Whitepaper AWS

Panoramica della sicurezza di AWS Lambda

Panoramica della sicurezza di AWS Lambda: Whitepaper AWS

Table of Contents

Panoramica della sicurezza di AWS Lambda

Data di pubblicazione: 12 febbraio 2021 (Revisioni del documento)

Riassunto

Questo whitepaper presenta una descrizione dettagliata del servizio AWS Lambda dal punto di vista della sicurezza. Fornisce un quadro completo del servizio, che è utile per i nuovi utenti e permette agli utenti attuali di approfondire la comprensione di Lambda.

Il destinatari principali di questo whitepaper sono i Chief Information Security Officer (CISO), gli ingegneri addetti alla sicurezza delle informazioni, gli architetti aziendali, i team di conformità e qualsiasi altra parte interessata a comprendere le basi di AWS Lambda.

Introduzione

Oggi, sempre più carichi di lavoro utilizzano AWS Lambda per ottenere scalabilità, prestazioni ed efficienza in termini di costi, senza la necessità di gestire l'infrastruttura sottostante. Questi carichi di lavoro sono scalabili fino a migliaia di richieste al secondo simultanee. Lambda è uno dei tanti importanti servizi offerti oggi da AWS. Lambda viene utilizzato da centinaia di migliaia di clienti Amazon Web Services (AWS) per soddisfare migliaia di miliardi di richieste ogni mese.

Lambda è adatto per applicazioni essenziali per l'organizzazione in molti settori. Una vasta gamma di clienti trae vantaggio da Lambda, da quelli del settore media e intrattenimento a quelli dei servizi finanziari e di altri settori regolamentati. Questi clienti riducono il time-to-market, ottimizzano i costi e migliorano l'agilità concentrandosi su ciò che sanno fare meglio: gestire la propria attività.

Il modello di ambiente di runtimegestito consente a Lambda di occuparsi di gran parte dei dettagli di implementazione dell'esecuzione di carichi di lavoro serverless. Questo modello riduce ulteriormente la superficie di attacco semplificando la sicurezza del cloud. Questo whitepaper presenta le basi di tale modello, insieme alle relative best practice, agli sviluppatori, agli analisti della sicurezza, ai team di sicurezza e conformità e ad altre parti interessate.

Informazioni su AWS Lambda

AWS Lambda è un servizio di calcolo serverless basato su eventi che estende altri servizi AWS con una logica personalizzata o crea ulteriori servizi di back-end che operano con scalabilità, prestazioni e sicurezza. Lambda può eseguire automaticamente del codice in risposta a eventi multipli, come richieste HTTP tramite Amazon API Gateway, modifiche agli oggetti contenuti in bucket Amazon S3, aggiornamenti di tabelle in Amazon DynamoDB e transizioni di stato in AWS Step Functions. Può anche eseguire il codice direttamente da qualsiasi app web o per dispositivi mobili. Lambda esegue il codice su un'infrastruttura di calcolo a elevata disponibilità e si occupa delle operazioni di amministrazione della piattaforma sottostante, incluse la manutenzione del server e del sistema operativo, il provisioning e la scalabilità automatica della capacità, l'applicazione delle patch, il monitoraggio del codice e la creazione di log.

Con Lambda, basta semplicemente caricare il codice e configurare il meccanismo di invocazione. Lambda si occupa di tutto quanto necessario per eseguire il codice con elevata disponibilità. Lambda si integra con molti altri servizi AWS e consente di creare applicazioni serverless o servizi back-end, che vanno da semplici attività di automazione attivate periodicamente ad applicazioni di microservizi complete.

Lambda può anche essere configurato per accedere alle risorse all'interno del proprio Amazon Virtual Private Cloud e, per estensione, alle risorse on-premise.

È possibile inglobare Lambda all'interno di un solido livello di sicurezza, utilizzando AWS Identity and Access Management (IAM) e altre tecniche discusse in questo whitepaper, per mantenere un elevato livello di sicurezza e controllo e soddisfare ogni esigenza di conformità.

Argomenti

Vantaggi di Lambda

I clienti che desiderano liberare la creatività e la velocità delle loro organizzazioni di sviluppo, senza compromettere la capacità del team IT di fornire un'infrastruttura scalabile, conveniente e gestibile, trovano che AWS Lambda consente loro di convertire la complessità operativa in agilità e prezzi migliori, senza compromettere le possibilità di ridimensionamento o l'affidabilità.

