	Internacion ONICAS INTEGRAL	Nota de Aplicación: CAN-107 Título: AWS IoT Analytics para ingenieros y desarrolladores de sistemas dedicados Autor: Sergio R. Caprile, Senior R&D Engineer
Revisiones	Fecha	Comentarios
0	18/09/20	

En el <u>CTC-104</u> analizamos los Amazon Web Services (AWS) y desarrollamos la utilización de AWS IoT Core, el servicio de conectividad, utilizando MQTT. En el <u>CTC-105</u> vimos como conectarnos a dicha plataforma IoT con un ESP32 y Mongoose-OS, enviando datos de telemetría; mientras que en el <u>CTC-106</u> lo hicimos con módulos SIMCOM. En esta oportunidad nos adentraremos un poco en el otro lado, en el almacenamiento y presentación posterior de los datos, lo cual haremos desde la óptica de un ingeniero o desarrollador de sistemas dedicados; es decir, no con la intención de servir páginas web a proyectos de clientes sino como una herramienta más para poder observar el funcionamiento de nuestros equipos o proveer de información a quienes deben tomar decisiones de negocios en base a estos datos.

Breve descripción de uso de AWS loT Analytics

En principio, usamos este servicio por simpleza y conveniencia; para una aplicación IoT real puede no ser la mejor opción y de esto nuestros colegas de sistemas saben mucho más de lo que podamos analizar aquí.

Se trata de un componente más de la serie de servicios de AWS IoT¹. El mismo nos permite almacenar nuestros datos en una base a tal efecto, para luego extraerlos posteriormente con el fin de analizarlos, disponiendo de diversas herramientas de procesamiento y selección en cada una de sus etapas constitutivas. El proceso de operación es el indicado en la figura siguiente:



- Definimos un canal (*channel*) mediante el cual ingresan los datos. Esto lo haremos en AWS IoT Core empleando el motor de reglas para direccionar datos ingresados por MQTT a AWS IoT Analytics.
- Procesamos los datos en una *pipeline*, si necesitamos realizar conversiones de unidades o algo similar podemos realizarlo aquí
- Almacenamos los datos en un *data store*
- Analizamos los datos mediante una consulta en base de datos, obteniendo de todos los disponibles aquéllos que nos interesan y generando un *data set*
- Construimos a partir de los *data sets*, los gráficos que nos permitirán observar lo que deseamos y así poder sacar conclusiones. Esto lo haremos en Quicksight

¹ Es fundamental en este punto haber leído el <u>CTC-104</u> para conocer de qué estamos hablando.

En AWS loT Core

Debemos crear una regla que nos permita tomar determinados mensajes y enviarlos a IoT Analytics. En el <u>CTC-105</u> generamos datos de telemetría enviando periódicamente la información de memoria libre en el ESP32; lamentablemente no incluimos un timestamp, pero podemos agregarlo aquí.¹

Comenzamos en la consola seleccionando $Act \rightarrow Rules$ y allí *Create a rule*. Ingresamos el nombre y una descripción.

aws	Services - R	esource Groups 🗸 🔭 🗘 RnD 🔻 Ohio
AWS IOT	×	AWS IoT > Rules > Create a rule
Monitor Activity • Onboard		Create a rule
Manage		
Greengrass		Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).
Secure		Name
Defend		can
▼ Act		
Rules		Description
Destinations		send to IoT Analytics
Test		
Software		
Settings		
Learn		

Definimos luego la consulta que vamos a hacer en base de datos. Como indicamos, debemos agregar un campo de timestamp, para lo cual nos valemos de las funciones provistas; la sintaxis es similar a SQL, nuestro query es: SELECT *,timestamp() as timestamp FROM '+/telemetry'

aws	Services - Resour	ce Groups 👻 🕏	Å RnD → Of
AWS IoT	×	Using SQL version	
Monitor		2010-03-23	
Activity		Rule query statement	
Onboard		SELECT <attribute> FROM <topic filter=""> WHERE <condition>. For example: SELECT temperature FROM 'iot/topi learn more, see AWS IoT SQL Reference.</condition></topic></attribute>	ic' WHERE temperature > 50. To
Manage		1 SELECT *,timestamp() as timestamp FROM '+/telemetry'	
Greengrass			
Secure			
Defend			
▼ Act			
Rules			
Destinations			
Test		Set one or more actions	
Software		Select one or more actions to happen when the above rule is matched by an inbound message. Actions define add	ditional activities that occur when
Settings		messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (*.required)	
Learn		Add extin	
Documentation 🛽		Add action	

