KES IST LOGIC.comp

KES IST LSGIC.simple





株式会社金沢エンジニアリングシステムズ

©Kanazawa Engineering Systems Inc.



アジェンダ



Greengrass と Lambda

AWS IoT Greengrass で Lambda を動かす







Lambda を設定する前の環境準備



・以下の機材があることをご確認ください。ハンズオン 接続編 を一通り実施しておくことをお勧めします。





Lambda を設定する前の環境準備

AWS IoT Greengrass でメッセージングができていること

PLC <-> KES IoT Logic <-> GG <-> AWS IoT と メッセージング ができていることを確認してください。





Greengrass と Lambda



Greengrass & Lambda

Greengrassグループ における Lambda の説明

Lambda は Greengrass Core にデプロイされ、メッセージング を利用し、Device、Cloudと通信やデータを処理します。

Greengrassグループ





Lambda と KES IoT Logic





Greengrass で Lambda を動かす



Lambda の 作成

















ES		AWS IoT	г	×	\lambda HelloWorld - Lambda	×	+							_		×
	←	\rightarrow C	🖰 🗎 us	-west-2.console.	aws.amazon.com/lambda/h	ome?reg	ion=us	-west-2#/fu	Inctions/Hello	Norld?tab=c	configura	tion		☆	Θ	:
		aw	IS	サービス ∽	リソースグループ 🗸	*				Ą		 オレ□ 	ゴン ・ :	サポート	• •	
	=	L	.ambda >	<u>関数</u> 〉 Hell	oWorld		ARN	I- 🗇 arn:a	aws:lambda:us	s-west-2:		💷 :functio	on:HelloW	orld	Î	١
		ł	Hell	スロット!	リング 限定条件 ▼		アクショ	ヨン ▼	テストイベン	ントの選択	•	ን አՒ	保存	2	ł	
		-			関数	がコー します	-ルさ -	ミれるト	リガーを							
			▼ Desi	gner												
			クリック	7		Hello Layers	World	ł	(0)							
			+ ト	リガーを追加								十 送信先	を追加]		
		ノフィ・	ードバック	😧 日本語	© 2008 -	2020, Am	azon We	b Services, Ir	nc. またはその関	連会社。無断	転用禁止。	プライノ	パシーポリシ	/—₹	▼ 利用規約	<u>1</u>

L









ES	🤘 AV	NS IoT			×	<mark>∧</mark> HelloWo	rld - Lambda	;	<	+						-		×
	$\leftarrow \rightarrow$	> C		us-west-2	.console.a	aws.amazor	n.com/lambo	la/home?re	gior	n=us-west-2#/	functions/He	elloWorld?tal	o=configu	iration		\$	Θ	:
		aw	'S -7	サービ	z .	עע-ג	グループ、	~ 🍾				Ĺ	7 🗕	■ - オ	レゴン・	サポー	ト ▾	
	=	L	ambda	〉 関数	> Hello	oWorld				ARN - 🗇 an	n:aws:lambd	a:us-west-2:	(::::::::::::::::::::::::::::::::::::::	ction:He	lloWorld	Â	١
		H	Hell.	🔼	ואעם	リング	限定条件	•	アク	クション 🔻) , , , , , , , , , , , , , , , , , ,	イベントの選	R ▼	テス ト		保存		
			⊘ ト す。	リガー Tes ,	stThingR	ule が関数 I	HelloWorld	に正常に追	加さ	されました。関	数は現在、	トリガーから	のイベン	トを受信し	ていま	×		
		-	設定	アク	セス権限	E=	タリング	これ ⁻ され3	でトまし	トリガーた した	が設定							
			▼ De	signer														
								🔊 Hell	oW	′orld								
							Ŵ	🕞 Laye	rs		(0)							
			¢	AWS I	σT			×]			_		十送信	言先を追け	ba		
			+	トリガーな	を追加												-	2
		フィー	-ドバッ?	ל 🕲 ו	日本語		© 20	08 - 2020, Ai	mazo	on Web Services	, Inc. またはそ	の関連会社。無	斷転用禁止	t. プラ	イバシーフ	ポリシー	利用規	約



















