Table of Contents

[1 Lambda function 1](#_Toc535686613)

[1.1 Adding a new Lambda function 1](#_Toc535686614)

[1.2 Updating the existing Lambda function 3](#_Toc535686615)

[2 AWS IoT core 4](#_Toc535686616)

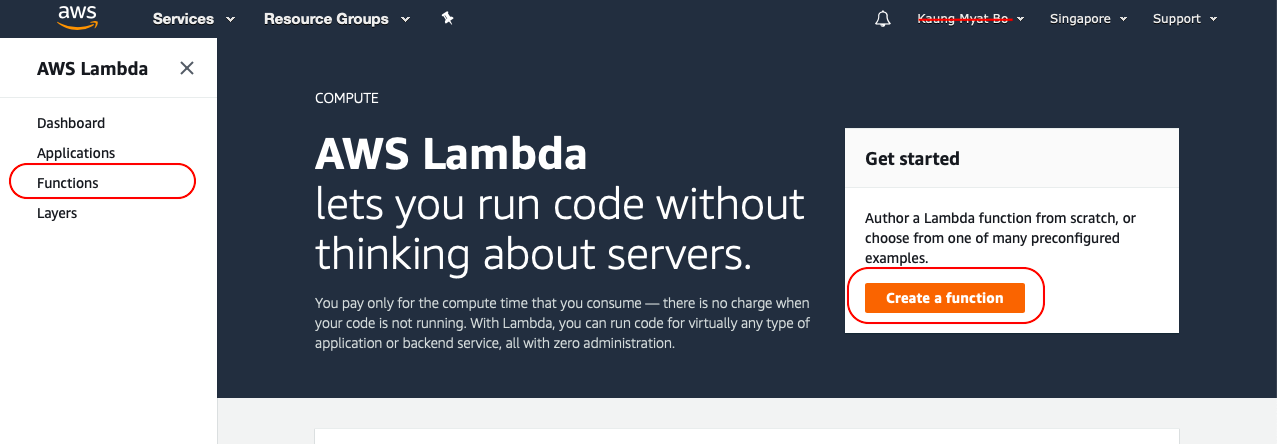
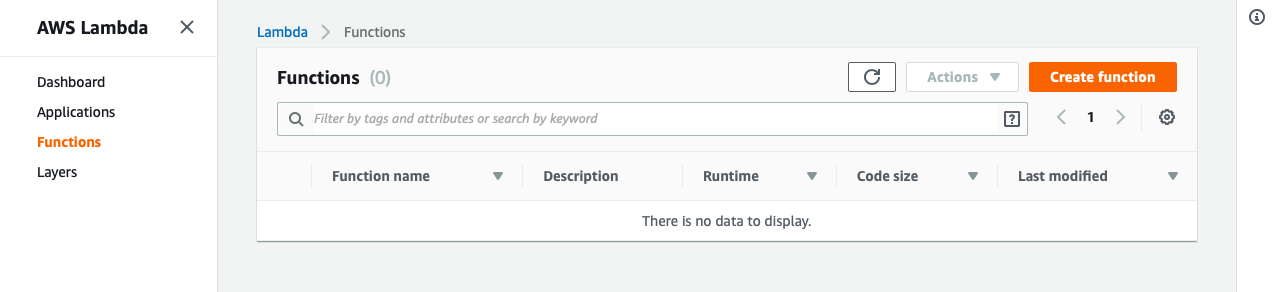
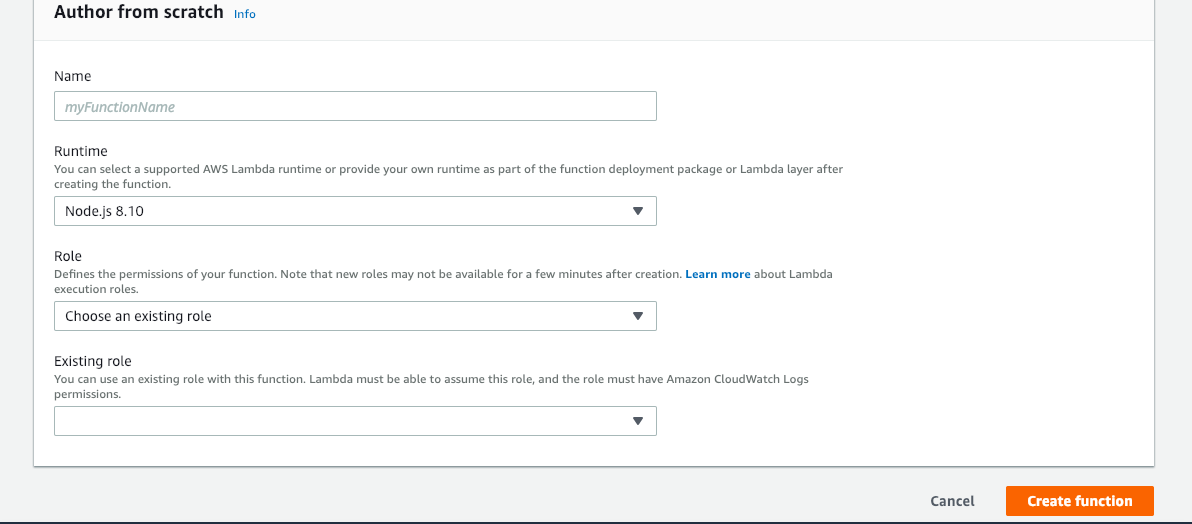
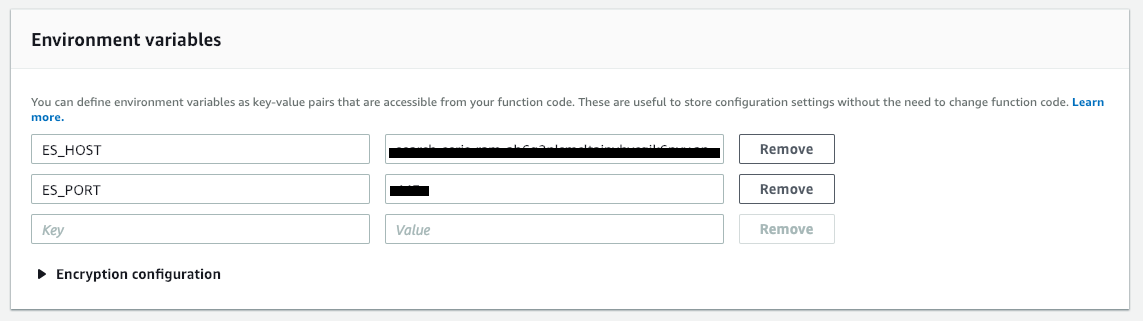
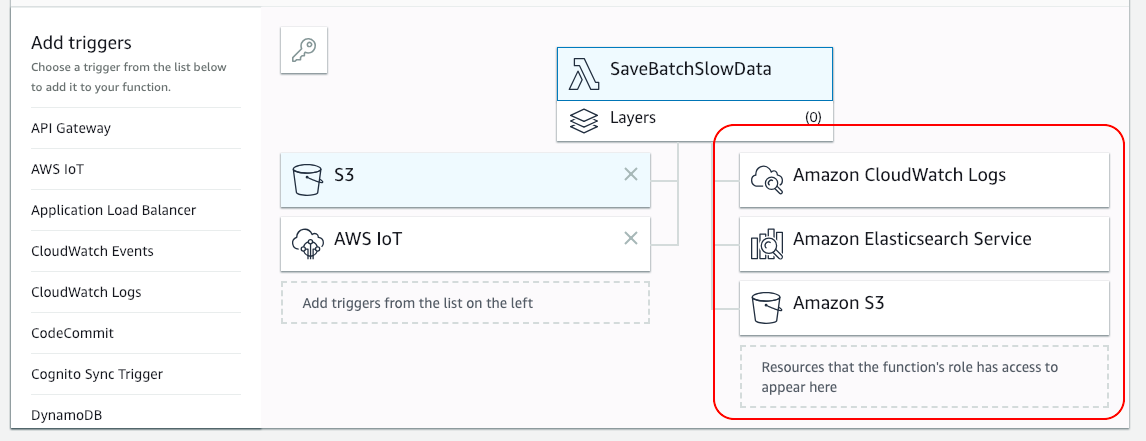
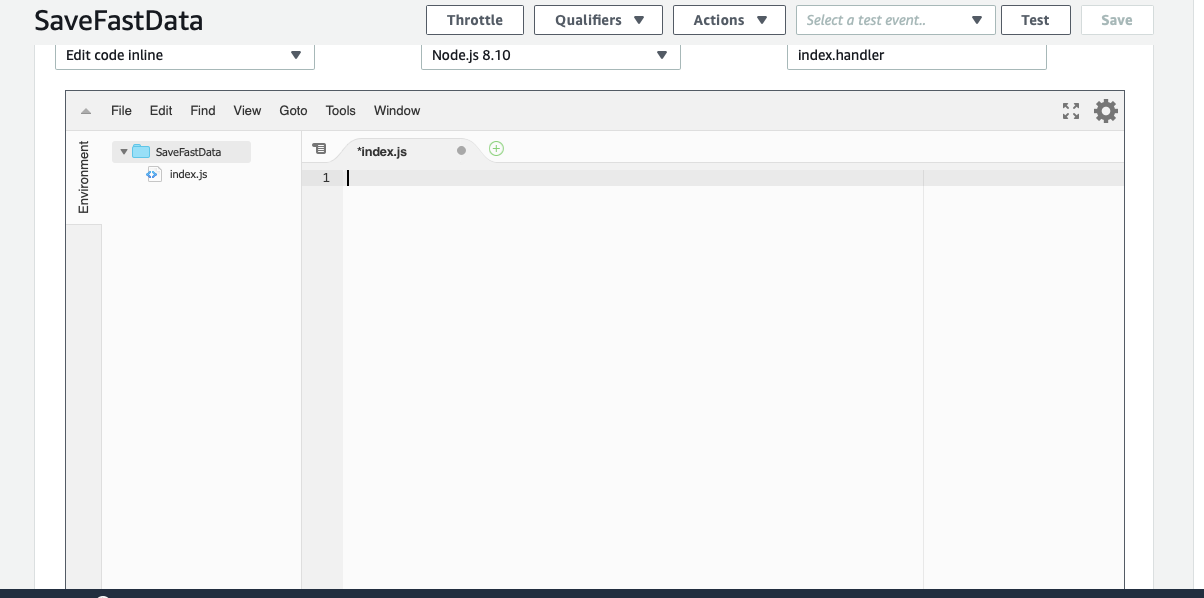
[2.1 Adding a thing 4](#_Toc535686617)