Creamos luego una acción, lo que nos permitirá hacer algo cuando se dispara la regla. La acción es "enviar un mensaje a IoT Analytics"

¹ En principio es deseable acompañar los datos de un timestamp, dado que corresponde al momento de generación de la información. Sin embargo, esto requiere la presencia de un RTC o equivalente (SNTP y mecanismo asociado, por ejemplo) en el dispositivo, lo cual no siempre es posible por costo o limitaciones. Un paso siguiente es valerse de un recurso de la nube para introducirlo, siempre que la demora asociada sea mucho menor que el tiempo entre mensajes podemos hacerlo sin mayores inconvenientes. De todos modos, el análisis debe hacerse en función de lo que se desea de la aplicación, siendo éstas consideraciones relacionadas a una nota de aplicación genérica.

aws	Services - Resourc	ce Groups 🗸 🐧	*	¢	RnD 👻	Ohio
AWS IoT	×	0	COUDWATCH ALARMS			
Monitor Activity		○ 🌷	Send message data to CloudWatch logs			
OnboardManage		•	Send a message to the Amazon Elasticsearch Service			
Greengrass Secure Defend			Send a message to a Salesforce IoT Input Stream SALESFORCE IOT			
▼ Act Rules Destinations		• 📀	Send a message to IoT Analytics			
Test		· •	Send a message to an IoT Events Input IOT EVENTS			
Software Settings Learn		0 🔯	Start a Step Functions state machine execution STEP FUNCTIONS			
Documentation		0	Send a message to a downstream HTTPS endpoint			
		Cancel		Configu	re action	

Seleccionando la forma rápida (Quick create) se crea todo lo necesario.

aws Services -	Resource Groups 🗸 🔭	🗘 RnD 🕶	Ohio
AWS IOT	X AWS IOT > Rules >		
Monitor Activity > Onboard > Manage	Configure action		
Greengrass Secure Defend	Ouick create IoT Analytics resources		
Act Rules Destinations Test Software	This option will create channel, pipeline, data store and SQL data set in IoT Analytics. *Resource Prefix ③ can Quick Create		
Settings Learn Documentation 🛃	 Manually select IoT Analytics Channel and role This option will allow you to select an existing IoT Analytics channel and role to send your data. 		
	Cancel		

Ponemos simplemente el nombre deseado y se crean el *channel, pipeline, data store* y *data set*; y un *role* (permisos IAM necesarios para que IoT Core pueda acceder por nosotros a IoT Analytics) asociado.