← → C 🔒 github.con	n/aws/aws-greengrass-core-sdk-python/	© ⇒
Code Unssues U	T Puil requests T P Actions Projects U C Secur	ity u 🗠 insignts
	Join GitHub today GitHub is home to over 50 million developers workir and review code, manage projects, and build sof Sign up	ng together to host () tware together.
SDK to use with function	s running on Greengrass Core using Bython _ https://aws.git	thub io /aws-groepgrass-
-0-8 commits	양 1 branch ⓒ 0 packages ⓒ 4 relea	ses At 4 contributors (1)クリッ
- O- 8 commits Branch: master - New pu	양 1 branch ⓒ 0 packages ⓒ 4 relea	Ises At 4 contributors ①クリッ Find file Clone or downlo
- O- 8 commits Branch: master - New pu	P 1 branch	Ises At 4 contributors ①クリッ Find file Clone or downlo Clone with HTTPS ③
-O- 8 commits Branch: master New pu It liuszeng Update version docs	¹ 1 branch	ses At 4 contributors ①クリッ Find file Clone or downlo Clone with HTTPS ⑦ Use Git or checkout with SVN using the web URL.
-O- 8 commits Branch: master - New pu It liuszeng Update version docs examples	P 1 branch O packages 4 relea Il request Il request compatibility description. Release of Version 1.5.0 Fix message formatting for HelloWorldCounter example.	ses At 4 contributors ①クリッ Find file Clone or downlo Clone with HTTPS ⑦ Use Git or checkout with SVN using the web URL. https://github.com/aws/aws-greengrass-co
-O- 8 commits Branch: master - New pu It liuszeng Update version docs examples greengrasssdk		As 4 contributors ①クリッ Find file Clone or downlo Clone with HTTPS ⑦ Use Git or checkout with SVN using the web URL. https://github.com/aws/aws-greengrass-co
-O- 8 commits Branch: master New pu It liuszeng Update version docs examples greengrasssdk CHANGELOG.rst		ses A 4 contributors ①クリッ Find file Clone or downlo Clone with HTTPS ⑦ Use Git or checkout with SVN using the web URL. https://github.com/aws/aws-greengrass-co じ Open in Desktop Download ZIP 7 monther





 \times

_

~ ?

	名前	更新日時	種類	サイズ
クイック アクセス	docs	2020/02/20 7:44	ファイル フォルダー	
PC	examples	2020/02/20 7:44	ファイル フォルダー	
j 3D オブジェクト	📙 greengrasssdk	2020/02/20 7:44	ファイル フォルダー	
ダウンロード	CHANGELOG.rst	2020/02/20 7:44	RST ファイル	1 KB
aws-greengrass-core-sdk-python-master		2020/02/20 7:44	ファイル	11 KB
aws-greengrass-core-sdk-python-master	MANIFEST.in	2020/02/20 7:44	IN ファイル	1 KB
docs	README.rst	2020/02/20 7:44	RST ファイル	4 KB
examples	requirements.txt	20/02/20 7:44	IXI ノアイル CEC ファイル	1 KB
greengrasssdk	setup.crg	9/20 7:44	Python File	2 KB
 ■ デスクトップ ■ ドキュメント ■ ピクチャ ■ ビデオ ■ ミュージック ■ ローカル ディスク (C:) 	gre	engrasssuk かめ安な	、ノイノノリオ	井になりま




Ln: 9 Col: 0















KES	λ Lamb	da	× +				-		
	\leftrightarrow \rightarrow	C 🔒 us-west-2.co	nsole.aws.amazon.com,	/lambda/home?region=us-	west-2#/functions/HelloWorld/e	edit/basic-settings?tab=co	onfi 🛧	₩ Θ =	
	a	WS サービス	✓ リソースグル-	-プ ~ 1 ★			ĭン ・ サポー	► *	
		Lambda 〉 関数 〉 基本設定を 基本設定 Info 説明 - オプション	、 HelloWorld 〉 基本 編集	設定を編集 プログラミンク pythonファイル に合わせ編集し 保存して、関数	、した レ名のエントリー関数 してください なのポータルに戻りま	文名 t す			
		Python 3.7 ハンドラ Info lambda_function メモリ (MB) 作成する関数には、設 タイムアウト 0 分 3	lambda_handler 定したメモリに比例する c	CPU が割り当てられます。	■ 128 MB		\$		46
	•ر 🔎	ィードバック 🔇 日オ	雨	© 2008 - 2020, Amazon Web) Services, Inc. またはその関連会社。	無断転用禁止。 プライノ	(シーポリシー	利用規約	40



