

# Hands-on: strumentazione di AWS Lambda con AWS X-Ray per analisi e debugging delle applicazioni

19 Gennaio 2024 - 3 min. read

[Application Monitoring](#)

[AWS Lambda](#)

[Debugging](#)

[Microservices](#)

## Introduzione

Lavorare con i microservizi presenta molti vantaggi, ma presenta anche alcuni difetti. Uno di questi è senza dubbio il complicarsi delle attività di debug e monitoraggio dell'intera applicazione.

Ogni microservizio può produrre log in una posizione diversa e passare da uno all'altro durante il debug potrebbe richiedere molto tempo.

Inoltre, sapere quali microservizi fanno parte di un'applicazione e trovare colli di bottiglia o punti deboli può essere complicato, soprattutto con applicazioni di grandi dimensioni e scarsamente documentate.

AWS X-Ray risolve entrambi i problemi fornendo un punto centralizzato da cui accedere ai log delle applicazioni e consultare una mappa di tutti i microservizi, le risorse e il modo in cui questi comunicano.

## Cos'è AWS X-Ray

AWS X-Ray è un servizio che consente di raccogliere dati dalle applicazioni che si desidera monitorare, in modo da poterli analizzare per trovare possibili problemi o modi per migliorare le prestazioni delle applicazioni. L'invio di dati riferiti a richieste in entrata e in uscita e altri eventi consente di vedere quali risorse AWS, microservizi e API Web vengono utilizzati.

Questo processo è chiamato "Strumentazione".

Esistono diversi tipi di strumentazione:

- **Strumentazione automatica:** viene eseguita tramite configurazioni e non richiede modifiche al codice.
- **Strumentazione della libreria:** richiede modifiche minime al codice dell'applicazione tramite librerie o framework specifici, come l'SDK AWS.
- **Strumentazione manuale:** aggiungi il codice di strumentazione all'applicazione a cui desideri inviare le informazioni di traccia.

Molti servizi AWS forniscono integrazione con AWS X-Ray, come AWS Lambda, Amazon API Gateway, Amazon Simple Notification Service o Amazon Simple Queue Service.

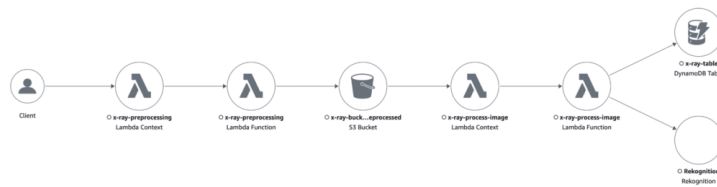
L'integrazione dei servizi AWS con AWS X-Ray può essere suddivisa in:

- **Strumentazione attiva:** richieste in arrivo di campioni e strumenti
- **Strumentazione passiva:** gli strumenti richiedono che un altro servizio abbia campionato
- **Tracciamento delle richieste:** aggiunge un'intestazione di tracciamento a tutte le richieste in entrata
- **Strumenti:** riceve segmenti dall'SDK X-Ray che esegue il demone X-Ray

## Basi di AWS X-Ray

I dati inviati dalle applicazioni sono chiamati **segmenti**. I segmenti contengono informazioni diverse in base alla natura della richiesta. Ad esempio, una richiesta HTTP conterrà informazioni come l'URL, il metodo e il codice di risposta.

Un segmento può essere suddiviso in **sottosegmenti** per fornire maggiori informazioni su tutti gli attori coinvolti nel lavoro. Il segmento dell'applicazione può essere suddiviso in diversi sottosegmenti per ciascuna funzione eseguita per conoscere l'ordine e la durata delle funzioni. I sottosegmenti vengono utilizzati anche da AWS X-Ray per creare "segmenti dedotti" per servizi che non supportano AWS X-Ray, come Amazon DynamoDB. Ciò consente a AWS X-Ray di mostrarli nella **trace map** (o mappa di tracciamento). La mappa di tracciamento mostra tutti i servizi coinvolti e può essere utilizzata per analizzare il flusso di lavoro dell'applicazione. La mappa delle tracce viene creata utilizzando le **tracce**. Una traccia è una raccolta di tutti i segmenti generati da una singola richiesta all'applicazione.



