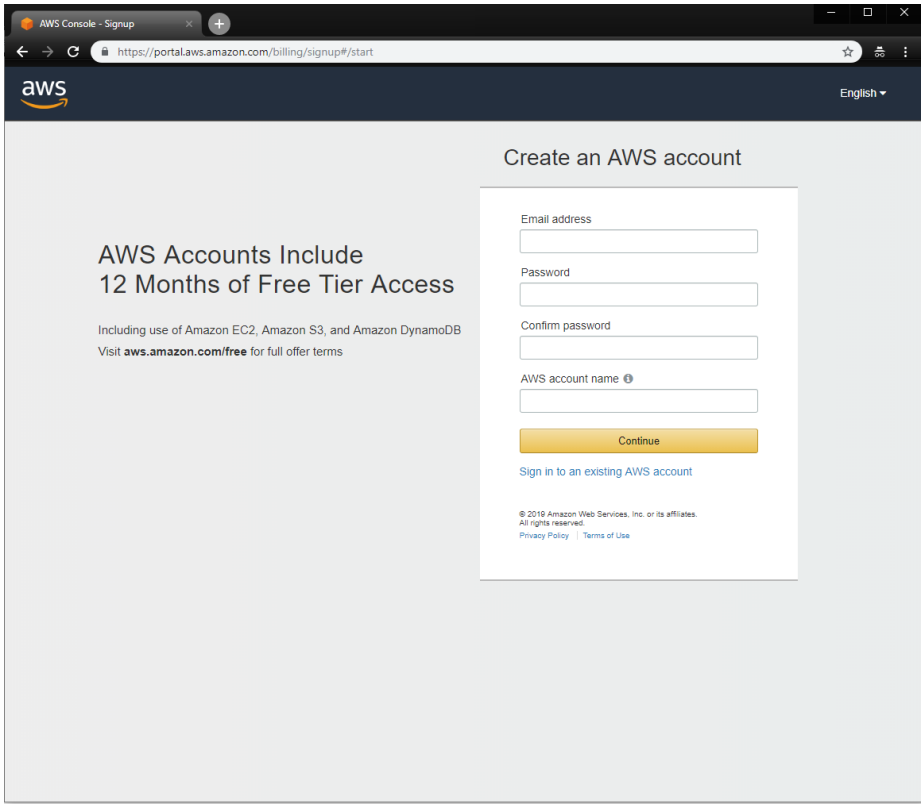


aws-greengrass-quickstart

AWS IoT Greengrass software is readily available on PHYTEC's phyBOARD-Zeta to enable cloud capabilities. Greengrass allows data to be collected, analyzed, and acted upon at the source, while using the cloud for storage, analytics, and securely connecting to other devices. Code can be deployed to individual SOMs or to device fleets through the cloud in AWS Lambda functions via Greengrass, facilitating the process of remote software updates and certificate rotation. **This quick-start will demonstrate how to deploy a simple "Hello World" Lambda function to the phyBOARD-Zeta and view the results in the AWS IoT Console.**

This tutorial is based off of "Getting Started with AWS IoT Greengrass" in the AWS IoT Greengrass Developer Guide, which can be found here: <https://docs.aws.amazon.com/greengrass/latest/developerguide/gg-gs.html>.

	Step	Image
--	------	-------

1	<p>Create or log into your AWS account. Here is the current link to the console sign-in: https://portal.aws.amazon.com/billing/signup#/start</p>	
2	<p>Connect the phyBoard-Zeta to the Internet. If this is the first time booting the board you will need to bring up the ethernet interfaces:</p> <ul style="list-style-type: none"> • If using ethernet, power on the phyBO ARD-Zeta, boot into Linux and log in • Run the listed commands to set up the Internet <ul style="list-style-type: none"> ◦ (modify to the ethernet port that you're using) 	<pre>ifconfig eth0 up udhcpc -i eth0</pre> <p># To prevent commands from wrapping over themselves:</p> <pre>shopt -s checkwinsize && resize</pre>
3		

Run the following commands to make sure everything is correctly configured for Greengrass:

```
mkdir ~/Downloads && cd ~/Downloads
wget https://github.com/aws-samples/aws-greengrass-samples/raw/master/greengrass-dependency-checker-GGCv1.7.1.zip
unzip greengrass-dependency-checker-GGCv1.7.1.zip
cd greengrass-dependency-checker-GGCv1.7.1
sudo modprobe configs
chmod +700 check_ggc_dependencies
sudo ./check_ggc_dependencies | more
```