Lambda で Publish を受信する



ES	e A	WS IoT				×	λ	Hello	World	d - Lam	nbda		×	+													-		×	
<	<u>,</u> .	→ C	A	ıs-we	st-2.c	onsole	e.aw	/s.ama	izon.c	com/la	ambda/	home	?regio	on=us	-west-	-2#/fui	nctio	ons/He	lloWo	orld?t	ab=c	onfig	jurati	ion			☆	6) :	
																							-							
			\$	LATI	EST	から	新(しい	バ—	ジョ	ンを孕	発行し	します	す。												×	rtd			
			e 新 訳 ン	しい) のコ・	バージ - ドを	^ジ ョン 編集	を発 する	行する ことは	ると、 までき	\$LAT きません	TEST バ ん。ク!	-ジョ リック	ョンの フしてネ	コート 潅定し	[×] と設; ,ます。	定のス	ナッ	プショ	3 y h	が保	存さ	ntr	す。新	fUl	いに一き	ジョ				
				ージ	ヨンの	〕説明	- オ	プショ	ョン 	1-	<u></u> 千音の)前[明を	γ	+=-	ਰ														
												hel	lloW	orld							+	ヤン1	セル		発行	Ŧ				
																								ク	リッ	ク				
																														2

Ĺ













	ない サービス ▼	リソースグループ 🗸	*		♪ <u> </u>	オレゴン ・	サポート 👻	
								\cap
Lan	nbda の選択							S
	○ すべての Lambda 関数	レログを絵弦						
			i ythôn 2.	,				
	0		Python 2.	7				
	0	i dina	Python 2.	7				
	0		Python 2.	7				
	● HelloWorld 7	- ェックを付けま [、]	Python 3.	7				
	0		Python 2.	7			_	
	0		Node.js 6.	10			*	
					ク	リック		
±+	アンセル				戻る	、次へ		



← REENGRASS グルー	-プ	
TestGG		
 正常に完了しま 	ました	アク ション •
デプロイ 🎝	Lambda	Lambda の追加
サブスクリプション	·	
コア	HelloWorld	
デバイス	LAMBDA 問数	エイリアスの使用: TESILAMBDAT
Lambda		
リソース	Greengrassグループ	にLambda
コネクタ	が追加されました	
タグ		
設定		

(







	est-2.console.aws.amazon.com/iot	r/nome?region=us-west-2#/greengrass/gro	ups/t	······································
	ーヒス 、 リソースクルーフ	· *	געא א משפע אָע	ン・サポート・
○ バイナリ				
Ŭ				Û
環境変数				\bigcirc
環境変数を、関数 関数コードを変更	↓コードからアクセス可能なキーと ■することなく構成設定を保存する	2値のペアとして定義できます。これらは、 5のに便利です。		
‡—		値		
例• 色		例· 吉		\otimes
				Ŭ
Lambda 関数の グループに Lamb た個別の Lambda 別バージョンの	の別バージョンの追加 oda 関数の過去バージョンを個別に a 関数として扱われます。 <mark>の追加</mark>	こ追加できます。各バージョンはグループ固	回有の設定のセットを持ち、グループに クリッ	:よって独立し / ク