AWS X-Ray non traccia ogni richiesta, ma segue una frequenza di campionamento. La frequenza di campionamento può essere personalizzata nelle impostazioni di AWS X-Ray creando regole da applicare a specifici servizi.

**Rule information**

**Name**  
Name the rule. You can't change the name after you create it.

**Priority**  
Set a rule priority of 1-9999. Rules are applied in order of priority, from lowest to highest.

---

**Limits**  
Choose a reservoir size that sets the limit for sampling requests that match the rule. You can also set a fixed rate of the remaining requests to sample after the reservoir size reaches the limit.

**Reservoir size**  
Set the maximum number of requests to sample per second.

**Fixed rate**  
Set the rate of sampling to allow after exceeding the reservoir size.

---

**Matching Criteria** info  
Choose the criteria of the front-facing services that determines what requests to match. Values can include a multi-character match wildcard (\*) or a single-character match wildcard (?).

**Service name**  
Name of the service to match. For example: billing.

**Service type**  
Type of service to match. For example: AWS-EC2:instance.

**HTTP method**

Se l'applicazione genera molte tracce, è possibile filtrarle per ottenere informazioni più accurate utilizzando le **filter expression**. Le filter expression consentono di filtrare informazioni come tracce che hanno generato errori in servizi specifici, tracce originate da determinati URL o tracce la cui durata ha superato un determinato periodo di tempo.

È anche possibile estendere il filtraggio creando un **gruppo**. I gruppi conterranno tutte le tracce che rispettano l'espressione del filtro generata dopo la creazione del gruppo. Nei gruppi è possibile attivare **insight** e notifiche di insight. AWS X-Ray identifica i problemi emergenti nelle applicazioni e crea insight quando gli intervalli vengono superati. Se le notifiche sono abilitate, riceverai una notifica per ogni evento di insight.

X-Ray tiene traccia degli errori che si verificano nell'applicazione e degli errori restituiti dai servizi richiamati. Gli errori sono raggruppati come segue:

- Errore: errori HTTP 4xx
- Guasto: errori HTTP 5xx
- Throttle - errore HTTP 429

AWS X-Ray crittografa le tracce e i dati inattivi con una chiave di crittografia gestita da AWS.

Se si desidera disattivare la crittografia o utilizzare una chiave gestita da noi a causa dei requisiti di conformità, è possibile configurare AWS X-Ray per utilizzare una chiave KMS.

## Prezzi

in AWS X-Ray si paga in base al numero di tracce registrate, recuperate e scansionate.

Il servizio fornisce una dimensione di traccia minima garantita di 100 KB.

I dati di tracciamento vengono conservati per 30 giorni da quando vengono registrati senza costi aggiuntivi.

AWS X-Ray ha un tier gratuito:

- Le prime 100.000 tracce registrate ogni mese sono gratuite.
- I primi 1.000.000 di tracce recuperate o scansionate ogni mese sono gratuite.

Puoi personalizzare la frequenza di campionamento per scegliere il giusto compromesso tra costi e quantità di dati da raccogliere.

## Applicazione di esempio

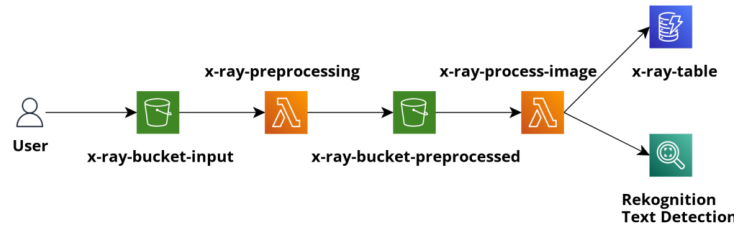
Abbiamo creato una semplice applicazione di esempio per mostrarvi la configurazione della strumentazione a raggi X e i suoi output.