If there are missing dependencies beyond Node v6.10 and Java 8, then the Greengrass option was not correctly configured in the image build. [Try rebuilding the image](#) and then proceed to the next step when all of the required dependencies are configured and installed.

```

=====Checking script dependencies=====
The device has all commands required for the script to run

=====Dependency check report for GGC v=====
System configuration:
Kernel architecture: armv7l
Init process: /lib/systemd/systemd
Kernel version: 4.9
C library: GNU libc
C library version: 2.24
Directory /var/run: Present
/dev/stdin: Found
/dev/stdout: Found
/dev/stderr: Found

-----Commands and software packages-----
Python version: 2.7.12
NodeJS version: 6.10.3
Java version: 1.8.0_102
OpenSSL version: 1.0.2
wget: Present
realpath: Present
tar: Present
readlink: Present
basename: Present
dirname: Present
pidof: Present
df: Present
grep: Present
umount: Present

-----Platform security-----
Hardlinks_protection: Enabled
Symlinks protection: Enabled

-----User and group-----
ggc_user: Present
ggc_group: Present

-----[Optional] Greengrass container dependencies-----

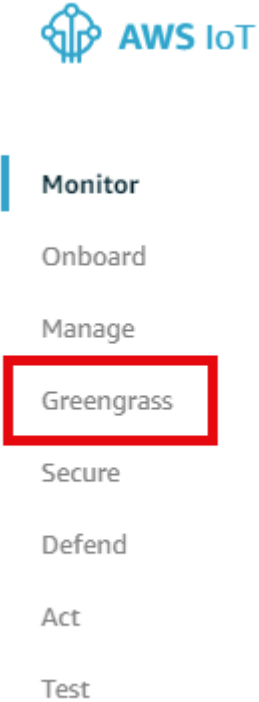
-----Kernel configuration-----
Kernel config file: /proc/config.gz

Namespace configs:
CONFIG_IPC_NS: Enabled
CONFIG_UTS_NS: Enabled
CONFIG_USER_NS: Enabled
CONFIG_PID_NS: Enabled

Cgroup configs:
CONFIG_CGROUP_DEVICE: Enabled
CONFIG_CGROUPS: Enabled
CONFIG_MEMCG: Enabled

Other required configs:
CONFIG_POSIX_MQUEUE: Enabled
CONFIG_OVERLAY_FS: Enabled
CONFIG_HAVE_ARCH_SECCOMP_FILTER: Enabled

```

		<pre>CONFIG_SECCOMP_FILTER: Enabled CONFIG_KEYS: Enabled CONFIG_SECCOMP: Enabled CONFIG_SHMEM: Enabled --More--</pre>
4	<p>Create a Greengrass group.</p> <ul style="list-style-type: none">• On the AWS IoT console, click "Greengrass" on the left• Click "Create a group".	 <p>The screenshot shows the AWS IoT console interface. At the top, there is the AWS IoT logo. Below it, a vertical sidebar contains several menu items: Monitor, Onboard, Manage, Greengrass, Secure, Defend, Act, and Test. The 'Greengrass' menu item is highlighted with a red rectangular box.</p>

5

- Select **"use easy creation"** to automatically generate certificates for the phyBoard-Zeta to connect to AWS Greengrass
- Select **"provision a core"** in the IoT Registry

Set up your Greengrass Group

Setting up your Group requires you to provision a Core device in the IoT Registry, acquire a role to your Group. If you're unfamiliar with any of these steps we recommend the easy Greengrass software on your Core device.

Easy Group creation (recommended)


This process will automatically provision a Core in the registry, use default settings to create a new Group, and provide your Core with a new certificate and a key pair.


Advanced Group creation

This customizable process will take you step-by step through the Core provisioning and you to customize the IAM Role for your Group and the certificate for your Core, and provision a key pair.

