

# ACQUISIRE E ANALIZZARE DATI IOT SU AWS: UN SEMPLICE ESEMPIO CON AURORA E ATHENA

Amazon Athena	Amazon Kii	nesis Data Fi	rehose	Ama	azon Rek	ognition	Amaz	on S3
Aurora Serverless	AWS Glue	Databrew	Data a	nd An	alytics	Data Ing	estion	
Data Visualization	Dataset	Internet of	<sup>-</sup> Things (	(loT)				



beSharp | 11 Dicembre 2020

Con l'avvento dell'Internet of Things, il numero di device connessi sta aumentando in modo esponenziale, così come la quantità di dati da essi generati. Per questo l'acquisizione e l'analisi dei dati è diventato uno degli argomenti più scottanti dell'attuale panorama IT. AWS offre un'ampia gamma di servizi che ci consentono di importare, raccogliere, archiviare, analizzare e visualizzare enormi quantità di dati in modo rapido ed efficiente.



In questo breve articolo andremo a presentare un'applicazione molto semplice, ma reale che abbiamo sviluppato per rappresentare la pipeline di acquisizione e analisi dei dati nel contesto di eventi e conferenze AWS e IoT.

Abbiamo modificato una macchina da caffè moderna tramite l'utilizzo di componenti elettronici personalizzati quali un Raspberry Pi Zero e una microcamera, affinché, al premere del pulsante di erogazione, scattasse una foto alla persona di fronte e la caricasse su S3. A seguito dell'upload della foto, Amazon Rekognition, triggerato da AWS Lambda, si occupa di analizzarla, individuando parametri quali: – esiste la persona nella foto? – la persona ha gli occhiali? – ha la barba? – ha la baffi? – sta sorridendo?

A termine dell'analisi dell'immagine, se questa contiene il volto di una persona, un record viene scritto dalla funzione Lambda in un db serverless Aurora MySQL insieme all'output del modello di Machine Learning di Rekognition. Infine abbiamo sviluppato una semplicissima applicazione web collegata al database per la visualizzazione delle statistiche e una query con AWS Athena che pulisce i dati e li sposta in un bucket S3 come file parquet.

Di seguito è mostrato uno schema dell'infrastruttura proposta.



Passiamo ora a descrivere i passaggi di un comune processo di acquisizione e trasformazione dei dati e come li stiamo organizzando nella nostra applicazione.

## La fase di importazione / archiviazione

In AWS un flusso di inserimento dati molto comune consiste nell'utilizzare AWS IoT Core (Secure MQTT) o Api Gateway (API REST o Websocket) come entry point di dati, connettendolo direttamente a Kinesis Firehose (utilizzando regole IoT o Api gateway Service integrations) e infine sfruttare le potenti funzionalità di Firehose per il buffering dei dati, la trasformazione del buffer (funzioni AWS Lambda), la crittografia del flusso (AWS KMS), la compressione dei dati (GZIP) e la consegna dei dati a batch di messaggi compressi e crittografati automaticamente sia ad uno storage di tipo duraturo (S3) che a un data warehouse (AWS Redshift) per query analitiche complesse sull'enorme quantità di dati raccolti.

Avere sempre tutti i dati acquisiti salvati in AWS S3 è un passaggio essenziale, non solo come salvavita in caso di problemi con altri archivi più caldi ma anche per creare un data lake condiviso che può essere successivamente analizzato con Athena, EMR, Glue Jobs, Glue Databrew e anche strumenti esterni.

Inoltre è possibile utilizzare Firehose per fornire direttamente i dati ad AWS ElasticSearch per analisi in tempo reale e, se necessario, è anche molto facile fornire i batch di dati importati ad un database relazionale (ad esempio Aurora Serverless Postgres / MySQL) utilizzando AWS Data Migration Service o funzioni Lambda basate su eventi. La migrazione dei dati inseriti (o di un'aggregazione di essi) ad un database relazionale esistente è spesso molto utile se si ha bisogno di arricchire un'applicazione legacy esistente che già utilizza il database.