Lo scopo dell'applicazione consiste nel pre-elaborare le immagini, inviarle a Rekognition per rilevare il testo al loro interno e salvare l'output in una tabella DynamoDB per la successiva analisi dei risultati.

L'applicazione comprenderà un bucket S3 in cui caricheremo le immagini che vogliamo analizzare.

La creazione di oggetti all'interno del bucket attiverà una funzione Lambda, che si occuperà di applicare alcune trasformazioni di pre-elaborazione all'immagine.

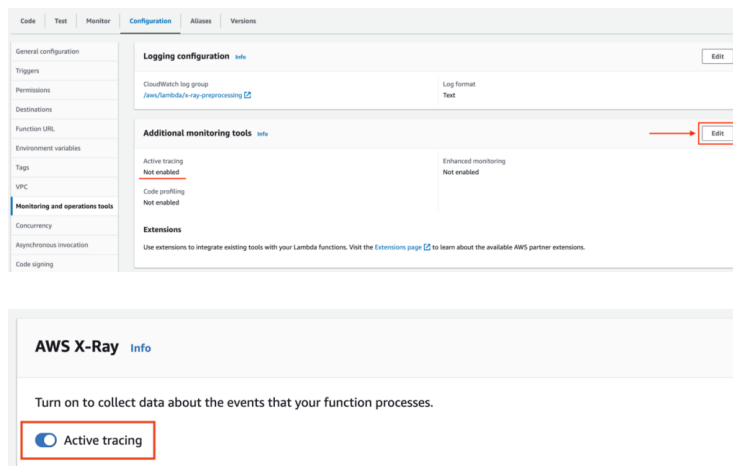
L'output della funzione Lambda di pre-elaborazione verrà salvato in un altro bucket S3, che attiverà un'altra funzione Lambda per inviare l'immagine a Rekognition e salvare i risultati della funzionalità "Text detection" in una tabella DynamoDB.



Innanzitutto, per raccogliere dati dalla nostra applicazione, dobbiamo attivare il tracciamento nelle sue funzioni Lambda.

Si può fare:

- Dalla console AWS:



- Attraverso la CLI di AWS:

```
aws lambda update-function-configuration --function-name my-function --tracing-config Mode=Active
```

- Come configurazione via IaC:
  - CloudFormation



- CDK

```
tracing: Tracing.ACTIVE
  ACTIVE Tracing.ACTIVE
  DISABLED Tracing.DISABLED
  PASS_THROUGH Tracing.PASS_THROUGH
  [Symbol.hasInstance] (value: any) boolean
Press ↵ to insert, ⇨ to replace
```

Ora che il tracciamento della funzione Lambda è abilitato, dobbiamo configurare AWS X-Ray nel nostro codice.

Un modo per farlo è tramite la strumentazione manuale grazie all'SDK di AWS X-Ray.

Le nostre funzioni Lambda vengono eseguite su Node.js e utilizzano SDK v3. Con l'SDK AWS v2, puoi attivare l'acquisizione di tutte le richieste in uscita effettuate dalla libreria. Poiché l'SDK AWS V3 ha una struttura modulare, è necessario acquisire ogni client utilizzato dall'applicazione. In questo modo, ogni volta che i client invieranno un comando, verrà estratto un sottosegmento dal segmento di invocazione Lambda, dandoci una visione più precisa di cosa è successo durante l'esecuzione di quel comando e quanto tempo è stato necessario per completarlo.