Lambda offre molti vantaggi, tra cui i seguenti:

Nessun server da gestire.

Lambda esegue il codice su un'infrastruttura a elevata disponibilità e tolleranza ai guasti distribuita su più Zone di disponibilità (AZ) in una singola regione, implementando il codice senza soluzione di continuità e occupandosi di ogni aspetto relativo ad amministrazione, manutenzione e patch dell'infrastruttura. Lambda fornisce inoltre funzionalità di logging e monitoraggio integrate, inclusa l'integrazione con Amazon CloudWatch, CloudWatch Logs e AWS CloudTrail.

Ridimensionamento continuo

Lambda gestisce con precisione il ridimensionamento delle funzioni (o dell'applicazione) eseguendo codice attivato da eventi in parallelo ed elaborando ogni evento singolarmente.

Tariffazione in millisecondi

Con AWS Lambda, l'addebito viene calcolato in base a ogni 1 millisecondo (ms) di esecuzione del codice e al numero di volte in cui il codice viene attivato. Il pagamento avviene per throughput costante o durata di esecuzione piuttosto che per unità di server.

Incremento dell'innovazione

Lambda libera le risorse di programmazione occupandosi della gestione dell'infrastruttura e consentendo loro di concentrarsi maggiormente sull'innovazione e sullo sviluppo della logica di business.

Modernizzazione delle applicazioni

Lambda consente di utilizzare funzioni con modelli di machine learning pre-addestrati per iniettare facilmente tecniche di intelligenza artificiale nelle applicazioni. Una singola richiesta API (Application Programming Interface) può eseguire classificazione di immagini, analisi di video, riconoscimento vocale, elaborazione del linguaggio naturale e altro ancora.

Ricco ecosistema

Lambda supporta gli sviluppatori tramite AWS Serverless Application Repository, per l'individuazione, l'implementazione e la pubblicazione di applicazioni serverless, AWS Serverless Application Model, per la creazione di applicazioni serverless, e integrazioni con vari ambienti integrati di sviluppo (IDE)

come AWS Cloud9, AWS Toolkit for Visual Studio, AWS Tools for Visual Studio Team Services e molti altri. Lambda è integrato con ulteriori servizi AWS per fornire un ricco ecosistema per la creazione di applicazioni serverless.

Costo per l'esecuzione di applicazioni basate su Lambda

Lambda offre un modello granulare di pagamento in base al consumo. Con questo modello, il costo addebitato è calcolato in base al numero di chiamate di funzioni e alla loro durata (il tempo necessario per l'esecuzione del codice). Oltre a questo modello di prezzo flessibile, Lambda offre anche 1 milione di richieste gratuite al mese a tempo indefinito, che consentono a molti clienti di automatizzare il proprio processo senza alcun costo.

Modello di responsabilità condivisa

La sicurezza e la conformità sono una responsabilità condivisa tra AWS e il cliente. Questo modello di responsabilità condivisa può contribuire a ridurre l'onere operativo, dato che AWS rende operativi, gestisce e controlla tutti i componenti, dal sistema operativo host e il layer di virtualizzazione fino alla sicurezza fisica delle strutture in cui operano i servizi.

Per AWS Lambda, AWS gestisce l'infrastruttura sottostante e i servizi di base, il sistema operativo e la piattaforma applicativa. L'utente è responsabile della sicurezza del suo codice e della gestione delle identità e degli accessi (IAM) al servizio Lambda e all'interno della sua funzione.

La Figura 1 mostra il modello di responsabilità condivisa applicato alle componenti comuni e specifiche di AWS Lambda. Le responsabilità di AWS appaiono sotto la linea tratteggiata in arancione, mentre le responsabilità degli utenti appaiono sopra la linea tratteggiata in blu.

