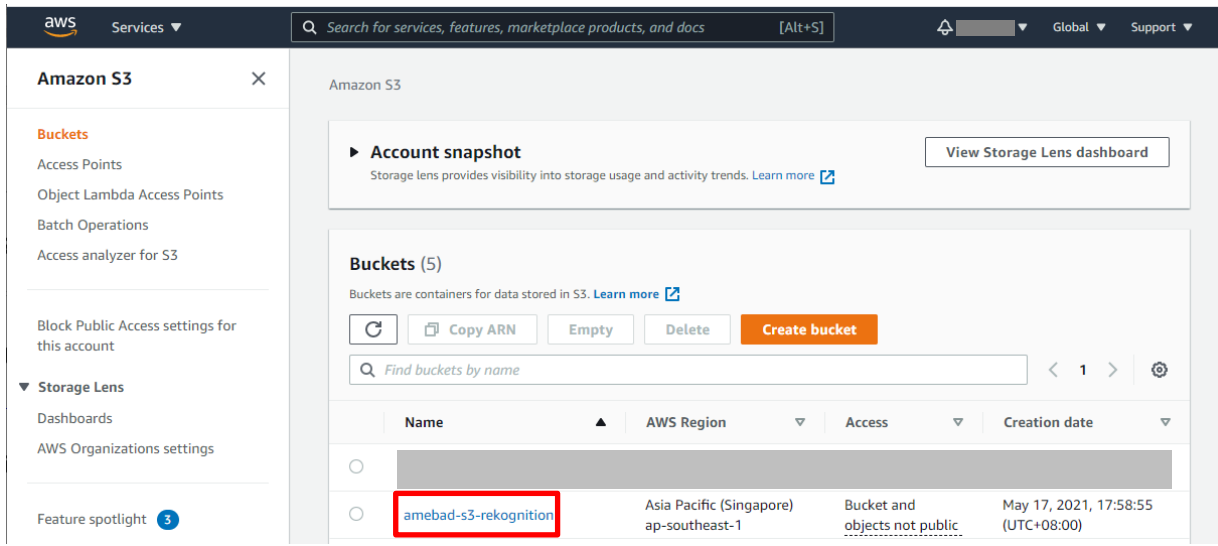


For Bucket name, enter a **unique name**.

For AWS Region, choose a **Region**. Note that **you must create your Lambda function in the same Region**.



After creating the bucket, you can see the list of buckets in your account in the current Region.

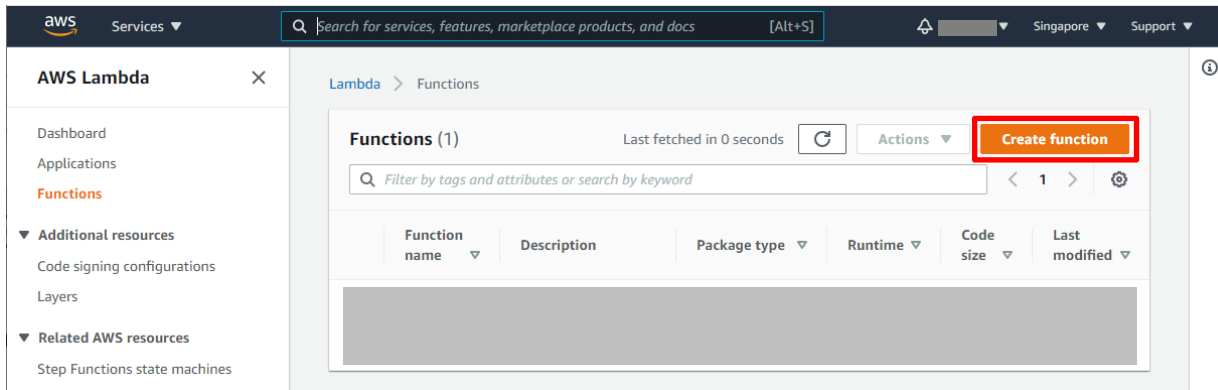


3.2 Create an AWS Lambda function to Trigger the Rekognition Service

We can refer <https://docs.aws.amazon.com/lambda/latest/dg/with-s3-example.html> to create a Lambda function easily.

Go to Functions page on the Lambda console: <https://console.aws.amazon.com/lambda/home#/functions>

Choose **Create function**



On the Create function page, choose **Use a blueprint**.