[2.2 Create a rule for incoming topic 7](#_Toc535686618)

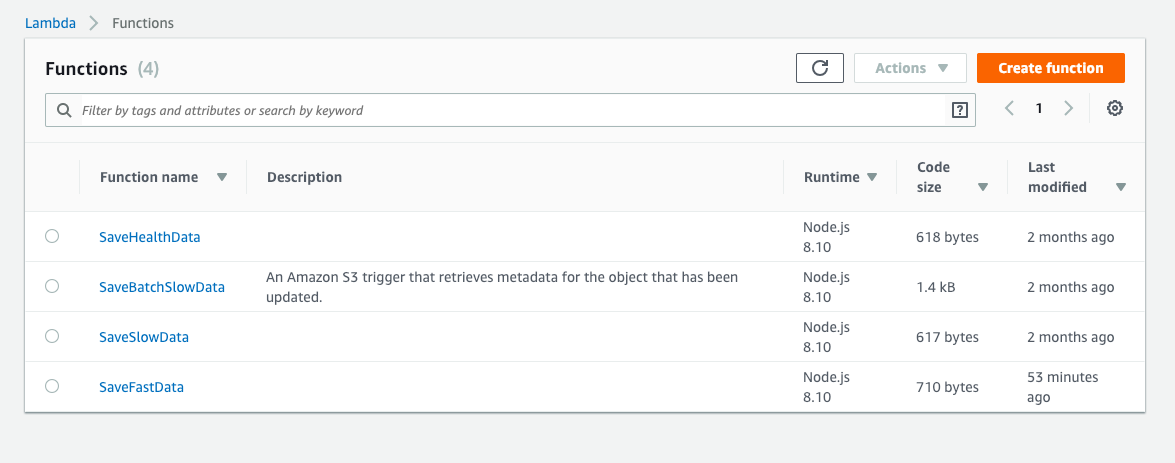
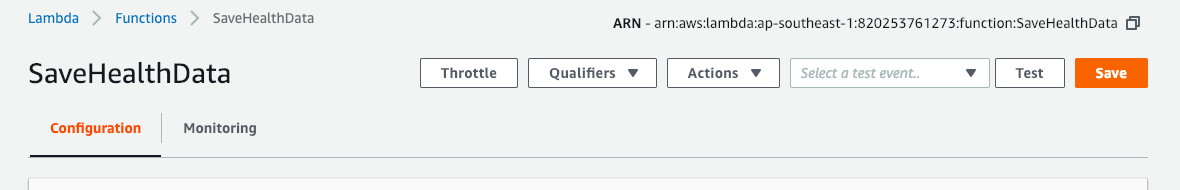
[3 AWS S3 bucket 12](#_Toc535686619)