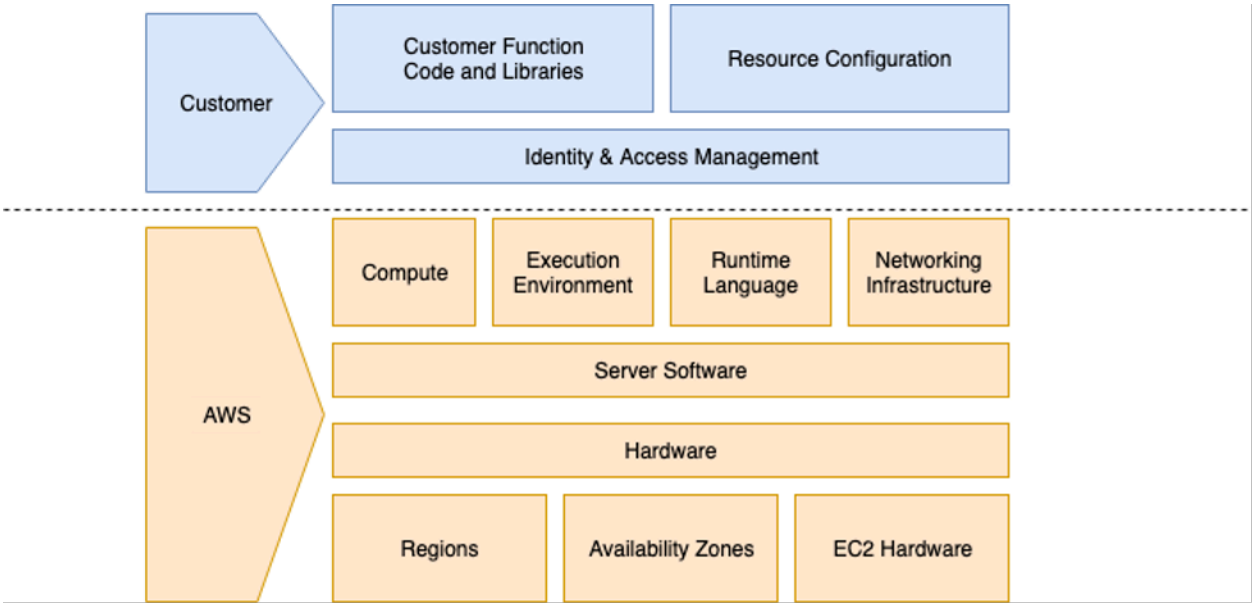

Figura 1: modello di responsabilità condivisa per AWS Lambda

Funzioni e livelli Lambda

Con Lambda, è possibile eseguire codice virtualmente senza alcuna necessità di amministrazione dell'infrastruttura sottostante. L'utente è responsabile del solo codice fornito a Lambda e della configurazione della modalità in base alla quale Lambda esegue tale codice per suo conto. Oggi, Lambda supporta due tipi di risorse di codice: Funzioni e Livelli.

Una funzione è una risorsa che può essere richiamata per eseguire il codice in Lambda. Le funzioni possono includere risorse comuni o condivise chiamate Livelli. I livelli possono essere utilizzati per condividere codice o dati comuni tra diverse funzioni o account AWS. L'utente è responsabile della gestione di tutto il codice contenuto nelle funzioni o nei livelli. Quando Lambda riceve il codice di funzione o livello da un utente, il servizio protegge l'accesso ad esso tramite crittografia dei dati a riposo utilizzando AWS Key Management Service (AWS KMS) e crittografia dei dati in transito utilizzando TLS 1.2+.

È possibile gestire l'accesso alle funzioni e ai livelli tramite policy AWS Lambda o autorizzazioni basate sulle risorse. Per un elenco completo delle funzionalità IAM supportate su IAM, consultare Servizi AWS che funzionano con IAM.

È inoltre possibile controllare l'intero ciclo di vita delle funzioni e dei livelli tramite le API del piano di controllo di Lambda. Ad esempio, è possibile scegliere di eliminare la tua funzione invocando `DeleteFunction` o di revocare le autorizzazioni da un altro account invocando `RemovePermission`.

Modalità di invocazione Lambda

L'API Invoke può essere invocata in due modalità: modalità evento e modalità richiesta-risposta.

- La modalità evento aggiunge il payload a una coda per una chiamata asincrona.
- La modalità richiesta-risposta invoca in modo sincrono la funzione con il payload fornito e restituisce immediatamente una risposta.

In entrambi i casi, l'esecuzione della funzione viene sempre eseguita in un ambiente di esecuzione Lambda, ma il payload segue percorsi diversi. Per ulteriori informazioni, consultare "Ambienti di esecuzione Lambda" in questo documento.

È anche possibile utilizzare altri servizi AWS che eseguono invocazioni per conto dell'utente. La modalità di invocazione utilizzata dipende dal servizio AWS in uso e da come è configurato. Per ulteriori informazioni su come altri servizi AWS si integrano con Lambda, consultare Utilizzo di AWS Lambda con altri servizi.