6	Name your group in the "Group Name" box and click "Next".	<div data-bbox="418 191 808 216">SET UP YOUR GREENGRASS GROUP</div> <div data-bbox="418 237 797 289"><h2>Name your Group</h2></div> <p data-bbox="418 422 1495 491">The Greengrass Group is a cloud-configured managed collection of local devices and I with each other through a Core device. Groups can contain up to 200 local devices.</p> <div data-bbox="418 527 583 552">Group Name</div> <div data-bbox="449 590 620 615" data-cs="2" data-kind="parent">MyFirstGroup</div> <div data-kind="ghost"></div>
7	Use the name provided for the core, and click "Next".	<div data-bbox="418 982 808 1008">SET UP YOUR GREENGRASS GROUP</div> <div data-bbox="418 1029 1219 1081"><h2>Every Group needs a Core to function</h2></div> <p data-bbox="418 1213 1495 1283">Every Greengrass Group requires a device running Core software. It enables communi cloud computing services. Adding information to the Registry is the first step in provi</p> <div data-bbox="418 1318 496 1344">Name</div> <div data-bbox="449 1381 696 1407" data-cs="2" data-kind="parent">MyFirstGroup_Core</div> <div data-kind="ghost"></div> <div data-bbox="418 1463 1143 1488">Show optional configuration (this can be done later) ▼</div>

8	Click "Create Group and Core" to finalize creating the Greengrass Group (AWS-side resources for the your set of devices).	<div>SET UP YOUR GREENGRASS GROUP</div> <div>Run a scripted easy Group creation</div> <p>In order to speed up and simplify Group creation AWS IoT Greengrass will handle the following processes and use default settings. By proceeding to the next step, you are giving permission for us to complete the following steps.</p> <p>AWS IoT Greengrass will take these actions on your behalf using default settings:</p> <table><tr><td>Create a new Greengrass Group in the cloud</td></tr><tr><td>Provision a new Core in the IoT Registry and add to the Group</td></tr><tr><td>Generate public and private key set for your Core</td></tr><tr><td>Generate a new security certificate for the Core using the keys</td></tr><tr><td>Attach a default security policy to the certificate</td></tr></table>	Create a new Greengrass Group in the cloud	Provision a new Core in the IoT Registry and add to the Group	Generate public and private key set for your Core	Generate a new security certificate for the Core using the keys	Attach a default security policy to the certificate			
Create a new Greengrass Group in the cloud										
Provision a new Core in the IoT Registry and add to the Group										
Generate public and private key set for your Core										
Generate a new security certificate for the Core using the keys										
Attach a default security policy to the certificate										
9	On the confirmation page, download the certificate package . No need to download the software configuration package. Click "Finish".	<div>Download and store your Core's security resources</div> <table><tr><td>A certificate for this Core</td><td>c6973960cc.cert.pem</td></tr><tr><td>A public key</td><td>c6973960cc.public.key</td></tr><tr><td>A private key</td><td>c6973960cc.private.key</td></tr><tr><td>Core-specific config file</td><td>config.json</td></tr></table> <div>Download these resources as a tar.gz</div>	A certificate for this Core	c6973960cc.cert.pem	A public key	c6973960cc.public.key	A private key	c6973960cc.private.key	Core-specific config file	config.json
A certificate for this Core	c6973960cc.cert.pem									
A public key	c6973960cc.public.key									
A private key	c6973960cc.private.key									
Core-specific config file	config.json									

10	Transfer the certificate package from your computer to the i.MX7.	<p>from Windows:</p> <pre>cd <insert path-to-downloaded-files> pscp -scp <insert hash>-setup.tar.gz root@<insert IP-address>:~/Downloads</pre> <p>from macOS/Unix</p> <pre>cd <insert path-to-downloaded-files> sudo scp <insert hash>-setup.tar.gz root@<insert IP-address>:~/Downloads</pre> <div style="border: 1px solid #f0e68c; padding: 10px; margin-top: 10px;">  See this link for more details regarding the transfer of files: Copying Files to the Device </div>
11	Return to the i.MX7. Decompress the certificate file and copy the root CA to the device. Check that the root.ca.pem isn't empty.	<pre>cd ~/Downloads tar -xzf <insert hash>-setup.tar.gz -C /greengrass cd /greengrass/certs/ wget -O root.ca.pem https://www.amazontrust.com/repository/AmazonRootCA1.pem</pre> <p># Check that the last command was successful</p> <pre>cat root.ca.pem</pre>
12	Start AWS GG daemon on the i.MX7 and check that the daemon is running (there will be a root entry for /greengrass/ggc/packages/1.7.0/bin/daemon).	<pre>cd /greengrass/ggc/core sudo ./greengrassd start</pre> <p># Check that daemon is running</p> <pre>ps aux grep -E 'greengrass.*daemon'</pre>

13	Create the "Hello World" Lambda function. Return to the AWS IoT Core Console and select "Software"	<div><ul style="list-style-type: none">MonitorOnboardManageGreengrassSecureDefendActTestSoftwareSettingsLearn</div>
----	----------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- 14 Download the SDK:
- Scroll down to "**SDKs**" and under "**AWS IoT Greengrass Core SDK**", select "**View all SDKs**".
 - Click "**v1.3.0**" under "**Python 2.7**" to download the AWS IoT Greengrass Core SDK for Python

SDKs



AWS IoT Device SDK

The AWS IoT Device SDK helps you to easily and quickly connect your hardware device to AWS IoT with the Device Gateway and Device Shadow.

