

Home > AI/ML

Turning unexpected services into a data source: an unconventional example

20 June 2024 - 5 min. read

When we interact with something, we generate data.

In 2016, a single internet user was projected to create 1.7 MB of data per second every day by 2020.

Those figures were probably underestimated.

Data is everywhere and sometimes hidden in unexpected places, like little treasures. As we'll see, this can happen for services used for what can be considered a humble purpose, but they turn out to contain valuable information.



This fact reminds me of the history of the lobster: historically, lobsters were not desired and considered food for poor people. Early Americans even used lobster as yard fertilizer and fishing bait. Prisoners and indentured servants complained because they were forced to eat lobster very often. By the 1920s, the demand for lobster started to increase until they became the precious food we know.

Let's see how we can find data in "today's lobster".

CloudWatch RUM

CloudWatch RUM is a relatively new addition to Amazon CloudWatch services.

It monitors real users' activity to track website performance metrics (web vitals) and identify issues within user sessions. Its real power is its X-ray integration for serverless applications, so you can track an error that occurred in a lambda and see what the user was doing. My colleagues Alessio and Daniele already told us about X-Ray.

When you deploy the CloudWatch RUM snippet on your website, it sends metrics and events to the service endpoints. Data will be available for visualization in the dashboard.

This is an example:



You can also see sessions, user journeys, and detailed error statistics.

Filter by selecting or typing attributes and values	(ex. "browserName=Chro	me") Sel	ect a filter 🔻	1h 3	h 12h 1	d 1w 1M	Custom
Performance Errors Http requests	Sessions Event	ts Browsers & Devices	User journey Co	onfiguration			
Average session length 14m 26s	Er 9.0	ror free sessions		Average er 4.7 +72.2%	rors per sess	sion	_~~
Session starts Number of session starts over time							1
lesion starts 40 0	6. Jun 1200	7. Jun 12:00 8. Ju 9 Session	n 12.00 9. Ju starts	n 12:00 10.Jun	12:00	= 11. Jun 12:0	
Sessions (160) Q. Filter by selecting or typing keys and values (e	r. "sessionId=1234")			< 1	1234	567_	6 > @
Session ID	▼ Duration ▼	Started at	⊽ Pages ⊽ Iss	aues 🔻 🛛 Browser 🔻	Device v	Language 🔻	Traces 🖸
0 69849a75-754a-497a-b765-4e6bdff4b62	7 55:08	Today Jun 10, 2024, 11:54:03 AM GMT	31 31	Chrome v125.0.0.0	desktop Windows	it-IT	c
C c852ac48-3f80-446f-a507-f5048d67daa5	12:56	3 days ago Jun 7, 2024, 6:55:29 PM GMT+2	27 23	Chrome v125.0.0.0	desktop Mac OS	it-IT	c
ssion details: c852ac48-3	3f80-446f-a	507-f5048d670	laa9				
ession details							
ession details	auration Start	End		Pages Issues	Browser	Device	Langu
ession details ession ID 1 152ac48-3f80-446f-a507-f5048d67daa9	Puration Start 12:56 3 days ag Jun 7, 202	20 3 da 4, 6:55:29 PM GMT+2 Jun 7	ys ago , 2024, 7:08:25 PM GMT+2	Pages Issues 27 23	Browser Chrome v125.0.0	Device Desktop Mac OS, 10.15.7	Langu it-IT
ession details ession ID IS2ac48-3H0-446F-a507-H5048667dax9 I age views	Duration Start 12:56 3 days ar Jun 7, 202	20 3 da 4, 655:29 PM GMT+2 Jun 2	ys ago 2024, 7:08:25 PM GMT+2	Pages Issues 27 23	Browser Chrome v125.0.0	Device Desktop Mac OS, 10.15.7	Langu it-IT 1 2 3
ession details ession ID 1 522ac48-3H0-446F-507-F5048667dax9 1 age views sge ID	Duration Start 12:56 3 days a Jun 7, 202 Timeline	20 3 da 4, 655:29 PH GMT+2 Jun 7 Issues Load time	ys ago 2024, 7:08:25 PH GMT+2 First input delay	Pages Issues 27 23 Cumulative layer	Browser Chrome v125.0.0.0	Device Desktop Mac OS, 10.15.7	Langu it-IT 1 2 3 atful paint
ession details ession D 522ac46-3507-450466758ar9 1 age views 199 ID	Duration Start 12:56 3 days a Jun 7, 202 Timeline 0:00	0 3 da 4,65529 FM GMT+2 3 da 1ssues Load time 0 1k ms	rs ago 2024, 708:25 PM GMT+2 First input delay 12.7 ms	Pages Issues 27 23 Cumulative layer	Browser Chrome v125.0.0.0	Device Desktop Mac 05, 10.15.7	Langu it-IT 1 2 3 etful paint
ession details ession D 552x<6-340-466-4507-f304667-fax9 3ge views 1gg ID	Duration Start 12:56 3 days as days and a days of the second seco	0 3 da 4, 655:29 PM GMT-2 3 da 3, an 7 1 Issues Load time 0 1 k ms 0 163 ms	rs ago 2024, 70825 PH GHT+2 First input delay 12.7 ms	Pages Issues 27 23 Cumulative layer -	Browser Chrome v125.0.0.0	Device Desktop Mac 05, 10.15.7 Largest conter 1.1 s	Langu it-IT 1 2 3 atful paint
ession details ession 10 1 1 S522xc63-3560-456450726x9 1 1 age views age views	Duration Start 12:56 3 days as Jun 7, 202 Timeline 0.00 0.07 0.08	End p0 4, 655:29 PH GMT+2 3 dm 7 Issues Load time 0 1k ms 0 163 ms 0 -	rs 100 2024, 70125 PH GHT+2 First input delay 12.7 ms -	Pages Issues 27 25 Cumulative layor - -	Browser Chrome +125.0.0.0	Device Desktop Mac 05, 10.15.7 C Largest conter 1.1 s	Langu it-IT 1 2 3 atful paint