# Lambda function

## Adding a new Lambda function

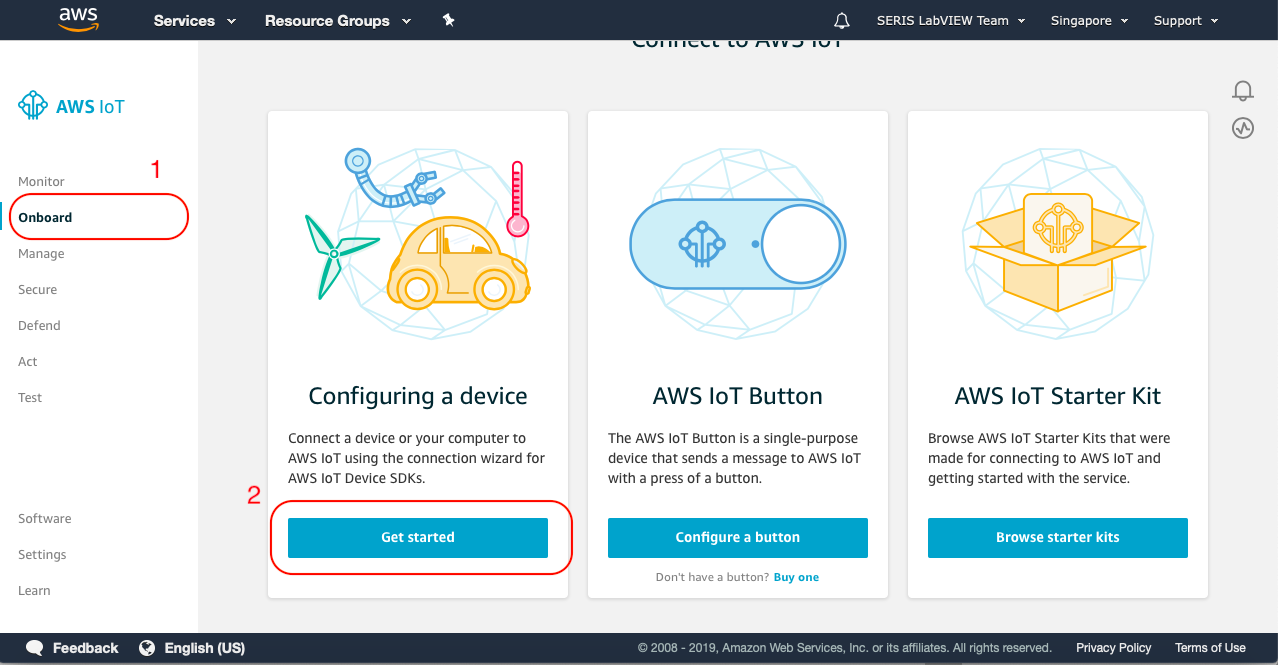
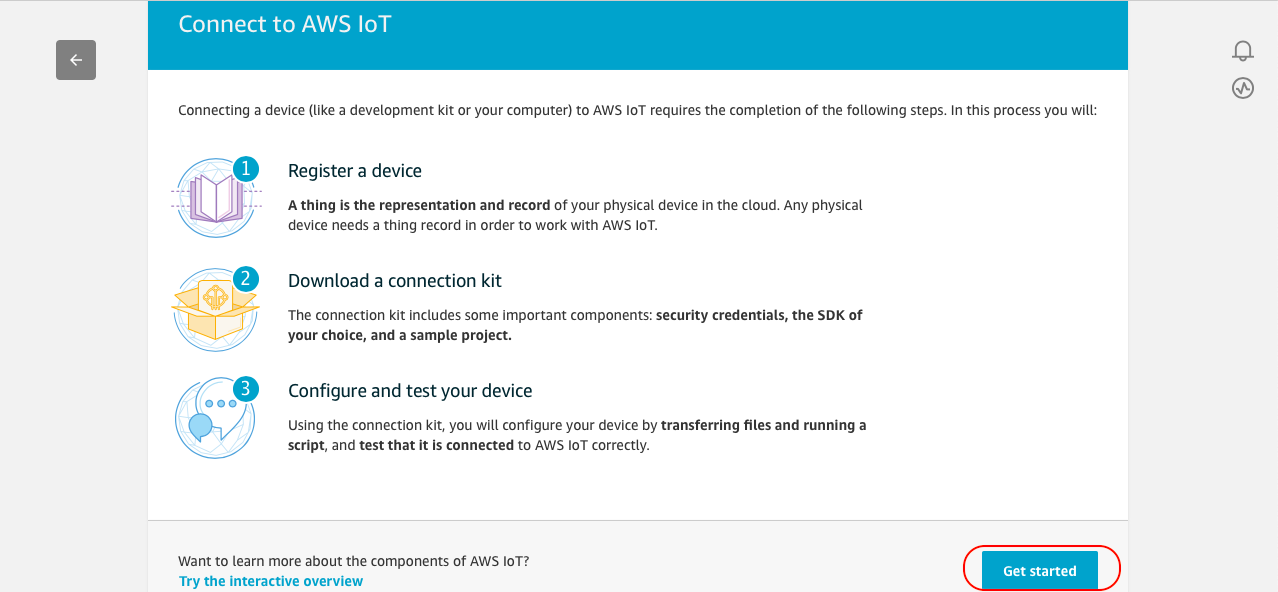
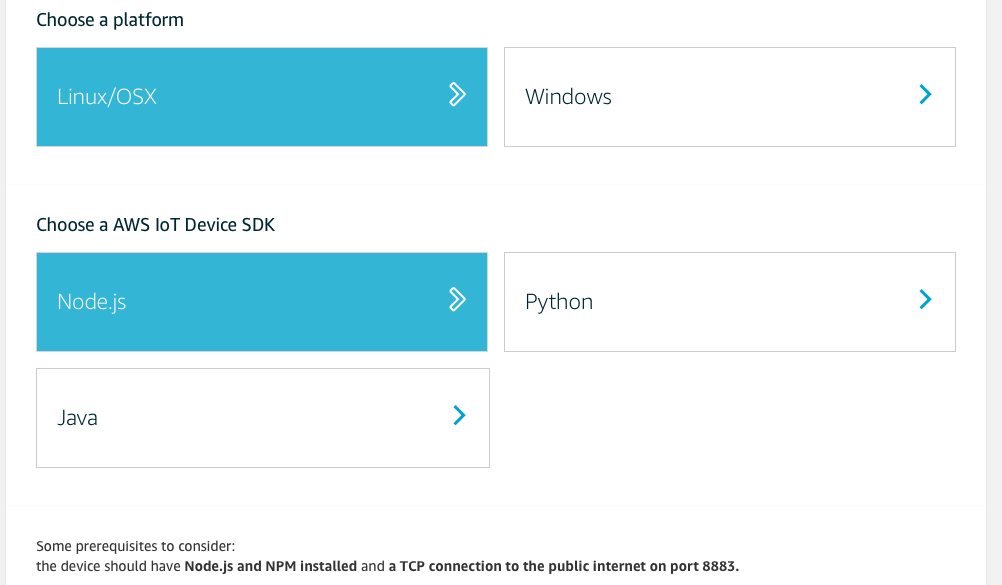
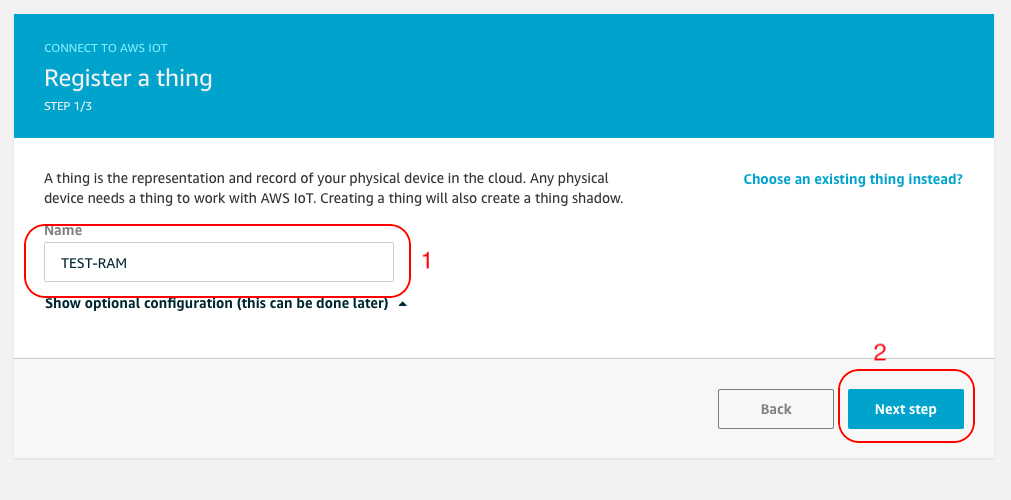
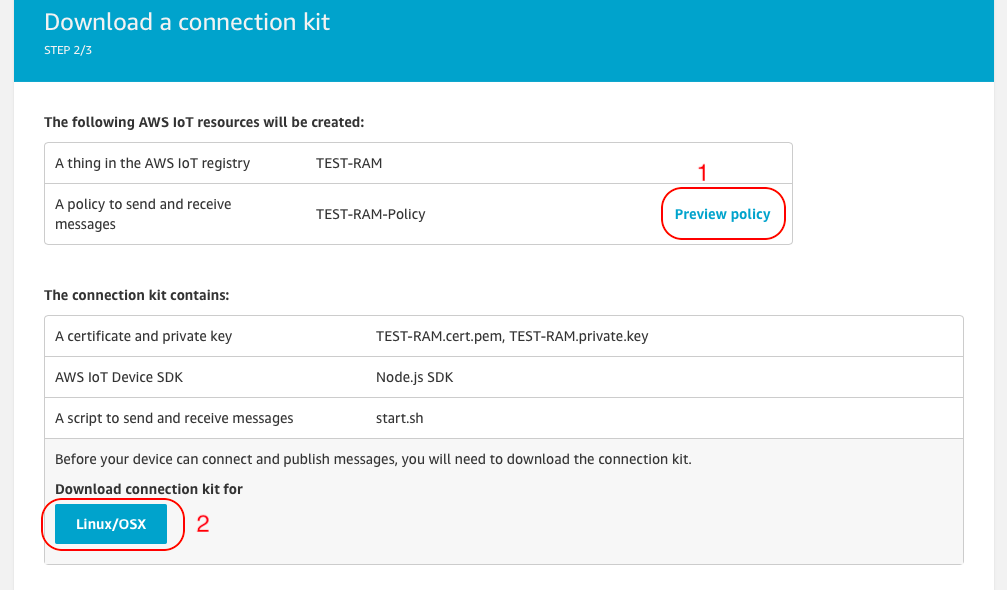