[View all SDKs](#)



AWS IoT Greengrass Machine Learning SDK

The AWS IoT Greengrass ML SDK enables the Lambda functions you author to consume the machine learning model available on the device.

[View all SDKs](#)



AWS IoT Greengrass Core SDK

The AWS IoT Greengrass Core SDK enables Lambda functions to interact with the Core on which they run.

[View all SDKs](#)



AWS SDKs

Simplify using AWS services, including AWS IoT, in your applications with an API tailored to your programming language or platform.

[View all SDKs](#)

- 15 Decompress the "greengrass-core-python-sdk-1.3.0.tar.gz" downloaded from the last step.

Windows: install 7-Zip, right click on file, and choose "7-Zip" "Open archive" drag folder to desired location (image shows content)

File Edit View Favorites Tools Help



Add Extract Test Copy Move Delete Info



C:\Users\Demo\Downloads\greengrass-core-python-sdk-1.3.0.tar.gz\greengrass-core-p

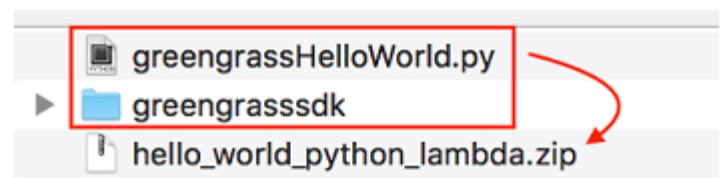
Name	Size	Packed Size	Mod
examples	7 347	8 192	2018
manual	70 450	71 680	2018
sdk	12 949	13 312	2018
LICENSE	10 884	11 264	2018
NOTICE	209	512	2018

macOS: double click the file, open "sdk", and then unzip python_sdk_1_3_0.zip

Unix: use the command line:

```
tar -xzf <path to greengrass-core-python-sdk-1.3.0.tar.gz>/greengrass-core-python-sdk-1.3.0.tar.gz
cd aws_greengrass_core_sdk/sdk
sudo unzip python_sdk_1_3_0.zip
```

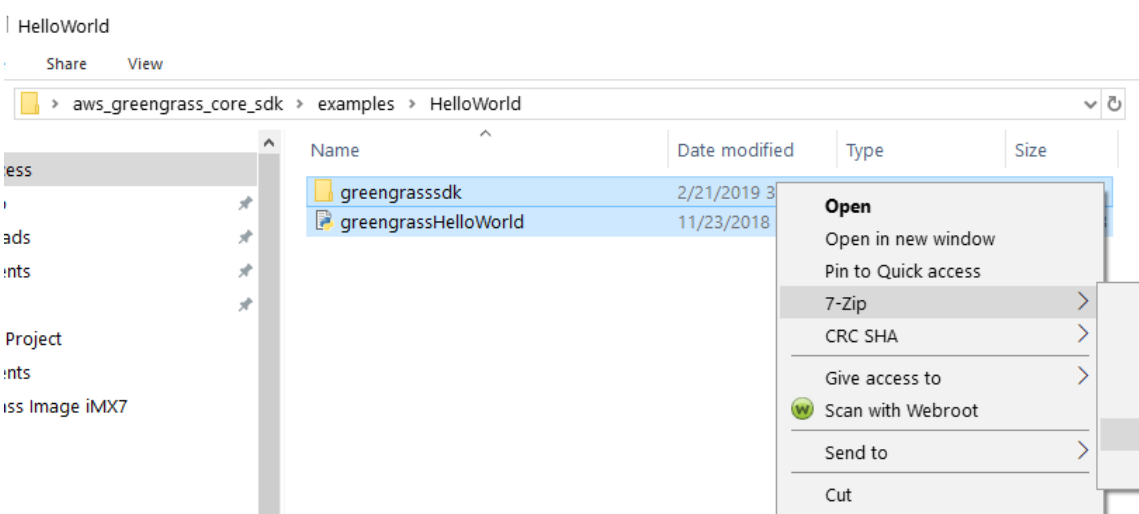
16 Now actually create the Lambda function. Move the **greengrass sdk** folder into the "examples" "HelloWorld" folder and **zip** them together.