Under Blueprints, enter **s3** in the search box.

For a Python function, choose **s3-get-object-python**.

However, the sample code provided is python2.7, we can revise it to python3.7 version later.

The screenshot shows the AWS Lambda 'Create function' page. At the top, there's a navigation bar with 'aws', 'Services', and a search bar. Below the navigation bar, the breadcrumb trail is 'Lambda > Functions > Create function'. The main heading is 'Create function' with an 'Info' link. A sub-heading says 'Choose one of the following options to create your function.' There are four options: 'Author from scratch' (radio button), 'Use a blueprint' (radio button, selected and highlighted with a red box), 'Container image' (radio button), and 'Browse serverless app repository' (radio button). Below these options is a section titled 'Blueprints' with an 'Info' link and an 'Export' button. A search bar is present with the text 'Filter by tags and attributes or search by keyword'. Below the search bar, there's a filter button labeled 's3' and a 'Clear filters' button. To the right of the search bar, it says '3 matches' and a pagination control showing '1'. Below the search bar, there are three blueprint cards. The first card is 's3-get-object-python' with a description 'An Amazon S3 trigger that retrieves metadata for the object that has been updated.' and 'python3.7 · s3'. The second card is 'rekognition-python' (selected and highlighted with a red box) with a description 'An Amazon S3 trigger that uses rekognition APIs to detect faces' and 'python2.7 · rekognition · s3'. The third card is 's3-get-object' with a description 'An Amazon S3 trigger that retrieves metadata for the object that has been updated.' At the bottom of the page, there's a footer with 'Feedback', 'English (US)', '© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', 'Terms of Use', and 'Cookie preferences'.

Under Basic information, do the following:

For **Function name**, enter your function name (ex: my-s3-function)

For **Execution role**, choose **Create a new role from AWS policy templates**.

For **Role name**, enter your role name (ex: my-s3-function-role)

Basic information Info

Function name
myFunctionName

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☐ Use an existing role

☒ Create a new role from AWS policy templates

Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Role name
Enter a name for your new role.
myRoleName
Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates - optional Info
Choose one or more policy templates.

Amazon S3 object read-only permissions S3

Amazon Rekognition write-only permissions Rekognition

Amazon Rekognition no data permissions Rekognition

Under S3 trigger, **choose the S3 bucket that you created previously**. Then, choose **Create function**.

Now, your lambda function is created, and it will be triggered automatically once AmebaPro upload an image file to the S3 bucket!

If you want to use python3.7, you can modify it as following. The sample code for food detection in python3.7 is provided along with this tutorial document: **lambda_function.py**

Code

```

22 response = rekognition.detect_labels(
23     Image={"S3Object": {"Bucket": "bucket", "Name": "key"}},
24     MaxLabels=30,
25     MinConfidence=33
26 )
27
28 # Sample code to write response to DynamoDB table 'MyTable' with 'PK' as Primary Key.
29 # Note: role used for executing this Lambda function should have write access to the table.
30 #table = boto3.resource('dynamodb').Table('MyTable')
31 #labels = [{"Confidence": Decimal(str(label_prediction['Confidence']))}, {'Name': label_prediction['Name']}] for label_
32 #table.put_item(Item={'PK': key, 'Labels': labels})
33 return response
34
35 def index_faces(bucket, key):
36     # Note: Collection has to be created upfront. Use CreateCollection API to create a collection.
37

```

Code properties

Package size 2.1 kB	SHA256 hash F2GERu8n7NWVvYy7XJIXGZNqDbEBHODtZs9Bi PZHzyw=	Last modified May 21, 2021, 01:44 PM GMT+8
------------------------	---	---

