3 feature rivoluzionarie (+1) di Amazon CloudWatch che probabilmente non conosci

12 Gennaio 2024 - 2 min. read

Amazon CloudWatch *Logging and Monitoring*

Tutti i professionisti IT hanno un debole per le infrastrutture stabili in cui tutto funziona senza intoppi, soprattutto durante il turno per il supporto 24/7.

Purtroppo però, nella maggior parte dei casi non è semplice nemmeno riuscire a capire se l'infrastruttura funziona correttamente oppure no, se è implementata a regola d'arte o se sta per esplodere tutto.

Questa situazione è strettamente legata al concetto di ***observability***, definito come

la capacità di monitorare, misurare e comprendere lo stato di un sistema o di un'applicazione esaminandone l'output, i log e le metriche.

Quando un'organizzazione cresce in dimensioni e complessità, la capacità di estrarre informazioni sullo stato di salute dell'infrastruttura diventa fondamentale. In queste situazioni le metriche sono il miglior strumento per determinare lo stato di salute di un'infrastruttura.

Le metriche sono una rappresentazione numerica dei dati misurati nel tempo; sono utili per identificare tendenze, previsioni e anomalie. Affinché portino valore, le metriche devono essere centralizzate, elaborate, aggregate e presentate in modo significativo.

All'interno di AWS, i termini *monitoring* e *observability* dovrebbero immediatamente far pensare ad Amazon CloudWatch. Tutti conoscono Cloudwatch, giusto? I log groups, le metrics e gli events, tutto qui, corretto?

E se vi dicessi che CloudWatch è molto di più?

Ecco dunque 3+1 funzionalità di Amazon CloudWatch che non tutti conoscono.

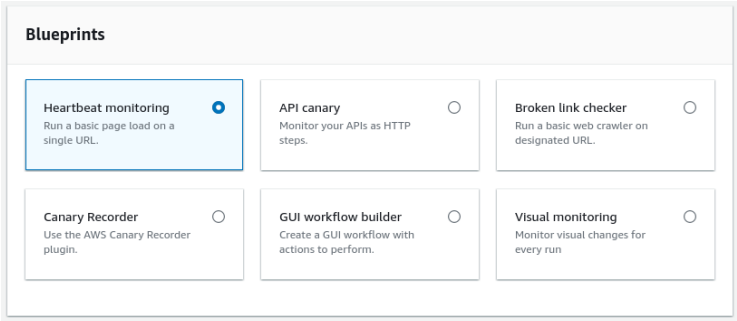
1. Synthetics

Questa è probabilmente la mia feature preferita: con Synthetics è possibile creare dei "*canaries*", ovvero degli script che vengono eseguiti in modo programmato per monitorare un sito web, un'API o un endpoint in generale. Con un canary è possibile simulare le azioni di un utente reale, rendendo davvero semplice scoprire se una pagina web non risponde o se un'API restituisce errori.

Esistono 3 metodi per creare un canary:

- utilizzare un blueprint fornito da Cloudwatch;
- scrivere uno script da zero all'interno dell'editor;
- importare script esistenti da S3.

L'opzione più interessante è l'utilizzo dei blueprint in quanto mostrano il potenziale di questo strumento. L'interfaccia è chiara, i parametri impostabili dall'utente sono ben definiti, facili da compilare e le modifiche aggiornano in tempo reale lo script.



Come mostrato nell'immagine, questi sono i casi d'uso previsti nei blueprint:

- Heartbeat monitoring: questo template richiede solamente l'URL da cui controllare il response code e, opzionalmente, fornisce lo screenshot della pagina.
- API canary: è fondamentalmente uguale al blueprint precedente, ma pensato per le API. Di base controlla solo il response code, ma è molto facile da estendere aggiungendo altri controlli sul corpo della risposta. Il codice dello script include alcuni commenti per aiutare gli utenti nel modificarlo.
- Broken link checker: un crawler che cerca tutti i link in una determinata pagina, li segue per assicurarsi che siano ancora attivi e, se impostato, ne fa uno screenshot.

- Canary Recorder: questa funzionalità dimostra quanto CloudWatch abbia semplificato il lavoro per l'utente. Per funzionare viene richiesta l'installazione del plugin per Google Chrome "CloudWatch Synthetics Canary" che, dopo l'attivazione, registra ogni click e pulsante premuto e genera uno script che replica tali azioni. Il plugin si basa sul progetto Headless Recorder.
- GUI workflow builder: consente di creare lo script "graficamente". È richiesta la selezione di un'azione (ad esempio un click) da un menù a tendina e dello specifico attributo html su cui eseguire l'azione.
- Visual monitoring: funziona scattando screenshot della pagina web selezionata e confrontandoli alla ricerca di differenze. È possibile definire una soglia di varianza visiva al di sotto della quale non viene registrata alcuna modifica.