Unix (including macOS):

```
sudo zip -r hello_world_python_lambda.zip greengrasssdk greengrassHelloWorld.py
```

Windows:



17 Package the Lambda. On the group page in the AWS IoT Console, select "**Lambda**" "**Add Lambda**" "**Create new Lambda**".

Add a Lambda to your Greengrass Group

Local Lambdas are hosted on your Greengrass Core and connected to each other and devices by Subscription individually to your Group.

Create a new Lambda function

You will be taken to the AWS Lambda Console and can author a new Lambda function.

[Create new Lambda function](#)

Use an existing Lambda function


You will choose from a list of existing Lambda functions.

[Use existing Lambda function](#)[Cancel](#)[Back](#)

- 18 In the AWS Lambda Console, fill out the "Author from Scratch" box like in the following screenshot, and then select "Create function" on the bottom:


Author from scratch

Start with a simple "hello world" example.



Blueprints

Choose a preconfigured template for your Lambda function.



Author from scratch [Info](#)

Name*

Runtime*

Python 2.7 ▼

Role*

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Create new role from template(s) ▼

Lambda will automatically create a role with permissions from the selected policy templates. Note that basic Lambda permissions (logging to CloudWatch) will automatically be added. If your function accesses a VPC, the required permissions will also be added.

Role name*

Enter a name for your new role.

Policy templates

Choose one or more policy templates. A role will be generated for you before your function is created. [Learn more](#) about the permissions that each policy template will add to your role.

▼

19

- Upload the Lambda function deployment package created in **Step 17** (could be named something different)
- **Fill out the fields** like in the image
 - Code entry type: .zip file
 - Runtime: Python 2.7
 - Handler: green.grass.HelloWorld.function_handler
- **Save** the function (upper right hand corner)

Function code Info

Code entry type

Upload a .zip file ▼

Runtime

Python 2.7 ▼

Function package*



Upload

hello_world_python_lambda.zip (12.8 kB)

For files larger than 10 MB, consider uploading using Amazon S3.

20

Publish the Lambda function:

- "Actions" "Publish new version"
- In "Version Description", enter anything (e.g. "First version" and then "Publish"

Lambda > Functions > Greengrass_HelloWorld **ARN** - arn:aws:lambda:us-west-

Greengrass_HelloWorld

Qualifiers ▾ Actions ▾ *Select a test event..*

✓ Congratulations! Your Lambda function "Greengrass_HelloWorld" is now deployed. You can now test your function.

Publish new version

- Create alias
- Delete function
- Export function

Publish new version from \$LATEST

Publishing a new version will save a "snapshot" of the code and configuration of the \$LATEST version. You can then edit the new version's code. Please click to confirm.

Version description

First version

21

Create an alias/
nickname for
the Lambda
function:

- "Actions"
"Create
alias"
- Enter a
name (e.g.
"GG_Hello
World")
- Set version
to "1"
- Click "Crea
te"

Version: 1 ▾

Actions ▾

Select a test event..

Publish new version

Create alias

Delete version

Export function

Create a new alias

An alias is a pointer to one or two versions. Choose each version that you want the alias to point to.

Name*

GG_HelloWorld

Description

Version*

1 ▾

You can shift traffic between two versions, based on weights (%) that you assign. Click [here](#) to learn more.

Additional version

▾

Cancel

22

Add the Lambda to your GG group to deploy:

- Return to the group created in the AWS IoT Console
- Choose "Use existing Lambda"
- Select the Lambda created in the last step

Add a Lambda to your Greengrass Group

Local Lambdas are hosted on your Greengrass Core and connected to each other and devices by Subscriptions, but they can be deployed individually to your Group.

Create a new Lambda function

You will be taken to the AWS Lambda Console and can author a new Lambda function.

[Create new](#)

Use an existing Lambda function

You will choose from a list of existing Lambda functions.