Runtime settings Info

Runtime
Python 3.7

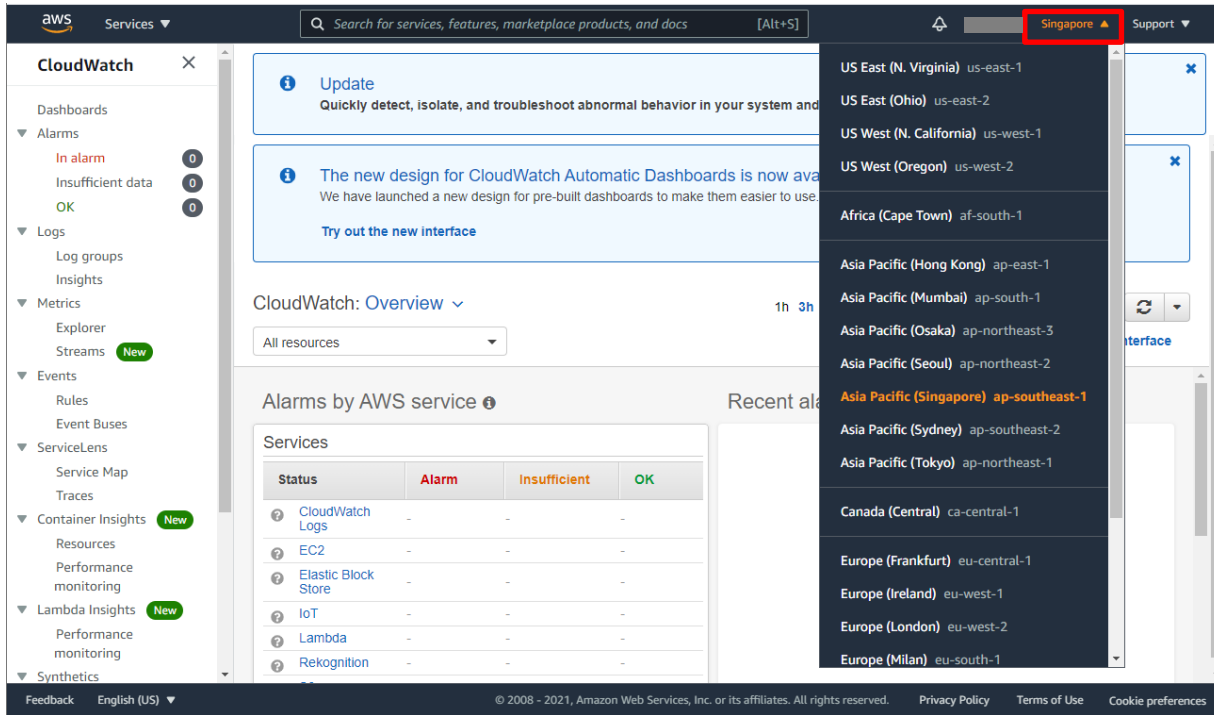
Handler Info
lambda_function.lambda_handler

Edit

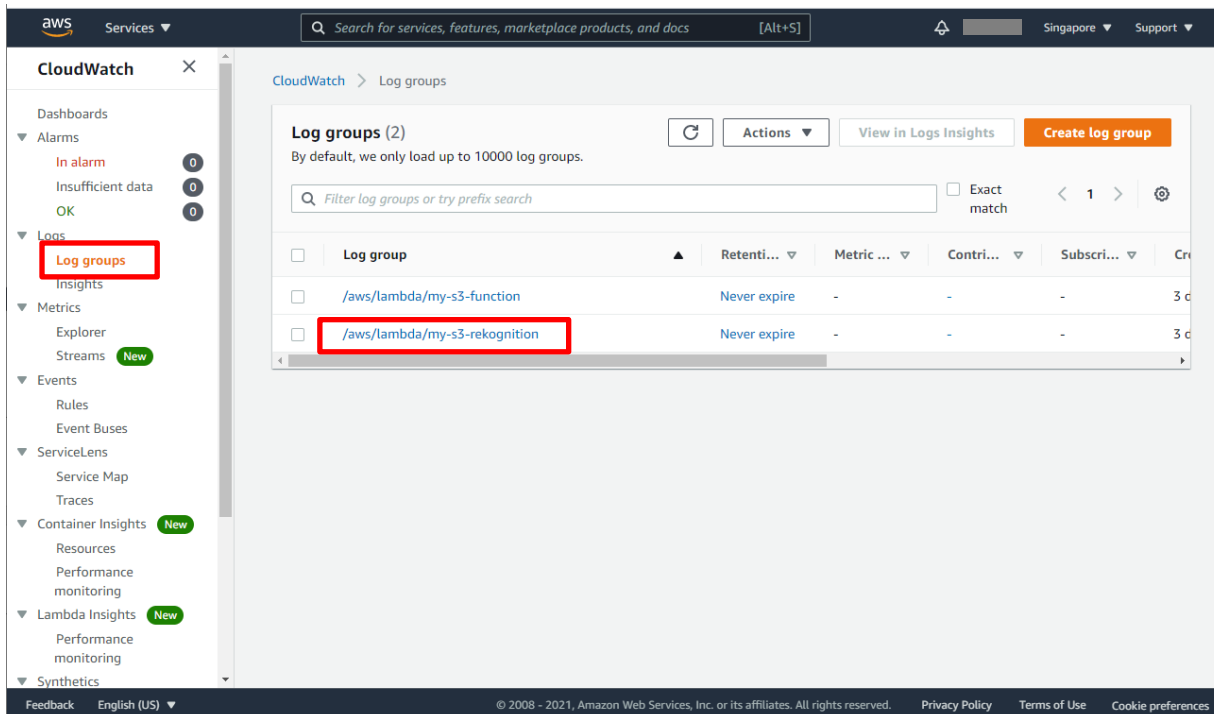
3.3 See the Object Detection Result in AWS CloudWatch