Puoi fare la stessa cosa con funzioni e metodi all'interno del tuo codice per sapere quanto è durata ogni invocazione rispetto alla durata totale dell'esecuzione della funzione Lambda.

```
import * as AWSXRay from "aws-xray-sdk-core";
import {S3} from "@aws-sdk/client-s3";

const s3 : S3 = AWSXRay.captureAWSv3Client(new S3( {configuration: {
  region: "eu-west-1"
}}));

/* usage new */
const handler = async function (event) : Promise<void> {
  console.log("Received event: ", JSON.stringify(event));

  await AWSXRay.captureAsyncFunc(
    {
      name: "adjustContrast",
      fn: async function (subsegment : AWSXRay.Subsegment) : Promise<void> {
        await adjustContrast();
        subsegment.close();
      },
    },
    AWSXRay.getSegment()
  );

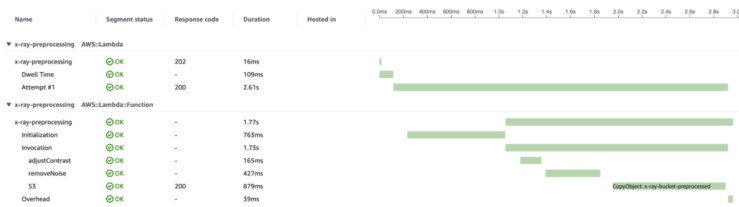
  await AWSXRay.captureAsyncFunc(
    {
      name: "removeNoise",
      fn: async function (subsegment : AWSXRay.Subsegment) : Promise<void> {
        await removeNoise();
        subsegment.close();
      },
    },
    AWSXRay.getSegment()
  );

  await s3.copyObject( {
    Bucket: "x-ray-bucket-preprocessed",
    Key: event.Records[0].s3.object.key,
    CopySource: event.Records[0].s3.bucket.name + "/" + event.Records[0].s3.object.key
  });
};
```

Nel caso della nostra funzione Lambda di pre-elaborazione, prevediamo che il segmento di invocazione sia suddiviso in tre sottosegmenti:

- Sottosegmento **adjustmentConstrast**
- Sottosegmento **removeNoise**
- Sottosegmento **copyObject S3**






Di seguito è riportato il risultato di un'invocazione della funzione:



Caricando alcune immagini nel bucket S3 di input, avremo una panoramica di come l'applicazione gestisce il carico di lavoro delle richieste.



▼ Legend and options

 Fault (5xx)	 Error (4xx)
 Throttle	 Current alarms
 Insights	
Service: <input checked="" type="radio"/>	Service Name
Resource: <input type="radio"/>	Resource Name
Unknown node: <input type="radio"/>	Node Name

---


Show on nodes

Metrics	<b>Icons</b>
---------	--------------

---

Settings

|| Size is requests

 Canaries with faults or errors

La trace map mostra che la funzione Lambda **x-ray-process-image** risponde con alcuni errori 4xx causati dalla limitazione della tabella DynamoDB.

Cliccando sull'icona di una risorsa si apre un menu laterale con tutti i tipi di informazioni sulla risorsa di interesse. Abbiamo potuto vedere rapidamente cosa stava causando l'errore facendo clic sulla scheda "Eccezioni", che mostrava il seguente errore: *"The level of configured provisioned throughput for the table was exceeded. Consider increasing your provisioning level with the UpdateTable API"*.

Ho modificato la DynamoDB dalla modalità "provisioned", con poche capacity units, alla modalità "on-demand".

Con la nuova configurazione, l'applicazione ha sostenuto il traffico senza errori.





## Conclusioni

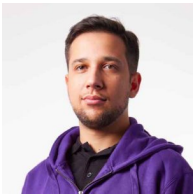
In un mondo in cui le applicazioni di microservizi crescono in numero e complessità, AWS X-Ray fornisce potenti strumenti per il debug e l'analisi. Con questo articolo volevamo creare una piccola guida sui suoi fondamenti e su come abilitarlo nelle funzioni AWS Lambda.

La nostra applicazione di esempio utilizzava la strumentazione manuale poiché era piccola e scritta per la guida, ma nel caso di applicazioni significative, la strumentazione automatica è un'opzione più rapida e semplice.

Avete già utilizzato AWS X-Ray per il monitoring delle vostre applicazioni? Fateci sapere se volete saperne di più!

## 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!



### Daniele Papa

DevOps Engineer e backend developer @ beSharp. Nel tempo libero amo giocare a videogiochi e giochi da tavolo. Negli ultimi anni mi sono avvicinato al mondo Cloud, ora passo da ruoli IAM... e giochi di ruolo.