AWS IoT	×	Quick of a constraint of the second secon	create IoT Analytics res	ources					
Monitor		This opti	ion will create channel, pipeli	ne, data store and SQL data set in	IoT Analytics.				
Activity		Quick Cr	reate Finished						
Onboard		can_c	channel		Created				
Manago					Created -				
Contraction		can_d	datastore		Created				
Greengrass		con	nineline						
Secure		can_p	pipetite		Created				
Defend		can_d	dataset		Created				
Rules									
Destinations		can_r	rote		Created 🍑				
Test		Your me	essages will be sent to this cha	nnel can_channel					
		 Manual 	Illy select IoT Analytics	Channel and role					
Software		This opti	ion will allow you to select ar	existing IoT Analytics channel an	nd role to send yo	our data.			
Settings									
Learn									
Documentation 🖪									
aws _{Serv}	rices y Resou	Cancel	\$				ſ	Add acti	ion
aws serv	rices - Resou X	Cancel urce Groups ~	*				Ĺ	Add acti	ion
aws Serv Aws lot	rices + Resou	Cancel	*				Ĺ	Add acti	ion
aws Serv Aws lot Monitor	rices Y Resou X	Cancel	* itement ute> FROM <topic filter=""> WH AWS IoT SQL Reference.</topic>	IERE <condition>. For example: S</condition>	ELECT temperat	ure FROM 'iot/topi	f. c' WHERE tempera	Add acti) •) •
AWS IOT Monitor Activity	rices - Resou X	Cancel	* itement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. `*.timestamp() as timestal</topic>	IERE <condition>. For example: S mp FROM '+/telemetry'</condition>	SELECT temperat	ure FROM 'lot/topi	L c' WHERE tempera	Add acti) •). To
AWS IoT Activity Onbeard	ices → Resou X	Cancel	* itement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. *.timestamp() as timesta</topic>	IERE <condition>. For example: S np FROM '+/telemetry'</condition>	ELECT temperat	ire FROM 'lot/topi	L c' WHERE tempera	Add acti) •). To
AWS IoT Activity Onboard Manage	ices - Resou X	Cancel	* itement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. Tutimestamp() as timestat</topic>	IERE <condition>. For example: S mp FROM '+/telemetry'</condition>	ELECT temperat	ıre FROM 'iot/topi	L c' WHERE tempera	Add acti) ¥
AWS IoT Activity Onboard Manage Greengrass	ices - Resou X	Cancel	* itement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. **.timestamp() as timestat</topic>	IERE <condition>. For example: S mp FROM '+/telemetry'</condition>	ELECT temperat	ıre FROM 'İot/topi	L c' WHERE tempera	Add acti	ion) •
AWS IoT Autivity Onboard Manage Greengrass Secure	rices - Resou X	Cancel	* ttement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. * timestamp() as timestal</topic>	ERE <condition>. For example: S mp FROM '+/telemetry'</condition>	ELECT temperat	ire FROM 'iot/topi	L c' WHERE tempera	Add acti	ion) •
AWS IoT Monitor Activity Onboard Greengrass Secure Defend	rices ~ Resou	Cancel	* ttement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. * timestamp() as timestar</topic>	ERE <condition>. For example: S mp FROM '+/telemetry'</condition>	ELECT temperat	ıre FROM 'lot/topi	L	Add acti	ion) •
AWS IoT Monitor Activity • Onboard • Greengrass • Secure • Defend • Act	rices - Resou	Cancel	* tement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. * timestamp() as timestal</topic>	ERE <condition>. For example: S mp FROM '+/telemetry'</condition>	ELECT temperat	ıre FROM 'iot/topi	L C'WHERE tempera	Add acti	ion) ¥
AWS IoT Monitor Activity Onboard Manage Greengrass Secure Defend Act Rules	ices - Resou X	Cancel	* itement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. Tutimestamp() as timestal</topic>	IERE <condition>. For example: S np FROM '+/telemetry'</condition>	ELECT temperat	ire FROM 'iot/topi	L C WHERE tempera	Add acti	ion) •
AWS IoT Monitor Activity Onboard Manage Greengrass Secure Defend Act Rules Destinations	ices Y Resou X	Cancel	* terment ute> FROM <topic filter=""> WF AWS IoT SQL Reference. *.timestamp() as timestal more actions</topic>	IERE <condition>. For example: S mp FROM '+/telemetry'</condition>	iELECT temperati	ire FROM 'lot/topi	c' WHERE tempera	Add acti	ion • •