Nel caso si decida di utilizzare le funzioni Lambda per spostare i dati importati su Aurora, metodo solitamente più veloce e scalabile, è possibile sfruttare direttamente le funzioni Lambda di trasformazione di Firehose o una funzione differente attivata ogni volta che Firehose scrive un oggetto su S3.



Una peculiarità di Firehose consiste nella possibilità di aggiungerlo anche in un secondo momento! Nella nostra semplice applicazione non lo utilizziamo immediatamente, pertanto le immagini e le analisi vengono salvate direttamente in S3 e AWS Aurora Serverless MySQL da Lambda Functions, nel caso in cui il flusso generato dall'applicazione dovesse crescere potremmo successivamente integrarlo senza difficoltà alcuna!

## Fase di analisi

Una volta archiviati i dati, è il momento di analizzarli. In questo caso le metodologie possono differire notevolmente. Gli esempi più comuni vanno dalle semplici query eseguite in database relazionali, ai job analitici complessi eseguiti nei data warehouse di Redshift, oppure all'elaborazione in tempo reale utilizzando EMR o ElasticSearch.

Nel nostro caso possiamo semplicemente eseguire query utilizzando il backend della nostra applicazione web e visualizzare i risultati nel browser.

Tuttavia, in futuro potremmo essere interessati ad eseguire query molto più avanzate sui nostri dati e magari fare qualche check sulla qualità del dato o training per modelli di Machine Learning. Per rendere possibili queste espansioni, occorre quindi spostare i dati da Aurora a S3 per analizzarli con job Glue e Databrew e, se necessario, caricarli facilmente con Apache Spark da Glue o AWS EMR. Per fare ciò possiamo seguire diversi strade: ad esempio potremmo usare il servizio AWS DataMigration per spostare i dati su S3 come file Parquet oppure potremmo creare un Glue Job, caricare i dati usando Glue Connection da RDS con Spark e poi scriverli in S3.

Dopo questo passaggio, sarebbe necessario eseguire un crawler Glue per creare un DataCatalog che verrà utilizzato da Athena e Glue per query e jobs.

Qui tuttavia mostreremo un percorso diverso e talvolta molto più flessibile per esportare i dati in modo pulito e catalogarli da un database relazionale: **Athena custom data source.** 

Per impostazione predefinita, Athena viene fornito con l'integrazione S3 – Glue Data Catalog, ma di recente AWS ha dato la possibilità di aggiungere un'origine dati personalizzata, ad esempio database connessi tramite JDBC, AWS CloudWatch o l'esecuzione di query su S3 ma utilizzando un metastore Apache Hive personalizzato. Nel nostro caso siamo interessati a connetterci a MySQL Aurora Serverless. Per fare ciò dobbiamo andare su Athena Home, configurare un workgroup denominato AmazonAthenaPreviewFunctionality e quindi aggiungere una path S3 di output delle query di Athena:

Ath	na Query editor	Saved queries	History Data so	tos Molgogi AnzenAlden	Tutorial	Help What's new
Workg	roups					
Use wor Learn m	igroups to separate us vie (2	ers, tearns, applica	rtions, er workloads, a	d to set limits on amount of data each query or the entire workgroup can process. You can also view query-related metrics in AWS CloudWatah.		
Create	View det	tails Switch we	arligroup			
	Name		Description	Creation time	Warkge	roup status O
	ArrazonAthenaPreview	Functionality	This functionalit	is a part of the Athena Preview Festares and should be run in the workgroup named AmazanAthenaPreviewFunctionality 2020/12/11 11:st7:02.UTC+1	Enabled	
	primary			2020/02/05 16:54:03 UTC+1	Enabled	

Dopodiché possiamo tornare alla home di Athena e selezionare Connect Data Source:

Athena	Query editor	Saved queries	History	I
				ວ
Data source		Conn	ect data sour	се
AwsDataCatalog				