[Use existing](#)

ADD A LAMBDA TO YOUR GREENGRASS GROUP

Use existing Lambda

Select a Lambda



Greengrass_HelloWorld

Python 2.7

23	<p>Edit the Lambda's configuration for the group:</p> <ul style="list-style-type: none">• For version, choose "Ali as: GG_Hello World", or whatever alias was chosen for the Lambda• Set the settings as the ones in the lower image• Choose "Update"	<div><div>GREENGRASS GROUP</div><div>MyFirstGroup</div><div>Not deployed</div></div> <div><div>Deployments</div><div>Subscriptions</div><div>Cores</div><div>Devices</div><div>Lambdas</div><div>Resources</div><div>Connectors</div><div>Settings</div></div> <div><div>Lambdas</div><div>Greengrass_HelloWorld</div><div>LAMBDA FUNCTION</div></div> <div><div>Memory limit</div><div>16</div><div>MB</div></div> <div><div>Timeout</div><div>25</div><div>Second</div></div> <div><div>Lambda lifecycle</div><div><div><input type="radio"/> On-demand function</div><div><input checked="" type="radio"/> Make this function long-lived and keep it running indefinitely</div></div></div>
24	<p>Create a subscription to the MQTT topic:</p>	

- Choose "Subscriptions"
- For "Select a source", choose "Select"
- Choose the "Lambdas" tab and then "Greengrass_HelloWorld"
- For "Select a target" choose "Select"
- Choose the "Services" tab, and then choose "IoT Cloud", "Next"
- In the "Topic filter" section, type "hello/world", and then hit "Next"

Deployments

Subscriptions

Cores

Devices

Lambdas

Resources

Connectors

Settings

Subscriptions



Want to connect assets in your Group?

Greengrass Cores can pass messages between Devices, Lambda functions and even AWS IoT Cloud. These components can interact using Subscriptions, which are pre-defined topics for security and predictable interactions.

[Learn about Subscriptions](#)
[Add your first Subscription](#)

Select a source



Greengrass_HelloWorld

LAMBDA

Services

Devices

Lambdas



Connectors



Search



Greengrass_HelloWorld

		<div><div>Source</div><div><div> Greengrass_HelloWorld</div><div>LAMBDA</div></div></div> <div><div>Topic filter</div><div>hello/world</div></div> <div><div>Target</div><div><div> IoT Cloud</div><div>SERVICE</div></div></div>
25	<div>Deploy the Lambda:</div> <ul style="list-style-type: none">• Select "Actions" "Deploy"<ul style="list-style-type: none">◦ The first time may take awhile◦ If deployment takes 5 minutes or more, try:<ul style="list-style-type: none">■ "Actions" "Reset Deployments"	<div><div>GREENGRASS GROUP</div><div>MyFirstGroup</div><div><div>●</div> Successfully completed</div></div> <div><div>Deployments</div><div>Subscriptions</div></div> <div><div>Automatically detect Core endpoints (recommended)</div><div>Greengrass will detect and override connection information as it changes.</div></div> <div><div>Manually configure Core endpoints</div><div>Manually manage connection information. This can be accessed via your Core device's settings.</div></div>

- Check the box that says "Do you want to force the reset?"
- Click "Reset deployment"
- Red deploy like before

- The "**Deployments**" tab will contain a log of successes /failures, also indicated by the **colored dot** in the heading
- Choose "**Automatic detection**" in the next window

26

See the MQTT messages:

- In the side bar of the AWS IoT Core console, choose "Test"
- Choose "Subscribe to topic" in the left-hand column and set the settings to look like the lower image
- Now click "Subscribe to topic" to the right of the "Subscription Topic" field



Monitor

Onboard

Manage

Greengrass

Secure

Defend

Act

Test

Subscriptions

[Subscribe to a topic](#)[Publish to a topic](#)

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe and receive these messages.

Subscription topic

Max message capture

Quality of Service



0 - This client will not acknowledge to the Device Gateway that message



1 - This client will acknowledge to the Device Gateway that message

MQTT payload display



Auto-format JSON payloads (improves readability)



Display payloads as strings (more accurate)



Display raw payloads (in hexadecimal)

27	There should now be a "Hello world!" MQTT message appearing in the AWS IoT test console every five or so seconds!	<div><div>Subscriptions</div><div><div>Subscribe to a topic</div><div>Publish to a topic</div></div><div><div>hello/world</div><div>x</div></div></div> <div><div>hello/world</div><div>Publish</div><div>Specify a topic and a message to publish with</div><div>hello/world</div><div><pre>1 { 2 "message": "Hello from AWS IoT core" 3 }</pre></div><div><div>hello/world</div><div>Feb 28, 2018 5:04:23 PM -</div><div>Hello world! Sent from Greengrass Core 4.9.30-v7+-armv7l-with-debian-8.0</div></div><div><div>hello/world</div><div>Feb 28, 2018 5:04:18 PM -</div><div>Hello world! Sent from Greengrass Core 4.9.30-v7+-armv7l-with-debian-8.0</div></div></div>
----	-------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Related articles