Lambda と KES IoT Logic



これで Lamdbda の作成と Thing のデータ受信の設定ができました。









KEG	🖺 デスクトップ¥HelloWorld.log - sakura 2.2.0.1	- 🗆	×
L RES	ファイル(E) 編集(E) 変換(C) 検索(S) ツール(I) 設定(O) ウィンドウ(W) ヘルプ(H)		
	[· · · · · · · · · · · · · · · · · · ·		
	HelloWorld.log lambda_function.py 9行目に書いた		×
	<pre>22 [2020-06-16T18:38:44.738+09:00][INF0]-ipc_cl] 8d79df] to <u>http://localhost:8000/2016-11-01/f</u> 23 [2020-06-16T18:38:44.746+09:00][INF0]-ipc_cl] d79df]↓ 24 [2020-06-16T18:38:44.746+09:00][INF0]-ipc_cl] にて出力しているログメッセージがあることを確認くだ tion:HelloWorld:1] from <u>http://localhost:8000</u> 25 [2020-06-16T18:38:45.747+09:00][INF0]-ipc_client.py:202,Got w with invocation id [78edb59d-1d27-43b9-4d55-</pre>	event))) さい -74f4553e8	22
	26 [2020-06-16T18:38:45.749+09:00][INF0]-lambda_function.py:9,-> publish json: {"tag": "tag", "seq": 0, "msg": {"entr "2020-06-16T18:38:45+09:00", "mac_address": "00:11:0C:1B:85:B9", "device_type": "test", "event_type": "3", "Conne "D100": 1000}}↓	ryDatetime ectError":	": 0
	27 [2020-06-16118:38:45.75+09:00][INF0]-ipc_client.py:223,Posting work result for invocation id [78edb59d-1d27-43b9-4 e8221] to <u>http://localhost:8000/2016-11-01/functions/arn:aws:lambda:us-west-2:111017:function:HelloWorld:1/v</u> 28 [2020-06-16T18:38:45.758+09:00][INF0]-ipc_client.py:232,Posted work result for invocation id [78edb59d-1d27-43b9-4 e8221].	id55-74†45: <u>vork</u> ↓ id55-74f45!	53<
	29 [2020-06-16T18:38:45.759+09:00][INF0]-ipc_client.py:192,Getting work for function [arn:aws:lambda:us-west-2: tion:HelloWorld:1] from <u>http://localhost:8000/2016-11-01/functions/arn:aws:lambda:us-west-2:</u> [////////////////////////////////////	HelloWorl	ncK <u>d:</u> K
	30 2020-06-16T18:38:55.761+09:00][INF0]-ipc_client.py:202,Got work item with invocation id [9102b163-17d0-4e28-4afb-	-918b03a76	50 <mark><</mark>
	31 [2020-06-16T18:38:55.763+09:00][INF0]-lambda_function.py:9,-> publish json: {"tag": "tag", "seq": 0, "msg": {"entr "2020-06-16T18:38:55+09:00", "mac_address": "00:11:0C:1B:85:B9", "device_type": "test", "event_type": "3", "Conne , "D100": 1000}}↓	ryDatetime ectError":	": 0
	32 [2020-06-16118:38:55.763+09:00][INF0]-ipc_client.py:223,Post result for invocation id [9102b163-17d0-4e28- a76500] to <u>http://localhost:8000/2016-11-01/1</u> 33 [2020-06-16T18:38:55.771+09:00][INF0]-ipc_client.py:223,Post result for invocation id [9102b163-17d0-4e28- 76500]↓ 34 [2020-06-16T18:38:55.772+09:00][INF0]-ipc_client.py:223,Post result for invocation id [9102b163-17d0-4e28- がお出力されます 34 [2020-06-16T18:38:55.772+09:00][INF0]-ipc_client.py:223,Post result for invocation id [9102b163-17d0-4e28- がお出力されます 1/work↓ EDE	4atb-918b	J3
			> 7
	1行 1桁 LF 5B SJIS	REC 3	重入