Ci viene presentata una pagina web dove dobbiamo selezionare il tipo di sorgente dati: optiamo per Query a data source (beta) e MySQL:

Athena Guery editor Saved queries History	Data sources Workproup : AmazonAthen	Settings
Connect data source		
Step 1: Choose a data source	Choose where your data is located	
Step 2: Connection details	Athena queries data where it is. Data is not loaded or moved. Learn more 🕑	
	Query data in Amazon 53     Query a data source	ze (beta)
	Choose an external data catalog. Configure a connect	tor for common data sources.
	Choose a data source (beta)	
	Choose the data source to query with Athena. After you choose a data source, you will configure a Lambda function to	handle the connection. Learn more 🧭
	Amazon CloudWatch Logs	loudWatch Metrics
	Amazon DocumentDB	lynamoDB
	Amazon Redshift	Base
	• wys MySQL	ji.

A questo punto ci viene chiesto di inserire il nome e la descrizione del nuovo catalogo e di selezionare o creare una funzione Lambda per gestire la connessione. Impostiamo il nome desiderato e facciamo clic su Configura nuova funzione AWS Lambda.

Lambda function     Choose or configure a new AWS Lambda function to connect to the data source.       Choose Lambda function        •	Choose a Lambda function tha	t is configured to connect to your data source,	or cr	eate and configure a Lambda function to	handle the connection. Learn	more 🗗	
Choose Lambda function   Choose Lambda function   Catalog name Create a unique name to specify this data source within a SQL statement.  Interaction  Law up to 127 characters and must be unique within your account. It cannot be changed after reaction. Valid characters are a z, A Z, O A (undercore), @ (of) and  Description  Interaction catalog	Lambda function	Choose or configure a new AWS Lambda function	n to c	connect to the data source.			
Catalog name         Create a unique name to specify this data source within a SQL statement.           International         International           Use up to 127 characters and must be unique within your account. It cannot be changed after reaction. Valid characters are a z, AZ, O.4., (and encode), ((i) and -0 yiphien).           Description         International		Choose Lambda function *	0	Configure new AWS Lambda function 🖉			
Instanticodo     Use up to 127 characters and must be unique within your account. It cannot be changed and recentors valid characters are a 2, A 2, O 9 (underscore), @ (At) and(bypthen).       Description     Instanticolo catalog	Catalog name	Create a unique name to specify this data sourc	e with	in a SQL statement.			
Ute up to 127 characteris and must be unique within your account. It cannot be changed after reantion. Valid characters are a 2, A7, D 9, _ (underscore), @ (d) and - (hyphen).		iotarticolo					
Description Instanticolo catalog		Use up to 127 characters and must be unique within changed after creation. Valid characters are a-z, A-Z - (hyphen).	your i 0-9, _	account. It cannot be (underscore), (i) (at) and			
	Description	iotarticolo catalog					
Use up to 1024 characters.		Use up to 1024 characters.					

Viene presentata questa pagina in cui bisogna inserire l'uri di connessione JDBC per Aurora e selezionare la subnet e il security group per la funzione Lambda che Athena utilizzerà per stabilire la connessione JDBC. Vanno scelti accuratamente, altrimenti la Lambda non raggiungerà l'istanza Aurora!