Go to AWS CloudWatch: <https://console.aws.amazon.com/cloudwatch/home#>

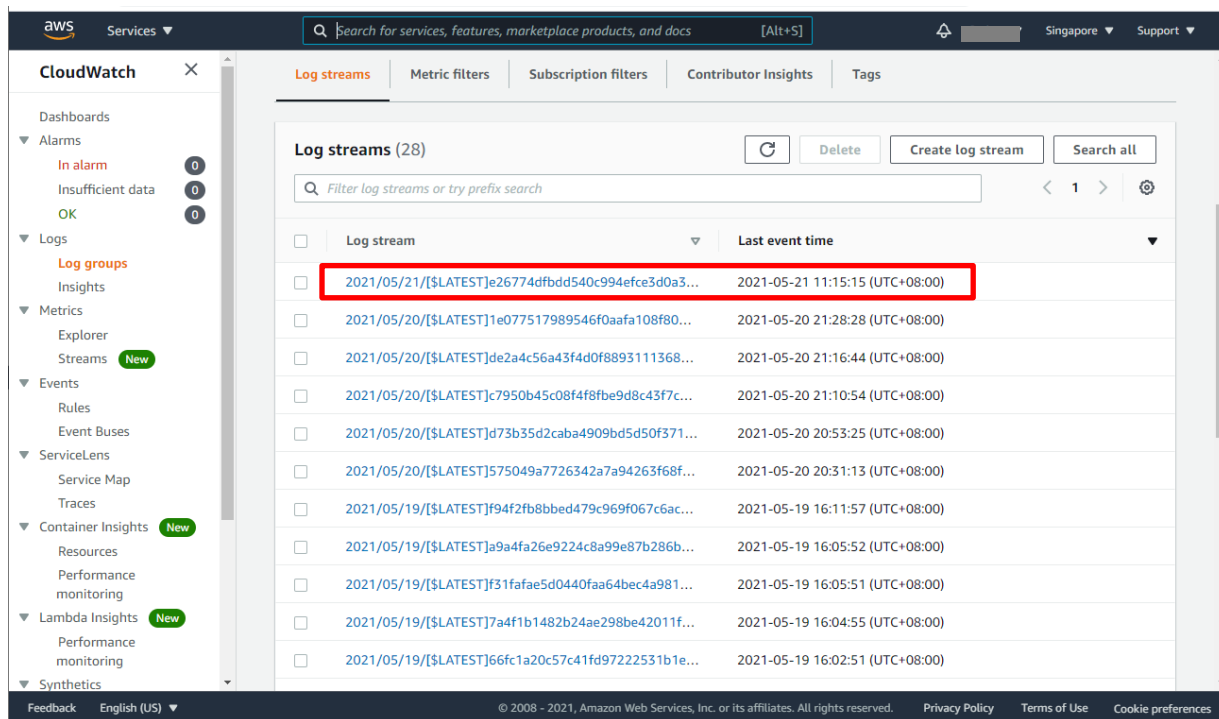
Choose the correct AWS region, our example default is **ap-southeast-1**