Lambda と KES IoT Logic

KES IoT Logic(IoT Thing) と Lambda 間の通信

これで Lamdbda で Thing のデータが受信できていることが確認できました。




Lambda から Publish を送信する



Lambda と KES IoT Logic

KES IoT Logic(IoT Thing) と Lambda 間の通信

次に Lamdbda から IoT Cloud ヘデータを送信します。





	1.77 (C) > ggtest > Helloworld	V O P Hellov	voriav使察	
🔪 🛃 ከፈጣስ ካስት ገ	名前	更新日時	種類	サイズ
	greengrasssdk	2020/06/16 15:23	ファイル フォルダー	
✓ □ PC	HelloWorld.zip	2020/06/16 15:50	圧縮 (zip 形式) フォ 圧縮 (zip 形式) フォ	30 KB
	lambda function.py	2020/06/29 13:44)土相 (Zip ルシシレ) ノオ… Python File	1 KB
> = デスクトップ			.,	
> 🗐 ドキュメント				
> 📰 ピクチャ	書き換	えたソースファイル	k greengrass	sdk を合
> 📕 ビデオ	自じ広わせて	zin圧縮します		очкец
> 🁌 ミュージック				
> 🏪 ローカル ディスク (C:)		マイル名は 任意 でる	†	
> 💣 ネットワーク				







S	HelloWorld:te	stLambda2 - Lambo 🗙	+					- [X
÷	→ C	us-west-2.console	.aws.amazon.com/lamb	oda/home?region=us-v	vest-2#/functions/HelloWo	rld/aliases/testLam	bda2?tab=con 🗴	*	e :
	aws	サービス 🗸	リソースグループ	~ *		↓ ,====	 オレゴン ▼ 	サポート	*
≡	Ø I1!	リアス testLambda2 カ	「正常に作成され、バー	ジョン 2 を指していま	:す。			×	Â
	Lamb	da \ 朋物 \ Hol	IoWorld \testLamb	oda2					
	Lamb			ARN - 🗇 arn:aws:	lambda:us-west-2:431099	347647:function:H	lelloWorld:testLamb	oda2	
	He	lloWorld:te	stLambda2						
				これで					
		イリアス: testLambd		Lambdaの影	没定は完了です	保住			
	設	定 アクセス権限	モニタリング	AWS IoT 側	での作業に戻りま	ミす			
	•	デザイナー							
									Ŀ
				HelloWorld:	testLambda2				
				😂 Layers	(0)				
		➡ トリガーを追加		*		Γ	→ 送信先を追加		
		〒 1.572 夜道加					〒 必福花を連加		-
	フィードノ	「ック 🔇 日本語	© 2	2008 - 2020, Amazon Web	Services, Inc. またはその関連会	会社。無断転用禁止。	プライバシーポリミ	シー 利	用規約

Ĺ



aws サービス v	リソースグループ 🗸 🏠	9===== ▼ オレゴン ▼ サポート ▼
GREENGRASS グループへの LAMBDA の	D追加	
既存の Lambda の使	明	
Lambda の選択		
Q すべての Lambda 関数とA		
	Python 2.7	
	Python 2.7	
	Python 2.7	
● HelloWorld ← f I	ニックを付けます Python 3.7	
	Python 3.7	
0		

(









	aws ±	ナービス ~ リ!	ソースグループ ∨	*		בעד • 💳 🗘	「ン ▼ サポート ▼
	環境変数 環境変数を、関数 関数コードを変感	数コードからアクセ 更することなく構成	ス可能なキーと値の/ 設定を保存するのに(ペアとして定 更利です。	義できます。これらは、		Д Ф
	キ ー 例: 色				値 例: 青		\otimes
-	Lambda 関数	マの別バージョン bda 関数の過去バー	ノの追加 -ジョンを個別に追加	できます。名	バージョンはグループ固有の設定	のセットを持ち、グループ(こよって独立し
	シルーフに Lami		します。				
	クルークに Lamb た個別の Lambd 別バージョン	la 関数として扱われ יの追加				クリッ	<u>ク</u>

Ĺ









💾 Log File Manager × +

← → C ③ 保護されていない通信 | 192.168.252.553/filemanager.php?p=ggc%2Fvar%2Flog%2Fuser%2Fus-west-2%2F