Audientian actitions
Abhreann serrings
Application name The stack name of this application created via AWS CloudFormation
AthenaJdbcConnector
scretistane/write Defin to enter source-based authorization policy for "scretismusger-GetSerentVolue" action. E.g. All Advers JDMC Federation server names can be prefixed with "AdversJDMC referention", policy will allow manascretismusgers/GMUSS-Bejond SUMSS-Accountify Secret-Adversa/GMC reference". Parameter value in this care double to "AdversJDMC reference". Parameter value in this care double to "Adver
plillbacet ne ranne d'he backet where this function can yell data.
▼ JdbcConnectorConfig
SefulationnectionString The sefulat connection string is used when catalog in "lumbda"s(Lambda"auctionstame)," Catalog specific Connection Strings can be added later. Format: S[DatabaseTspec]//S[DatabaseTspec]/
NisableSpillEncryption 1st to Tate: data spilled to S3 is encrypted with AES GOM
false
ambd/inction/ane for some you will yoe to the catalog is Athena. It will also be used as the function rame. This name must satisfy the pattern 1(s-d>9_1)(3d))
ambdaMemory ambda memory in M8 (min 128 - 3008 ma).
3008
ambdaTimeout Annum Lambda Invocation runtime in seconds. (min 1 - 900 max)
900
iccurityGroupIds The or more SecurityGroup Dis corresponding to the SecurityGroup that should be applied to the Lambda function, (e.g. sg1,sg2,sg1)
SplitPrefix. The perfex within SplitLoder where this function can split data.
athena-spill
Submettals The of more Solvent Disconsequencing to the Solvent that the Landad Function can use to access you data source. (in g. submet.)
I adenoviedge that this app creates custom IAM roles. Info
Cancel Previous Dealors

Il prefisso del Secret viene utilizzato per memorizzare le credenziali del database in AWS Secret Manager, questo è essenziale per l'ambiente di produzione e se lo si lascia vuoto, l'integrazione non verrà creata. Dopo aver selezionato deploy e selezionato la Lambda appena creata nella dashboard di Athena, verrà creato un nuovo catalogo diverso dallo standard AwsGlueCatalog:



Tuttavia all'inizio Databeses e tabelle non appariranno. Controllando i log della Lambda su CloudWatch si troverà un errore del tipo:

Catalog is not supported in multiplexer. After registering	the catalog <b>in</b>	Athena, mus
t <b>set</b> 'iotarticolo_connection_string' environment variable	in Lambda. See	JDBC connect
or README for further details.: java.lang.RuntimeException		

Impostiamo quindi la variabile di environment richiesta per la funzione Lambda utilizzando la stessa stringa di connessione JDBC usata come stringa DefaultConnection nel passaggio precedente. Dopodiché la connessione funzionerà e si potrà interrogare il DB direttamente da Athena! Ottimo!

Data source Connect data sour	e	New gee	ry 1 💿 New query 2	O New query 3	O New query 5	O New query 6	O New query 7	O New qu	ery 8 O	+			
istaticalo -	) (	1 SEL	ECT * FROM "lotart	icelor.flotf.fc	offees" limit 2	1991							
Database													
lot ·													
Filter tables and views		Run que	Save as (Run	line: 3.28 seconds, 5	outa scanned: 0 KZ()								Format query Cle
* Tables (3)		Use Ctrl + I	Enter to run overy. Ctrl + So	ace to autocomplete									
+ ar_internal_metadata (Partitioned)													
+ coffees (Partitioned)													
+ schema, migrations (Partitioned)		Results											
		▲ id *	phota_art *					senilo	beard ~	nustache *	glasses -	coffee_bour_str ~	partition_name *
		1.1	UH_#_caso					true	false	true	true	2017-06-19 00:00:00	
		2 2	Url_a_caso					true	false	true	true	2017-06-19 00:01:00	
		3 3	UH_#_calo					true	fabre	true	true	2017-06-19 00:02:00	
		4 4	UIL_a_caso					true	false	true	true	2017-06-19 15:03:00	•
		5 5	UH, a, cano					true	fabre	true	true	2017-06-19 15:00:00	
		6 6	U1_8_0850					true	folse	true	true	2017-06-19 16:01:00	
		7 7	https://s3-eu-went-1.	ornazonavo.com/iet	becoffee/2018-03-	25116-45-43.5992.0	0	true	false	false	false	2018-03-26 16:45:46	
		8 8	https://s3-eu-west-1.	arrazonaws.com/iet	becoffee/2018-03:	26716:52:58.1422.j	9	false	ose	true	true	2018-03-26 16:53:00	