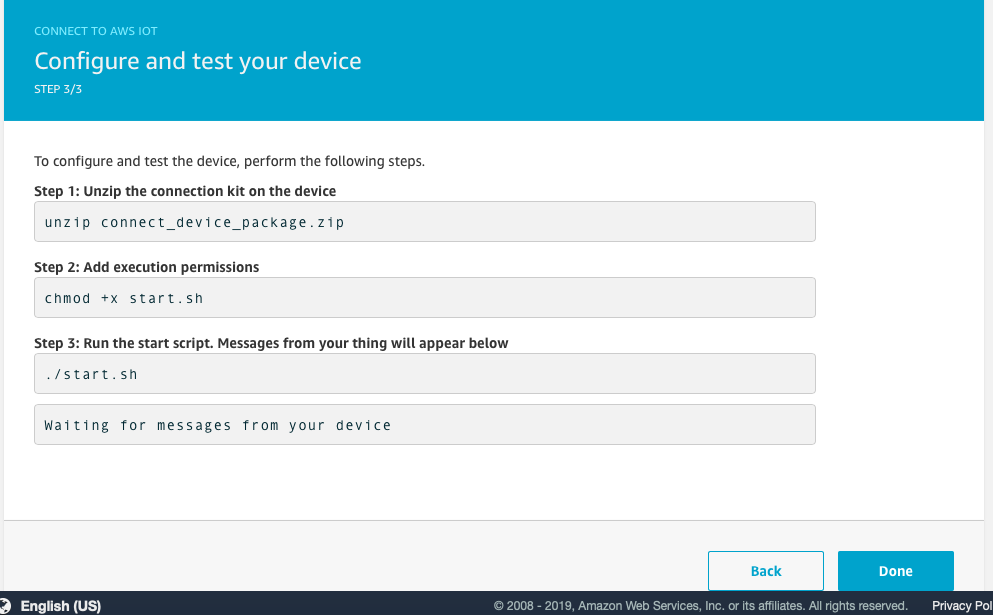
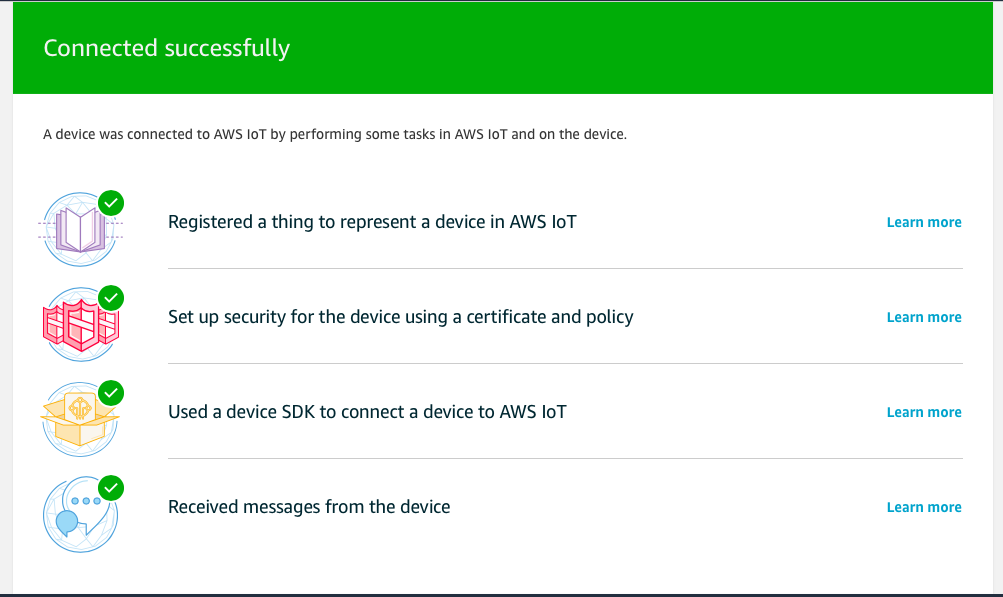
1. Navigate to Lambda service page. Click “Functions” menu item or “Create a function” button straight away to create a new function.  
   