Gli script possono essere scritti sia in Node.js utilizzando Puppeteer che in Python con Selenium, anche se alcuni blueprint sono privi dell'implementazione in Python. I runtime supportati consistono quindi in varie combinazioni di questi linguaggi, librerie e versioni di Chromium.

2. Metriche derivate

Se vi piace analizzare le metriche, è molto probabile che vi piaccia anche la matematica. Se non fosse questo il caso, non preoccupatevi, in questo paragrafo non verrà utilizzata alcuna funzione differenziale... forse.

Nelle occasioni in cui vi sia la necessità di una logica più complessa rispetto al semplice controllo dell'utilizzo della CPU, Cloudwatch viene in aiuto con la possibilità di utilizzare metriche derivate dalla combinazione di altre metriche.

Supponiamo che uno sviluppatore abbia appena corretto un bug che si presenta all'avvio di una funzione Lambda e, dopo la correzione, voglia verificare che la funzione non presenti ancora problemi nei primi millisecondi di esecuzione. Per fare ciò, il programmatore potrebbe verificare la presenza di errori mentre monitora il tempo di esecuzione medio della funzione.

L'immagine fornita mostra un'implementazione pratica di questo processo.

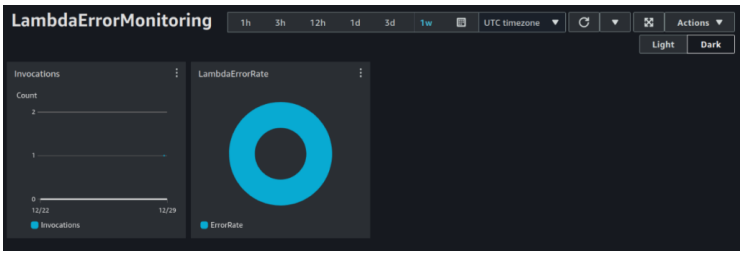
Naturalmente questo è un esempio molto semplice, ma attraverso l'uso di condizioni e delle funzioni presenti nel "math menu" è possibile creare espressioni più complesse. Il menù in questione offre moltissime funzioni, a partire dagli operatori matematici più comuni fino a condizioni, conversioni, ordinamento, filtri e ad altre query più specifiche.

Le metriche appena create possono poi essere aggiunte ad una dashboard (spoiler) o utilizzate per la creazione di allarmi.

3. Dashboards

Si può considerare monitoring se non ci sono delle dashboard? Non credo. Fortunatamente Cloudwatch mette a disposizione un servizio per la loro creazione. Le dashboard di Cloudwatch rendono molto semplice il monitoraggio di un intero workload, anche se composto da servizi diversi in regioni diverse e su account diversi, tutto in un'unica pagina.

Ad esempio, ecco una semplice dashboard che mostra il numero di chiamate a una particolare funzione Lambda e il suo tasso di fallimento.



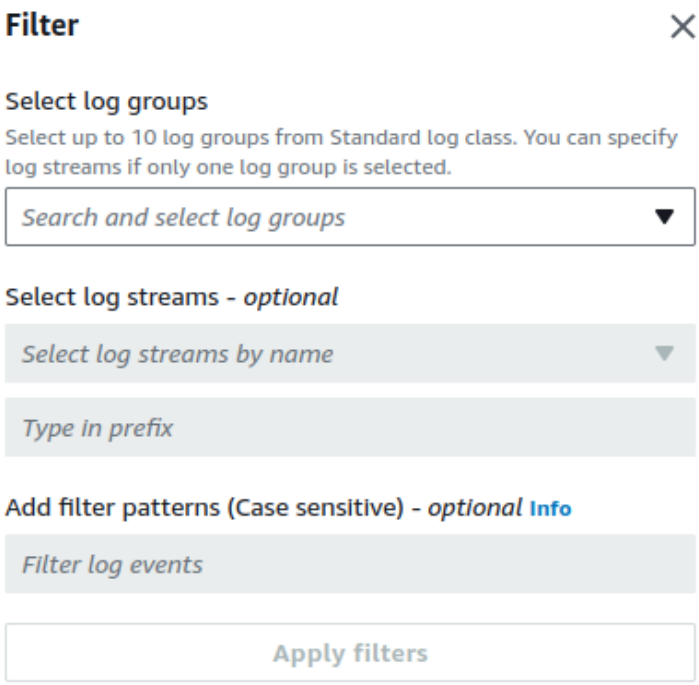
"E se la dashboard dovesse essere vista da persone che non hanno un account AWS?" Nessun problema. È possibile condividerla a un elenco di indirizzi e-mail selezionati o specificando un Identity Provider (IdP) di terze parti. Il proprietario di uno degli indirizzi e-mail consentiti dovrà creare la propria password che poi potrà utilizzare per visualizzare la dashboard.