Ad uno sguardo più attento, però, notiamo immediatamente che qualcosa è in conflitto con i dati: ecco una schermata di ciò che possiamo leggere direttamente da MySQL:



Come si può vedere Athena è abbastanza intelligente da convertire i dati tinyint(1) in bool ma non riesce a leggere le colonne datetime da mysql. Ciò è dovuto ad un problema molto noto con il connettore jdbc e la soluzione più semplice è creare un nuovo campo dove datetime è una stringa in formato datetime Java:

```
UPDATE coffees SET coffees.coffee_hour_str=DATE_FORMAT(coffee_hour, '%Y-%m-%d %H:%i:% s');
ALTER TABLE coffees ADD COLUMN coffee_hour_str VARCHAR(255) AFTER coffee_hour;
```

A questo punto Athena potrà leggere il nuovo campo. Ed ora siamo pronti per un bellissimo trucco: andiamo nella dashboard di Glue e creiamo un nuovo Database: un database è solo un contenitore logico per metadati, si può scegliere il nome che si preferisce.

	Databases A database is a set of associated table definition	ons, organized into a logical group.	
AWS GIUE			×
	Action +	Add database	
Databases	Name	Database name	
Tables	covid-19	iotarticologlue	
Connections		<ul> <li>Description and location (optional)</li> </ul>	_
Crawlers		Location A	
Classifiers	Iotarticologiue	Enter location	
Schema registries	O sampledb		
Schemas		Description	_
Settings		Enter description	
AWS Glue Studio New			
Workflows			
Jobs			
ML Transforms			
Triggers			
Dev endpoints			
Notebooks			
		Create	0
		c	

A questo punto possiamo tornare ad Athena ed eseguire una query come questa:

```
CREATE table iotarticologlue.coffees
WITH (
    format='PARQUET', external_location='s3://besharp-athena/coffees_parquet', parquet_c
    ompression='GZIP'
) AS SELECT photo_url,smile,beard,mustache,glasses,coffee_hour_str FROM
    "iotarticolo"."iot"."coffees"
WHERE photo_url LIKE 'https://%';
```

Ciò creerà una nuova tabella nel database che abbiamo appena aggiunto al nostro catalogo di dati Glue e salverà tutti i dati in S3 come file GZIP Parquet. Inoltre è anche possibile, se lo si desidera, cambiare la compressione (es. Snappy o BZIP).

Run query Save as Create ~

Oltre ad esportare i dati come Parquet, la query andrà ad eliminare quelli con la formattazione errata per l'url di S3.

Abbiamo quindi un modo super veloce per esportare il nostro db in S3 come parquet e creare automaticamente il catalogo Glue.



Diventa quindi semplice visualizzare questo nuovo catalogo in AWS Glue Databrew: andiamo alla dashboard di Databrew e creiamo un nuovo progetto

=	D	3538×	ew > Projects																		
		Pro	pjects (1) nets								Open project View job	66	etails View line	1091	<ul> <li>Bun job</li> </ul>	Acti	0115 🔻		Create	s projec	a O
MEACT			Project name	٣	Associated dataset	v	Attached recipe	v	Jobs	,	Create date 🔹		Created by	v	In use by		-	Tags			٧
			covid-19-databrew- project		covid19-enigme-jhu		covid-19-databrew- project-recipe				21 days ago November 20, 2020, 2:43:55 pm		diristian.calabrese								

e un nuovo set di dati nella sezione aggiungi dataset.

	Your so	urce from Data Catalo	og Info Tees parr	met/		
Data lake/data store	(B) Por	mission from MWS L	ako Eormat	ion will apply to datasets with this ison		
🗟 Amazon S3	- Pei	THISSION FROM AW'S L	ike Formau	on will apply to datasets with this icon		
AWS Glue Data Catalog	AWS (	Glue databases > i	otarticolo	glue		
🔁 Amazon S3 tables	Q	Search tables by n	те			
Amazon Redshift tables					< 1	> ©
🔅 Amazon RDS tables						
🕏 All AWS Glue tables		Table name	$\bigtriangledown$	Last updated		Size
Others	•	coffees		<b>a minute ago</b> December 11, 2020, 4:07:01 pm		3.6 KB

In caso di errore sarà necessario rinominare il file s3 come .parquet e scansionare nuovamente la tabella con i crawler Glue.