AWS IoT Monitor Activity Onboard Manage Greengrass Secure Defend Act Rules Destinations Test	ices - Resou X	Cancel	terment ute>FROM <topic filter=""> WH AWS IoT SQL Reference. Timestamp() as timestat nore actions to happen when t b, like storing them in a datab</topic>	IERE <condition>. For example: S np FROM '+/telemetry' he above rule is matched by an in ase, invoking cloud functions, or s</condition>	SELECT temperat	Ire FROM 'iot/topi	C WHERE tempera	Add acti	ion) ×). To
AWS IoT Monitor Activity Onboard Manage Greengrass Secure Defend Act Rules Destinations Test	ices - Resou	Cancel	terment ute> FROM <topic filter=""> WH AWS IoT SQL Reference. Thimestamp() as timestat nore actions tore actions to happen when t b, like storing them in a datab</topic>	IERE <condition>. For example: S mp FROM '+/telemetry' he above rule is matched by an in ase, invoking cloud functions, or s</condition>	ELECT temperation	Ire FROM 'iot/topi Actions define add	C WHERE tempera	Add act) •
AWS IoT Monitor Activity Onboard Manage Greengrass Secure Defend Act Rules Destinations Test Software Software	ices - Resou	Cancel Irce Groups Rule query stat SELECT <attribute learn="" m="" more,="" o<="" one="" or="" see="" select="" set="" td=""><td>terment ute> FROM <topic filter=""> WF AWS IoT SQL Reference. *.timestamp() as timestal nore actions to reactions to happen when t l, like storing them in a datab Send a message to IoT</topic></td><td>IERE <condition>. For example: S mp FROM '+/telemetry' he above rule is matched by an in ase, invoking cloud functions, or s Analytics</condition></td><td>ELECT temperation</td><td>ure FROM 'lot/topi Actions define add</td><td>c' WHERE tempera</td><td>Add acti A RnD ture > 50 at occur Edit</td><td>ov D. To whe</td></attribute>	terment ute> FROM <topic filter=""> WF AWS IoT SQL Reference. *.timestamp() as timestal nore actions to reactions to happen when t l, like storing them in a datab Send a message to IoT</topic>	IERE <condition>. For example: S mp FROM '+/telemetry' he above rule is matched by an in ase, invoking cloud functions, or s Analytics</condition>	ELECT temperation	ure FROM 'lot/topi Actions define add	c' WHERE tempera	Add acti A RnD ture > 50 at occur Edit	ov D. To whe
AWS IoT Monitor Activity • Onboard • Manage • Greengrass • Secure • Defend • Act Rules Destinations Test Software Settings Learn	ices ~ Resou	Cancel Irce Groups Rule query stat SELECT <attribulearn arrive<="" m="" messages="" more,="" one="" or="" see="" select="" set="" td=""><td>tement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. *.timestamp() as timestar nore actions tore actions to happen when t e, like storing them in a datab Send a message to IoT can_channel</topic></td><td>IERE <condition>. For example: S mp FROM '+/telemetry' he above rule is matched by an in ase, invoking cloud functions, or s Analytics</condition></td><td>ELECT temperation</td><td>ire FROM 'lot/topi Actions define add</td><td>t WHERE tempera</td><td>Add acti RnD ture > 50 tat occur Edit</td><td>ion) ×). To whe</td></attribulearn>	tement ute> FROM <topic filter=""> WH AWS IoT SQL Reference. *.timestamp() as timestar nore actions tore actions to happen when t e, like storing them in a datab Send a message to IoT can_channel</topic>	IERE <condition>. For example: S mp FROM '+/telemetry' he above rule is matched by an in ase, invoking cloud functions, or s Analytics</condition>	ELECT temperation	ire FROM 'lot/topi Actions define add	t WHERE tempera	Add acti RnD ture > 50 tat occur Edit	ion) ×). To whe
AWS IoT Monitor Activity • Onboard • Manage • Greengrass • Secure • Defend • Act Rules Destinations Test Software Settings Leam Decumentation [2]	ices ~ Resou	Cancel	 Itement Itement Itemestamp() as timestal *,timestamp() as timestal nore actions nore actions to happen when to live storing them in a datab Send a message to IoT con_channet 	IERE <condition>. For example: S mp FROM '+/telemetry' he above rule is matched by an in ase, invoking cloud functions, or s Analytics</condition>	SELECT temperati	ure FROM 'lot/topi Actions define add	C WHERE tempera itional activities the Remove	Add acti A RnD ture > 50 at occur Edit	ion))))))) whe