2. Click “Create function” button if “Create a function” is not clicked in previous step.  
   
3. Fill the function name and select Role. Then click “Create function”.  
   
4. Add ES\_HOST and ES\_PORT environment variables to connect to Elasticsearch database. These environment variables will be used in NodeJs code.  
   
5. Add necessary roles for the Lambda function  
   
6. Implement the NodeJs code below and click “Save” button.  
   

## Updating the existing Lambda function

1. Select an existing function that needs to be modify.  
   
2. Click “Save” after modification.  
   

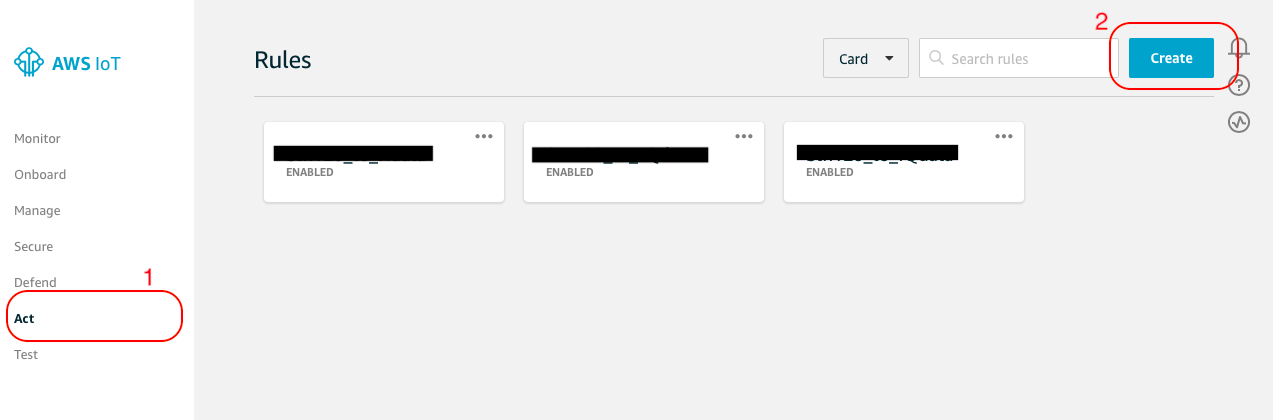
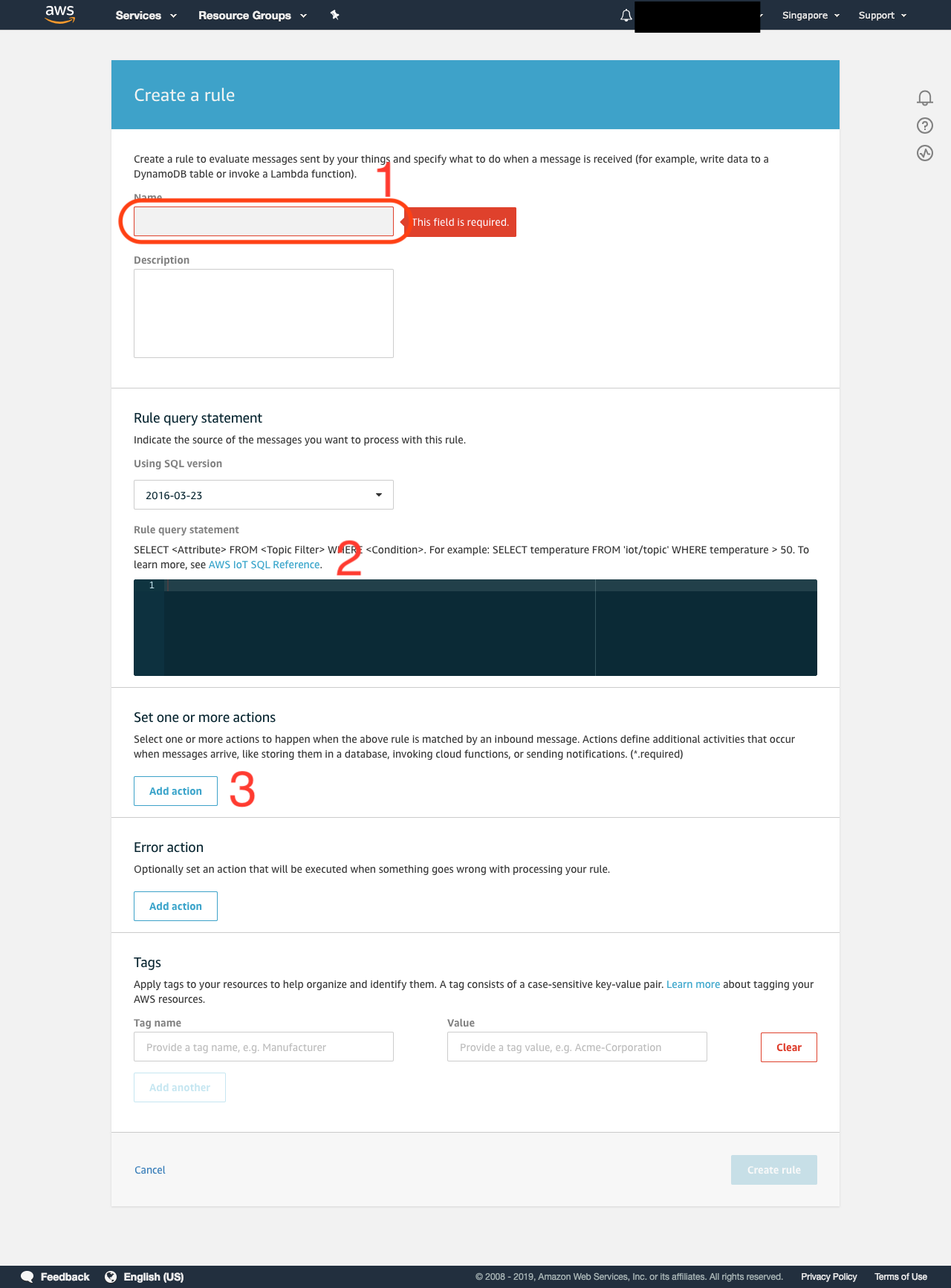
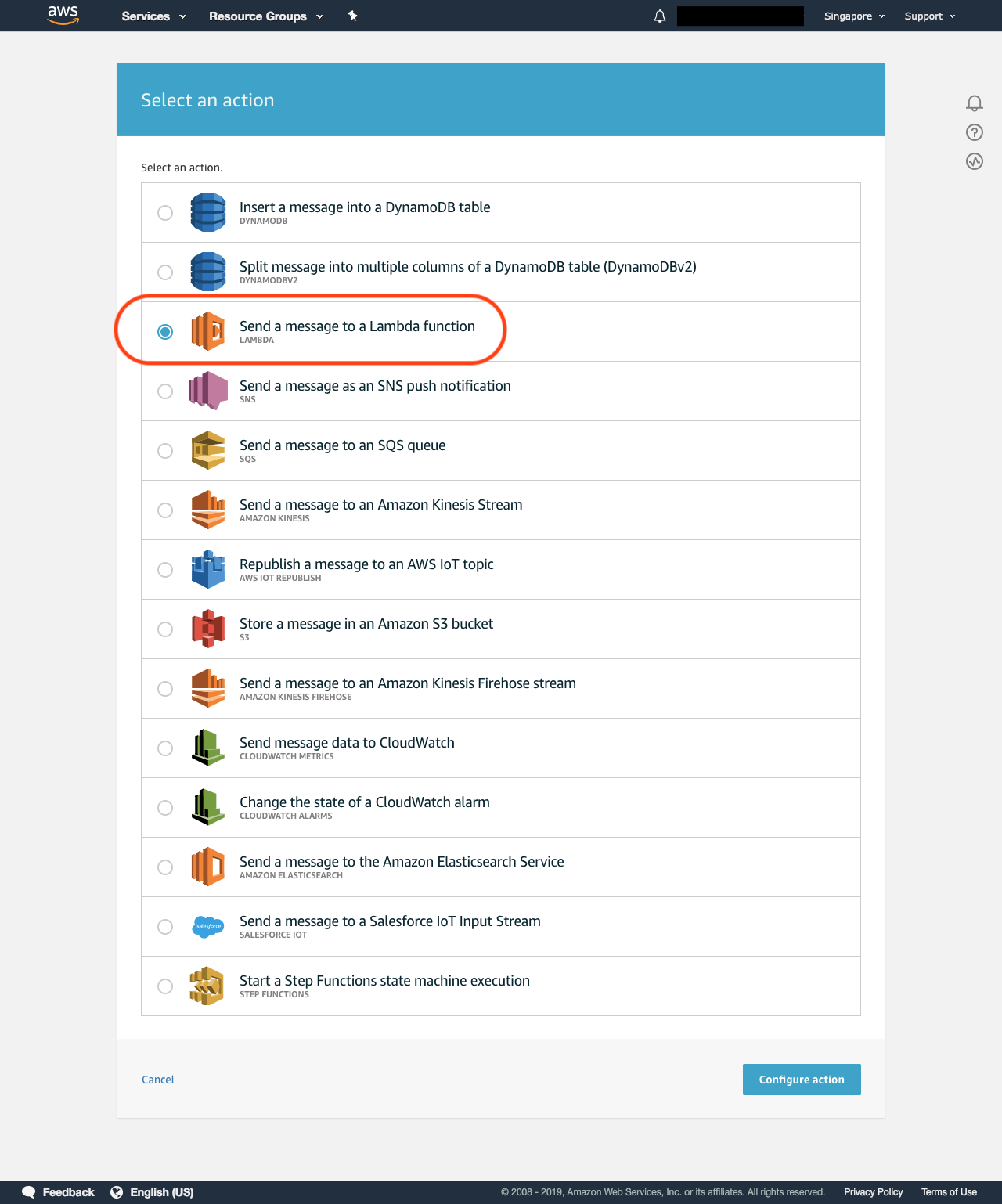
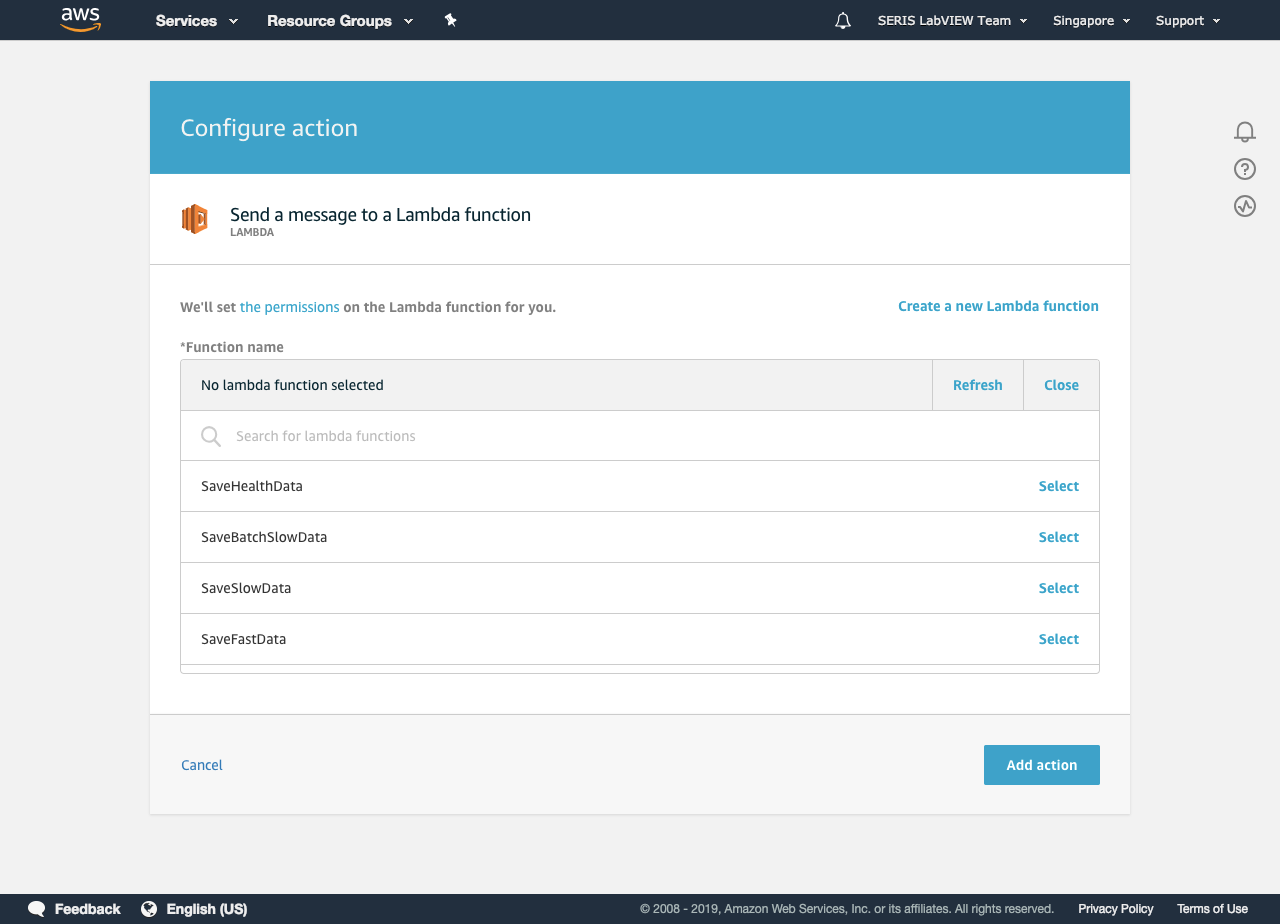
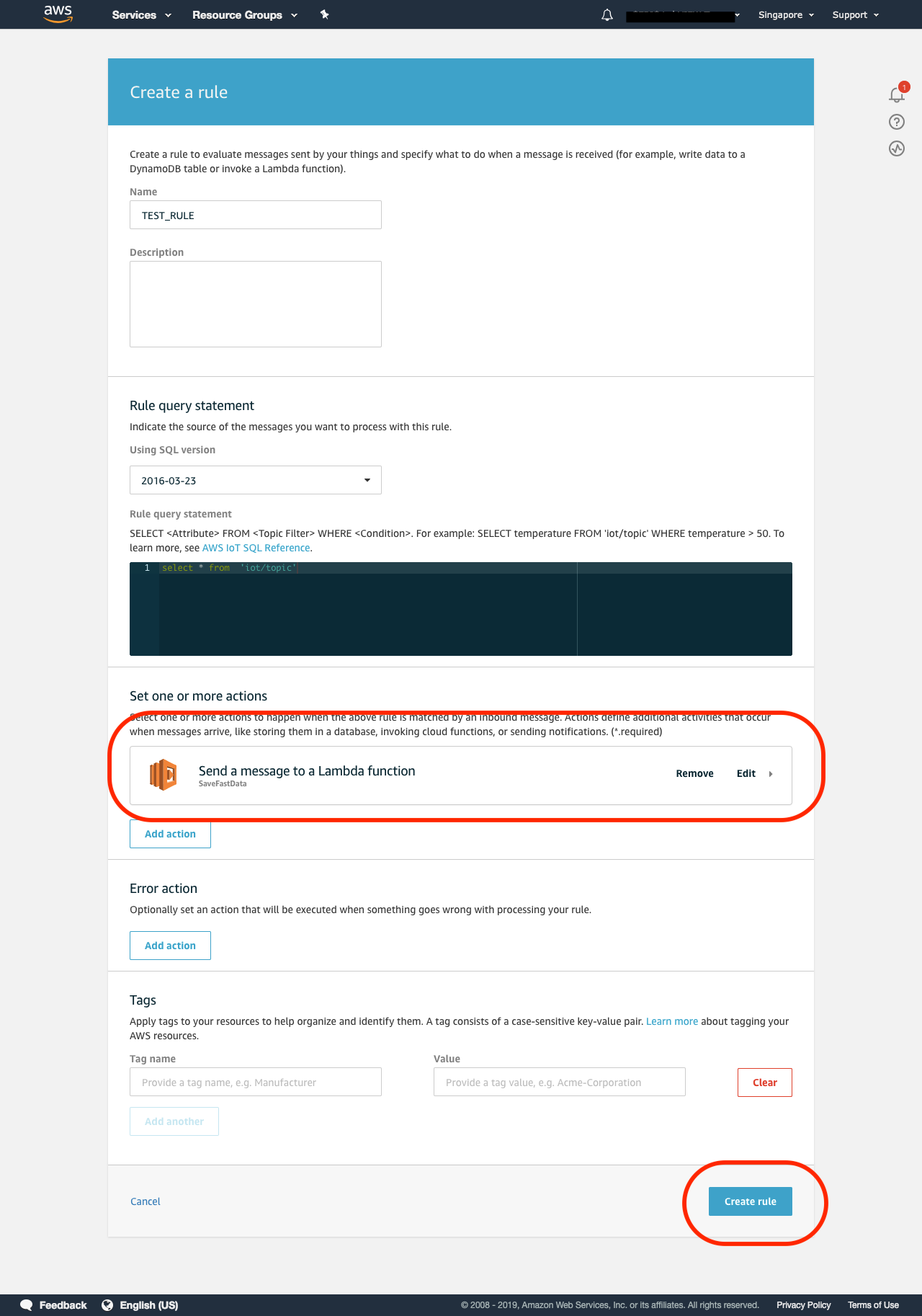
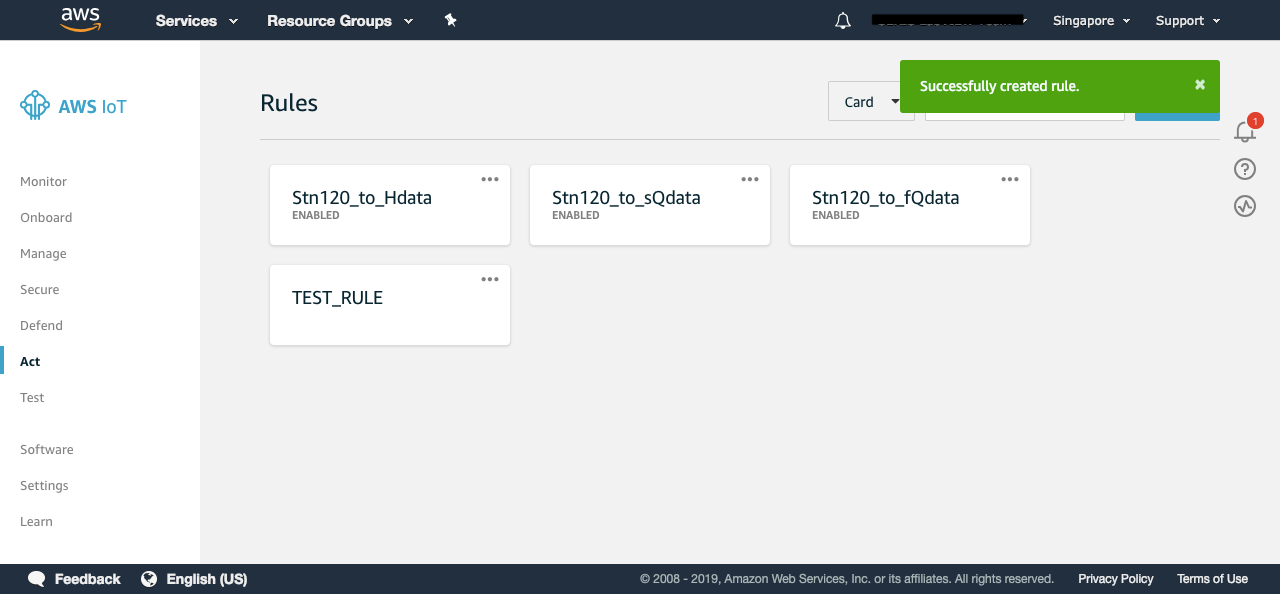
# AWS IoT core