Et voilà una bellissima visualizzazione dei dati del nostro dataset completa di statistiche delle colonne!

iotarticolo	o-brev larticolo	v -brow-detaset   ){{ Sample: First	n sampla (199 row	u						Creat	te jøb 🔰 🐉	 ACTONS
5 2	<b>V</b>	ENIRE AL AL E			E Internet			B				E CON
© Viewing	5/6 col	umos 🔻 199 rows			N COSCI	spany		■ 6430 □ SCHDAA	🗟 recruz	🗉 Column details		×
	7	(# beard	7	🕐 mustoche	γ	d glasses	γ	alt caffee_hour_str	ν			
	64 199	Unipe 1	544 10	Orige 2	564, 100	Unique 2	144 <b>19</b>	Unique 1988	1001 100	👎 Boolean glasses		
101	92,89%	fithe	301 30.735	fabr	125 01.015	A04	107 53.77%	2010-03-25 18-05-06	1 0.5%			
	9.85%	but.	<b>55</b> 40.201	tur	36 Sk195	tur	92 46.15%	2018-03-25 18:30 00 2018-03-27 08:27 30 Alberter values	1 0.5% 1 0.5%	Column statistics	∛ Recommendatio	***
		false		fabre		felat		2018-03-25 16:45:45		Data quality		
		top		true		trac		2018-03-26 16:53:00	_	Data quanty		
		tue		true		true		2018-03-27-06-37-23				
		false		false		false		2018-03-27-06:43-32		VAUD WILDES	MISSING V	MLUES
		tue		true		false		2018-03-27 07:19:43				
		15.0		false		true		2018-03-27 07:20:48				
		true		true		trae		2018-03-27 07:22:17		Value distribution		
		false		false		true		2018-03-27 07:25:28		Unique 2	Total	1 199
		true		true		fulse		2018-03-27 07:27:11				
		toe		true		Dies .		2018-03-27 07:34:10				
		false		false		false		2018-03-27 07:36:23				
		title .		true		false		2018-03-27 07:41:39				
		false		false		trae		2015-03-27 07:44:01				
		toe		true		true		2015-03-27 07:47:40				
		false		false		false		2018-03-27 07:59:19		655	Luo	
		toe		true		true		2018-03-27-08:15:28				
		false		false		true		2018-03-27 08:21:15				
¢.		20		12.1				1943 AT 17 AT 17 AT 17		Unique values		
Zana 🗿		100% #										

### Conclusioni

In questo articolo abbiamo descritto un'applicazione IoT molto semplice che utilizza Amazon Rekognition e Amazon Aurora. Abbiamo spiegato come può essere migliorata sfruttando Firehose e infine abbiamo utilizzato Athena per trasformare e pulire i dati raccolti. Abbiamo visto anche come salvarli molto facilmente come parquet e come possono essere analizzati con Glue Databrew, Athena e altri strumenti AWS come EMR.

Avete mai provato configurazioni simili per i vostri processi di Data Analysis?

Fateci sapere! Saremo felici di offrirvi un caffè...connesso 😀

Per oggi è tutto.

Continuate a seguirci: ci vediamo tra 14 giorni qui su **#Proud2beCloud!** 



#### beSharp

Dal 2011 beSharp guida le aziende italiane sul Cloud. Dalla piccola impresa alla grande multinazionale, dal manifatturiero al terziario avanzato, aiutiamo le realtà più all'avanguardia a realizzare progetti innovativi in campo IT.

#### Get in touch

beSharp.it proud2becloud@besharp.it

Copyright © 2011-2021 by beSharp srl - P.IVA IT02415160189