Besides being a useful tool, it can help us turn our business into data-driven.

While this service is always used as a monitoring and reporting tool, it can also be used to extract data to reveal hidden insights: in fact, its real value relies upon the events and metrics captured by the service.

Let's unveil them and see their real potential.

As we said, the first kind of data (readily available and accessible) is metrics, such as page views, session counts, and so on. You can find a list here.

You can also define custom metrics based on events. For example, you can define a custom metric that counts only Chrome and Safari access from the US.

As with every metric, you can set up alarms and define metric maths, but this isn't something new for us.

The second kind of data you can collect is less evident but more valuable: raw events generated by real users.

When you configure RUM, you can enable CloudWatch Logs integration. This integration stores all user-generated activity events in a JSON log group.

Data storage

Choose to send data to your CloudWatch Logs account for longer retention. Additional pricing applies. Learn more 🛂

Check this option to store your application telemetry data in your CloudWatch Logs account. Learn more The name of the log group created will be /aws/vendedlogs/RUMService_<Name>+<first 8 digit of app monitor ID>.

Be aware that log groups don't set data retention by default, and RUM generates a lot of data. This sample is for a single-page view.

Log You ci	og events C Actieve v Start talling Crute methe filter berefer i som og som til som og som til filter pattern g					
Q, /	ilter events - press enter to search	Clear 1m 30m 1h 12h Custom 🗐 UTC timezone 🔻 Display 🔻 🛞				
٠	Timestamp	Message				
		No older events at this moment, Anny				
*	2024-06-11709:38:56.0002	- (*vert_timestamp*1710000730000,*vert_type*1*com.mazon.rum.session.start_verent,*ventLid*1*74001Fbc-0027-0078-0433-0505770122**,*vent_version*1*12.0.0*,*fug_stream*1*2024-04-1172**,*application_id*1*400045a				
٠	2024-06-11709:38:56.0002	("event_timestamp":1718000710000, "event_type":"com.mazon.rum.page_view_event", "event_bit":"00337456-ac2-4465-000c-50800015580a", "event_version":"1.0.0.", "igstream":"2020-00-13782", "application_id":"400046456-7790.				
٠	2024-06-11709:38:56.0002					
٠	2024-06-11709:38:58.0002	("event_timestamp":1719990739000, "event_type")"com.amizon.rum.largest_contentfvl_paint_vent", "event_til":171902088-0022-000-6972-00035912727, "event_vention", "1.0.4", "log_stream":12020-04-117127, "paptication_in".				
٠	2024-06-11709:38:58.0002	- (*vert_limiting*171000073000,*vert_lyp*1*cm.mazon.rm.com/atler_layor_shift_rvent*,*vert_l*1*950100-2526-455-667-3176652012*,*vert_versin*(*1.0.0*,*10_0_strem*1*2224-46-13702*,*pgllcation_i4*).				
		No newer events at this moment. Auto retry passed. <u>Resume</u>				