 \times

* 0 (:) ☆

Name	Size	Modified	Perms	Owner	
<u>€.</u> ダウンロード					
HelloWorld.log	1.18 KiB	2020/06/30 14:30	0600	root:root	ee 🛓
HelloWorld.log-1593494556011873413	25.39 KiB	2020/06/30 14:22	0600	root:root	ee 🛓
HelloWorld.log-1593494716011869567	25.48 KiB	2020/06/30 14:25	0600	root:root	ee 🖕
HelloWorld.log-1593494876012019962	25.51 KiB	2020/06/30 14:27	0600	root:root	ee 🖕
HelloWorld.log-1593495048512342679	25.41 KiB	2020/06/30 14:30	0600	root:root	ee 🖕
Full size: 102.97 KiB, files: 5, folders: 0					

filemanager を開き、lambdaが データ受信できていることを確認

91

2

KEG	🖺 C:¥Users¥genda¥Downloads¥HelloWorld.log - sakura 2.2.0.1 — 🗆 🗙
	<mark>ファイル(E)</mark> 編集(E) 変換(<u>C</u>) 検索(<u>S</u>) ツール(<u>T</u>) 設定(<u>O</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)
	HelloWorld.log 🗸 🖌
	64 [2020-06-30T14:32:02.019+09:00][INF0]-lambda_function.py:12,-> publish json: {"tag": "tag", "seq": 0, "msg": {"entryDatetime" : "2020-06-30T14:32:01+09:00", "mac_address": "00:11:0C:1B:85:B9", "device_type": "test", "event_type": "3", "ConnectError": 0, "D100": 0}}↓
	65 [2020-00-30T14:32:02:02+09:00][INF0]-ipc_chient.py:107,Postice function [arn:aws:lambda:function:GGRouter] to http://
	66 [2020-06-30T14:32:02.035+09:00][INF0]-ipc_ lambda_function.py 9行目に書いた
	$ \begin{bmatrix} 2020 - 06 - 30T14: 32: 02.035 + 09: 00 \end{bmatrix} [INF0] - ipc \\ 2bc74d \end{bmatrix} to \\ \frac{http://localhost: 8000/2016 - 11 - 0}{2020 - 06 - 30T14: 32: 02.039 + 09: 00] [INF0] - ipc } \\ logging.info("-> publish json: {}".format(json.dumps(event))) \\ \frac{10^{k}}{12^{k}} \\ \frac{12^{k}}{12^{k}} \end{bmatrix} $
	69 [2020-06-30T14:32:02.04+09:00][INF0]-ipc_c にて出力しているログメッセージがあることを確認ください ion:HelloWorld:2] from <u>http://localhost:80</u> Lambda が Thing のPublish を受信していることになります 2 ^K
	70 [2020-06-30T14:32:12.031+09:00][INF0]-ipc
	71 [2020-06-30T14:32:12.032+09:00][INF0]-lambda_function.py:12,-> publish json: {"tag": "tag", "seq": 0, "msg": {"entryDatetime" : "2020-06-30T14:32:11+09:00", "mac_address": "00:11:0C:1B:85:B9", "device_type": "test", "event_type": "3", "ConnectError": 0, "D100": 0}}↓
	72 [2020-06-30114:32:12.034+09:00][INF0]-ipc_client.py:167,Posting work for function Larn:aws:lambda:::function:GGKouter] to http://www.info.com/com/com/com/com/com/com/com/com/com/
	73 [2020-06-30T14:32:12.044+09:00][INF0]-ipc_client.py:177,Work posted with invocation id [07d4b966-9091-4d15-75e8-5bded901c723]
	74 [2020-06-30T14:32:12.046+09:00][INF0]-ipc_client.py:223,Posting work result for invocation id [f9a45787-447e-4609-7f6d-b56f2dk
	75 [2020-06-30T14:32:12.054+09:00][INF0]-ipc_client.py:232,Posted work result for invocation id [f9a45787-447e-4609-7f6d-b56f2d2k
	09c99]↓ 76 [2020_06_20T14:22:12_055±00:00][INEO]_inc_client_nu:102_Catting_work_for_function_Form:owe:lembdo:up-wort-2:
	tion:HelloWorld:2] from <a 2016-11-01="" 8000="" 800<="" arn:aws:lambda:us-west-2:"http:="" functions="" href="http://localhost:8000/2016-11-01/functions/arn:aws:lambda:us-west-2:" http:="" localhost:8000="" th="">
	コンジェンジョン 75行 20桁 LF 32 SJIS REC 挿入





お疲れさまでした! これでLambda編は完了です!



©Kanazawa Engineering Systems Inc.