Con esto finalizamos

aws	Services 🗸 Re	source Groups 👻 🗙	¢	RnD 👻	Ohio 👻	Support 👻
AWS IOT	×	Success Successfully created rule.				
Monitor Activity		AWS IOT > Rules				
Onboard		Rules				Create
Manage						
Greengrass		Search rules				
Secure						
Defend		Name	Status			
▼ Act Rules		🗆 can				•••
Destinations						

A partir de este momento, los mensajes MQTT que satisfagan la consulta SQL (todos los mensajes publicados en el tópico que usamos para reportes de telemetría) recibirán un timestamp y serán almacenados en el *data store* de que disponemos en IoT Analytics.

En AWS IoT Analytics

Todo lo necesario ha sido creado ya, sin embargo, es conveniente modificar la configuración del *data store*, a fin de que no guarde los datos por siempre sino por un determinado tiempo.

El uso de este servicio tiene un costo, por procesamiento, por consulta en base de datos y por almacenamiento.¹ Dentro de los usos mencionados aquí, estamos dentro del *free tier* por el período de doce meses desde la apertura de nuestra cuenta AWS.

De modo similar, podemos configurar la *pipeline* para eliminar o procesar datos, acorde a nuestras necesidades, y configurar consultas.

Para poblar el *data set*, podemos configurar un disparo cada un determinado tiempo. En este caso lo haremos manualmente. La acción se realiza mediante otra consulta con sintaxis SQL, en este caso como al principio teníamos datos a los que no habíamos agregado el timestamp, generamos una consulta SELECT * FROM can_datastore WHERE timestamp > 0 para poblar el *data set*.

etails			
QL query			
select * from	<pre>can_datastore where</pre>	timestamp > 0	
elta window			
elta window has no	ot been set yet.		
esult preview			
total	free	timestamp	dt
282696	201196	1600192381668	2020-09-15 00:00:00.000
282696	201196	1600192391664	2020-09-15 00:00:00.000
282696	201196	1600192401670	2020-09-15 00:00:00.000
282696	201196	1600192621667	2020-09-15 00:00:00.000
282696	201196	1600192631671	2020-09-15 00:00:00.000

Aquí, "total" y "free" son los campos contenidos en los mensajes de telemetría que enviamos desde el ESP32 con Mongoose-OS en el <u>CTC-105</u>; "timestamp" lo agregamos en IoT Core en el motor de reglas, como hemos visto.²

En Quicksight

Quicksight es un servicio del paquete de servicios Analytics (no IoT Analytics).

Antes de poder hacer cualquier cosa, deberemos habilitar este servicio eligiendo una forma de uso. Podemos elegir las prestaciones en función del costo, en este caso hemos elegido la opción gratuita que permite que sólo un usuario con capacidad de crear gráficos disponga de 1GB de almacenamiento.

Aquí todo es muy gráfico, en principio para realizar un gráfico simple y acorde a nuestras necesidades sólo debemos configurar que importamos de un data set en AWS IoT Analytics

^{1 &}lt;u>https://aws.amazon.com/iot-analytics/pricing/</u>

² Corresponde por lo tanto al momento de recepción, no al de envío. En las pruebas realizadas observamos que no existe demora apreciable entre el momento en que se envía la información y el que se la observa en un cliente MQTT, dado que se trata completamente de un broker MQTT. Los mensajes han sido generados cada 10 segundos y podemos observar una diferencia levemente mayor a 10000 ms entre los respectivos timestamps. Otras plataformas, como GCP, realizan un procesamiento por lotes (en PubSub) y la operación es diferente.

VuickSight			ج 39904.
Data Sets	New AWS IoT Analytics	data source ×	SPICE capacity for this region: 14.5MB of 1GB
Athena	Data source name Can_dataset Select an AWS IoT Analytics data	a set to import:	Redshift Manual connect
MySQL	Cancel	Create data source	Aurora
La MariaDB	Presto	Spork Spark	Teradata Provided by Teradata
Snowflake	AWS IoT Analytics	GitHub	Twitter
Jira	กษพ ServiceNow	Adobe Analytics	
FROM EXISTING DATA SOURCES			

para luego proceder a generar el gráfico

	+ り C ¹ Add Undo Redo		☆ can_dataset analysis	Autosave ON 🗸	() Print	Ĵ Share	<u>ج</u> 39904
م00 <	Data set	1	Field wells X axis # timestamp Y axis # free Group/Color Size				*
Visualize	SPICE can_dataset 🗸 🗸	100%	Sheet 1 $ \smallsetminus $ +		Import complete:		×
Filter	Fields list	۹	Count of Records by Timestamp and Free	×	100% success 130 rows were imported to SPICE		
	💾 _dt		SHOWING TOP 50 IN TIMESTAMP AND TOP 1 IN FREE	×*	0 rows were skipped		
Story	# timestamp		300,000				
ŶŶŶ	# total						
Parameters			200,000				
10							
Actions			100,000				
Themes							
ര	Visual types	~	0 1,600,191,500,000 1,600,191,700,000 1,600,191,900,000 1,600,192_/				
Settings							
		~					
	~						
	🔛 🖟 🚱 🕭						

Como podemos apreciar, hemos logrado visualizar la evolución de una variable de telemetría en función del tiempo a lo largo de un tiempo determinado (el cual fijamos al momento de poblar el data set, en IoT Analytics).¹

De aquí en más, es posible compartir los gráficos con otras personas y realizar cosas más complejas, para lo cual remitimos a la documentación.²

¹ Si bien el gráfico no resulta muy interesante debido a que la variable no ha cambiado durante la medición (lo cual es bueno porque se trata de la cantidad de memoria libre en la *heap*), podemos apreciar el procedimiento y la utilidad del servicio.

² https://docs.aws.amazon.com/quicksight/latest/user/welcome.html