So, we finally have navigation and performance data ready; this is a sample entry for an event:

```
{
    "event_timestamp": 1718098736000,
    "event_type": "com.amazon.rum.performance_navigation_event",
```

```
"event_id": "91ee89c6-044d-476c-8227-743c8f1b21d1",
"event version": "1.0.0",
"log stream": "2024-06-11T02",
"application id": "48e6845a-7799-4e76-90f6-a3602dac887a",
"application version": "1.0.0",
"metadata": {
    "version": "1.0.0",
    "browserLanguage": "en-US",
    "browserName": "Edge",
    "browserVersion": "125.0.0.0",
    "osName": "Linux",
    "osVersion": "x86 64",
    "deviceType": "desktop",
    "platformType": "web",
    "pageId": "/",
    "interaction": 0,
    "title": "test",
    "domain": "blog.besharp.it",
    "aws:client": "arw-script",
    "aws:clientVersion": "1.16.1",
    "countryCode": "IT",
    "subdivisionCode": "PV"
},
"user details": {
    "userId": "ac3f587b-3887-4661-8200-52d8ce5768f7",
    "sessionId": "13df94e4-7076-44a2-8b49-5dbb49e55e59"
},
"event details": {
    "version": "1.0.0",
    "initiatorType": "navigation",
    "navigationType": "navigate",
    "startTime": 0,
    "unloadEventStart": 0,
    "promptForUnload": 0,
    "redirectCount": 0,
    "redirectStart": 0,
    "redirectTime": 0,
    "workerStart": 0,
    "workerTime": 0,
    "fetchStart": 4.799999998137355,
```

```
"domainLookupStart": 34,
    "dns": 0,
    "nextHopProtocol": "h2",
    "connectStart": 34,
    "connect": 15.79999999813735,
    "secureConnectionStart": 37.9000000037253,
    "tlsTime": 11.89999999441206,
    "requestStart": 50.1000000055879,
    "timeToFirstByte": 151.899999994412,
    "responseStart": 202,
    "responseTime": 1.100000005587935,
    "domInteractive": 276.2000000018626,
    "domContentLoadedEventStart": 276.2000000018626,
    "domContentLoaded": 0,
    "domComplete": 452.100000005588,
    "domProcessingTime": 249,
    "loadEventStart": 452.100000005588,
    "loadEventTime": 0.09999999962747097,
    "duration": 452.2000000018626,
    "headerSize": 300,
    "transferSize": 944,
    "compressionRatio": 2.0683229813664594,
    "navigationTimingLevel": 2
}
```

Knowing where our data is, we can now export it into our DataLake, for example, integrating CloudWatch logs with Kinesis Firehose to write events on S3.

}



Once data is exported, you can leverage all this information and integrate it into internal analysis tools, and the possibilities are endless.

Think about correlating sales for an e-commerce website with user location, page response time, or session navigation data.

You could discover unexpected results, such as weather conditions influencing sales in a particular area. You can also generate custom events in RUM that can act as placeholders and ease the job of behavior correlation.

With all this information, you can better understand your users and your SEO positioning, streamline internal processes, or simply have a better monitoring system so you will know who and when to alert in case of anomalies and outliers in data. Don't forget about making pretty dashboards for internal and external customers; everyone loves a good dashboard :)

Once you enrich your existing DataLake, even with information intended for monitoring, you can make accurate data-driven decisions.

This solution does not aim to replace Google Analytics but to ease data integration and analysis: You can export data from GA-4 to BigQuery, but if you have an AWS DataWareHouse solution in place, its integration can take time. We aim to support business decisions rather than put effort into maintaining integrations between different systems.

To Conclude

Choosing the right data source instead of working on integrations can be beneficial. Sometimes, you have to dig into services to find that you already have all the information you want (and sometimes a lot of it!).

As you can see, if you are already using CloudWatch RUM, you can leverage its integration with AWS services with little effort, using only managed services, even if you didn't think about it.

Have you ever thought about getting data from unusual sources?

What kind of correlation can you think of after knowing this integration? Let us know in the comments!

About Proud2beCloud

Proud2beCloud is a blog by beSharp, an Italian APN Premier Consulting Partner expert in designing, implementing, and managing complex Cloud infrastructures and advanced services on AWS. Before being writers, we are Cloud Experts working daily with AWS services since 2007. We are hungry readers, innovative builders, and gem-seekers. On Proud2beCloud, we regularly share our best AWS pro tips, configuration insights, in-depth news, tips&tricks, how-tos, and many other resources. Take part in the discussion!



Damiano Giorgi

Ex on-prem systems engineer, lazy and prone to automating boring tasks. In constant search of technological innovations and new exciting things to experience. And that's why I love Cloud Computing! At this moment, the only "hardware" I regularly dedicate myself to is that my bass; if you can't find me in the office or in the band room try at the pub or at some airport, then!

Copyright © 2011-2024 by beSharp spa - P.IVA IT02415160189