Inoltre, vale la pena menzionare le "Dashboard automatiche". Si tratta di dashboard preconfigurate, incentrate su specifici servizi e già popolate di informazioni. Queste dashboard si rivelano molto utili in quanto consentono un rapido monitoraggio delle risorse senza la necessità di crearne di personalizzate.

4. Bonus: Live Tail

Il nome di questa funzione potrebbe ricordare il celebre comando Unix presente nelle GNU coreutils... ed è giusto così! Live Tail replica la funzionalità di "tail -follow", che legge continuamente le ultime righe di un file, ma su CloudWatch.

Con Live Tail è possibile seguire tutti i log di un log group selezionato senza dover continuamente premere il pulsante di aggiornamento nella pagina dei log events. Non è tutto, dopo aver selezionato il log group, la selezione del log stream è opzionale. Quest'ultima parte potrebbe non sembrare di grande importanza, ma chiunque abbia mai sviluppato un Lambda conosce la fatica di dover cambiare continuamente i log group per visualizzare i log più recenti. Questa funzionalità permette quindi di eliminare alcuni click ad ogni aggiornamento del codice della funzione, risparmiando molto tempo (ed energia mentale) durante lo sviluppo e, soprattutto, durante il debugging.



Per facilitare l'estrazione delle sole informazioni rilevanti dai logs è possibile utilizzare una regex, alcuni esempi:

- *%ERROR%* visualizza tutti gli eventi di log contenenti la parola chiave “ERROR”
- *{ $.names = %Steve% }* visualizza i log JSON in cui Steve è presente nella proprietà "name"
- *[w1 = %abc%, w2]* visualizza gli eventi di log delimitati da spazi e con “abc” come prima parola

Per chi non ama le regex (comprensibile) esiste anche una sintassi più semplice:

- *error 404* visualizza solo gli eventi di log che includono sia “error” che “404”
- *?Error ?error* visualizza gli eventi di log che includono “Error” oppure “error”
- *-INFO* visualizza tutti gli eventi di log che non includono “INFO”

- *{ $.eventType = "UpdateTrail" }* visualizza tutti i log JSON in cui il valore del campo *eventType* corrisponde a “UpdateTrail”

Conclusioni

In conclusione, il monitoring è spesso sottovalutato. Non ci si deve limitare a raccogliere i log "per sicurezza"; la comprensione del valore estratto dalle metriche grezze è la chiave per il miglioramento dell'infrastruttura. Naturalmente un approccio corretto al monitoraggio richiede uno sforzo, ma non si tratta di energia sprecata, bensì di un investimento. Capire come l'infrastruttura si comporta e reagisce ai cambiamenti è essenziale per mantenere una buona business continuity, aiuta a comprendere certi comportamenti e rende più facile operare in modo consapevole.

Siete a conoscenza di altre magie di Amazon CloudWatch? Non vediamo l'ora di ascoltarle!

About Proud2beCloud

Proud2beCloud è il blog di beSharp, APN Premier Consulting Partner italiano esperto nella progettazione, implementazione e gestione di infrastrutture Cloud complesse e servizi AWS avanzati. Prima di essere scrittori, siamo Solutions Architect che, dal 2007, lavorano quotidianamente con i servizi AWS. Siamo innovatori alla costante ricerca della soluzione più all'avanguardia per noi e per i nostri clienti. Su Proud2beCloud condividiamo regolarmente i nostri migliori spunti con chi come noi, per lavoro o per passione, lavora con il Cloud di AWS. Partecipa alla discussione!

Andrea Pusineri

DevOps Engineer @ beSharp. Mi diverto a risolvere problemi e sono cintura nera nel trovarli. Linux enthusiast e security guy wannabe, mi piacciono le CTF e nel tempo libero sono un avido lettore di fumetti/manga/libri. btw I use Arch

---