Click the Log groups button, and choose your group

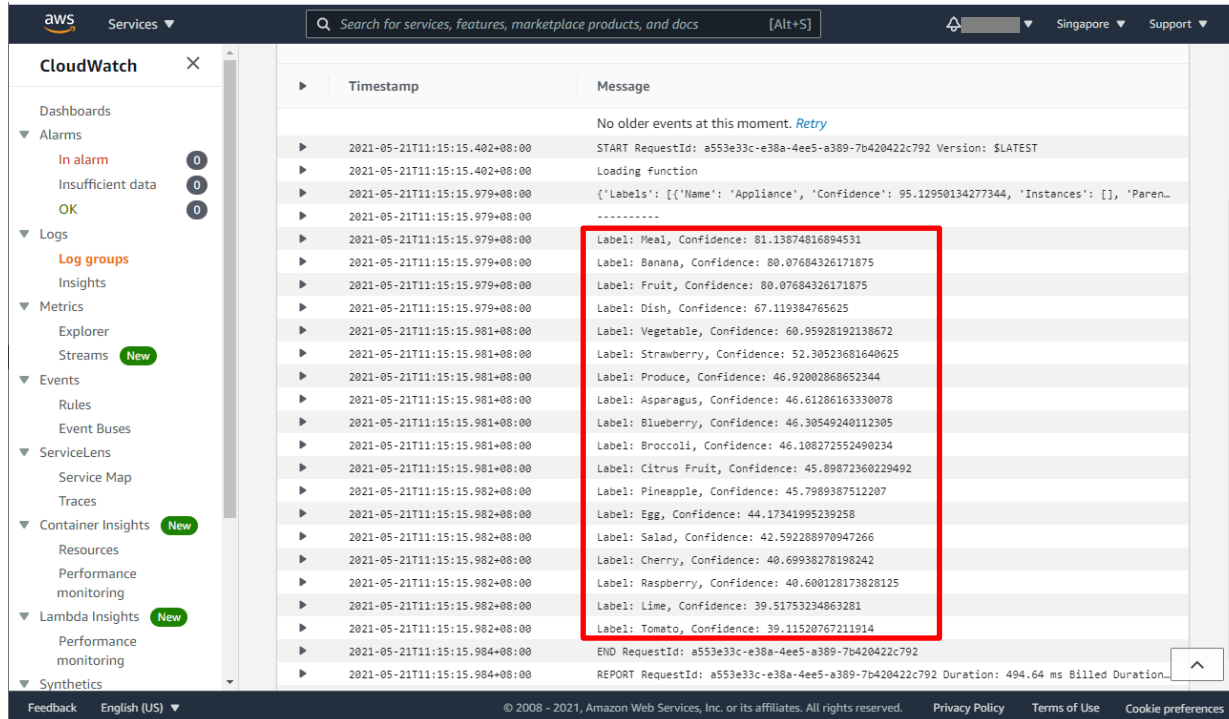


Choose the latest one to see the latest result



The screenshot shows the AWS CloudWatch console. The 'Log streams' tab is selected, displaying a list of 28 log streams. The first log stream, '2021/05/21/[\$LATEST]e26774dfbdd540c994efce3d0a3...', is highlighted with a red box. The 'Last event time' for this stream is '2021-05-21 11:15:15 (UTC+08:00)'.

Here is the object detection results from Rekognition!



The screenshot shows the AWS CloudWatch console with the 'Log streams' tab selected. The log stream '2021/05/21/[\$LATEST]e26774dfbdd540c994efce3d0a3...' is selected, and the 'Messages' tab is active. The log messages show the results of an object detection request. The results are highlighted with a red box, showing a list of labels and their confidence scores:

Label	Confidence
Meat	81.13674616894531
Banana	80.07684326171875
Fruit	80.07684326171875
Dish	67.119384765625
Vegetable	60.95928192138672
Strawberry	52.38523681640625
Produce	46.92002868652344
Asparagus	46.61286163330078
Blueberry	46.30549240112305
Broccoli	46.108272552490234
Citrus Fruit	45.89872360229492
Pineapple	45.7989387512207
Egg	44.17341995239258
Salad	42.592288970947266
Cherry	40.69938278198242
Raspberry	40.600128173828125
Lime	39.51753234863281
Tomato	39.11520767211914

According to AWS document, the bounding boxes are returned for common object labels such as people, cars, furniture, apparel or pets.

https://boto3.amazonaws.com/v1/documentation/api/1.9.96/reference/services/rekognition.html#Rekognition.Client.detect_labels