## Adding a thing

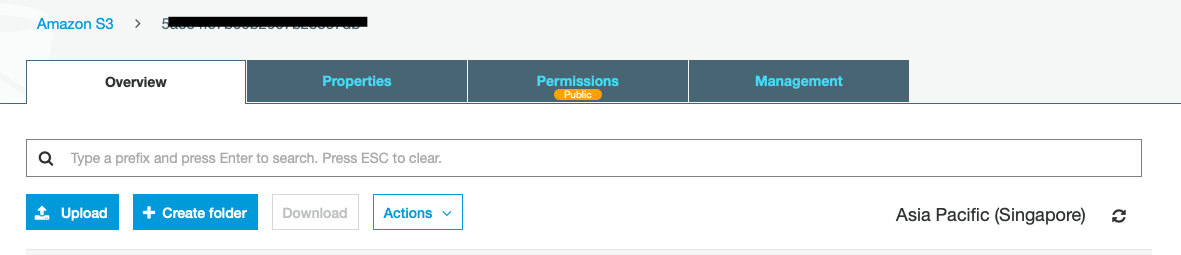
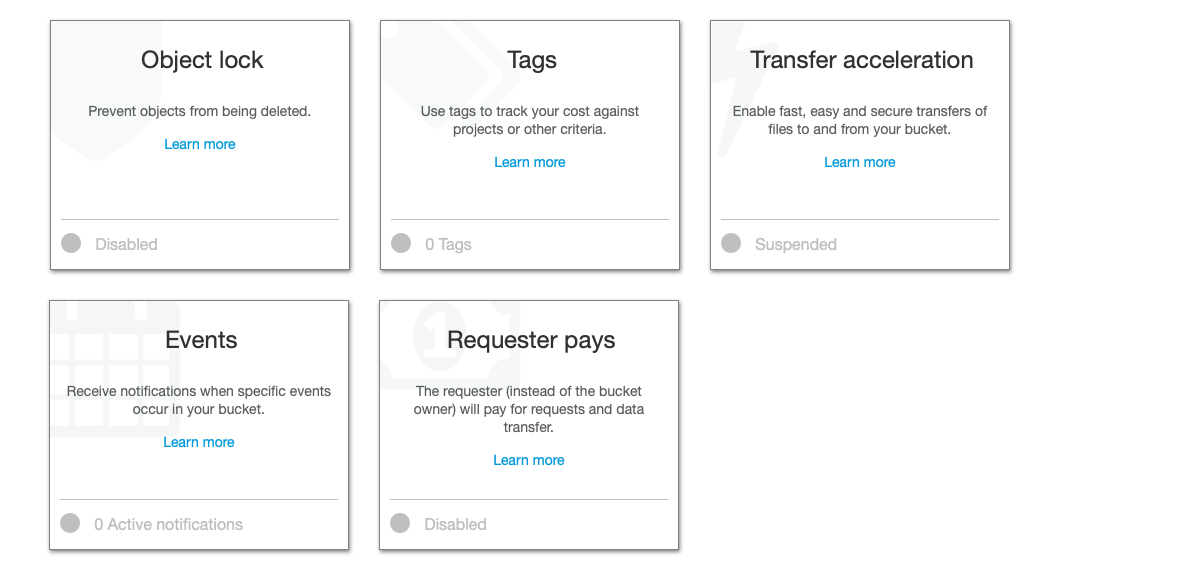
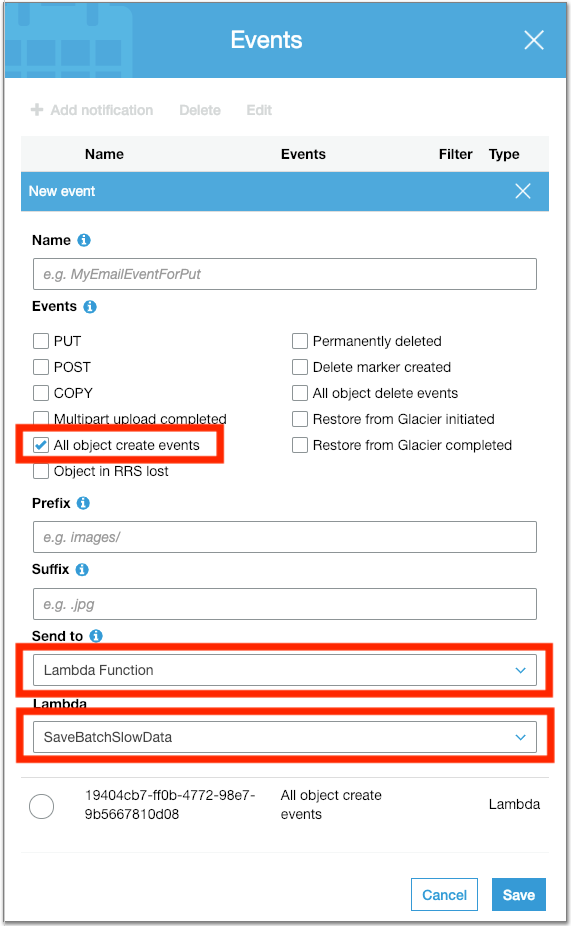
1. Navigate to IoT core service page. Click “Onboard” menu item and select “Get started” option.  
   
2. Click “Get started”  
   
3. Select a platform and choose a relevant SDK and then click “Next”.  
   
4. Fill name for a thing and click “Next step”.  
   
5. Check the policy and download connect kit. Then click “Next step”.  
   Policy will describe which topics are allow to publish or subscribe which is very important for a developer. Topics which are not allowed in this policy will not be accepted by IoT core.  
   
6. This step shows how to run example kit which is downloaded in step 5. After everything, click “Done” and an IoT thing is successfully created.  
   
7. Success screen.  
   

## Create a rule for incoming topic

Section 1.1 must be completed before this section.

1. Click “Act” menu item and click “Create” button.  
   
2. Fill a rule name, rule query statement and click “Add action” button.  
   
3. Select “Send a message to a Lambda function” for an action of this rule and click “Configure action”.  
   
4. Select the Lambda function created at [Section 1](#_Lambda_function) and click “Add action” button.  
   
5. A newly created Lambda action will be displayed in a rule creation form. Then click “Create rule” button at the bottom of the form.  
   
6. New created rule will de appeared on rule screens as below.  
   

# AWS S3 bucket

1. Create a S3 bucket for slow-data batch file. Click “Properties” tab.  
   
2. Click “Events” and click “Add notification”  
   
3. Check “All object create events”, select “Lambda Function” item for “Send to” dropdown, select the Lambda function name which is created for batch data processing and click “Save”.  
   

# Configure the application properites

## Sepcify the app server’s directory in src\main\resources\application.properties

#download file path

download.path=/home/ubuntu/seris-download/

#email templates folder path

email.template.folder=/home/ubuntu/seris-email-templates/

## Copy the email templates

Copy the directory email-templates from the source code to the “email.template.folder”

# Deploying web application

## Install maven

You may refer to the online instruction

<https://maven.apache.org/install.html>

## build the code and upload the app server by the command below

cd <source code root directory>

mvn clean package -DskipTests

scp -i ../ec2-13-250-30-61.ap-southeast-1.pem target/seris-rams-0.1.0.jar [ubuntu@seris-ram.dyndns-ip.com:/home/ubuntu](mailto:ubuntu@seris-ram.dyndns-ip.com:/home/ubuntu)

p.s. pem file is the key which get from aws server

## connection to ec2 via ssh

ssh -i ec2-13-250-30-61.ap-southeast-1.pem [ubuntu@seris-ram.dyndns-ip.com](mailto:ubuntu@seris-ram.dyndns-ip.com)

## run the java as backend progress

nohup java -jar seris-rams-0.1.0.jar &

## last step is optional

if you want to kill the process

ps –ef | grep “seris-rams-0.1.0.jar”

kill -9 <pid>