Quando Lambda riceve un'invocazione di tipo richiesta-risposta, il payload viene passato direttamente al servizio Invoke. Se il servizio Invoke non è disponibile, i chiamanti possono mettere temporaneamente in coda il payload lato client per ritentare la chiamata un determinato numero di volte. Se il servizio Invoke riceve il payload, questo tenta di identificare un ambiente di esecuzione disponibile per la richiesta e passa il payload a tale ambiente di esecuzione per completare la chiamata. Se non esistono ambienti di esecuzione esistenti o appropriati, ne verrà creato uno dinamicamente in risposta alla richiesta. Durante il transito, i payload di invocazione inviati al servizio Invoke sono protetti con TLS 1.2+. Il traffico all'interno del servizio Lambda (dal bilanciatore del carico in poi) passa attraverso un Virtual Private Cloud (VPC) interno isolato, di proprietà del servizio Lambda, all'interno della regione AWS a cui è stata inviata la richiesta.

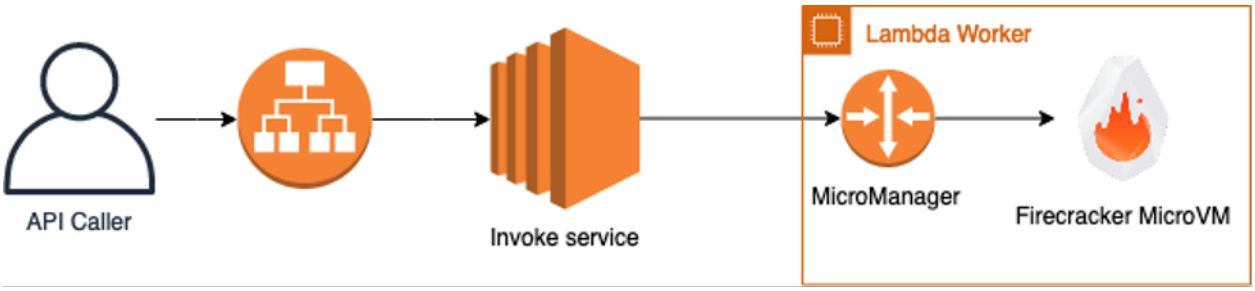

Figura 2: Modello di invocazione per la modalità richiesta-risposta di AWS Lambda

I payload della modalità di invocazione evento vengono sempre aggiunti a una coda per l'elaborazione prima della chiamata. Tutti i payload destinati all'elaborazione sono aggiunti a una coda Amazon Simple Queue Service (Amazon SQS). Gli eventi in coda sono sempre protetti in transito con TLS 1.2+, ma al momento non sono soggetti a crittografia dei dati a riposo. Le code Amazon SQS utilizzate da Lambda sono gestite dal servizio Lambda e non sono visibili all'utente. Gli eventi in coda possono essere archiviati in una coda condivisa, ma possono essere migrati o assegnati a code dedicate in base a una serie di fattori che non possono essere controllati direttamente dagli utenti (ad esempio, ritmo di invocazione, dimensione degli eventi e così via).

Gli eventi in coda vengono recuperati in batch dal parco istanze di poller di Lambda. Il parco istanze di poller è un gruppo di istanze EC2 il cui scopo è quello di elaborare le chiamate di eventi in coda che non sono ancora state elaborate. Quando il parco istanze di poller recupera un evento in coda che deve essere elaborato, lo fa passandolo al servizio Invoke proprio come farebbe un utente che utilizza la modalità richiesta-risposta.

Se l'invocazione non può essere eseguita, il parco istanze di poller memorizzerà temporaneamente l'evento in memoria sull'host finché non sarà in grado di completare correttamente l'esecuzione o finché non sarà stato superato il numero di tentativi di esecuzione. Nessun dato di payload viene mai scritto su disco nel parco istanze di poller stesso. Le attività del parco istanze di polling possono essere condivise da tutti i clienti AWS, garantendo il minor tempo possibile per l'invocazione. Per ulteriori informazioni su quali servizi possono utilizzare la modalità di invocazione evento, consultare Utilizzo di AWS Lambda con altri servizi.

Esecuzioni Lambda

Quando Lambda esegue una funzione per conto dell'utente, gestisce sia il provisioning che la configurazione dei sistemi sottostanti necessari per eseguire il codice. Ciò consente agli sviluppatori di concentrarsi sulla logica di business e sulla scrittura di codice, non sull'amministrazione e sulla gestione dei sistemi sottostanti.

Il servizio Lambda è suddiviso nel piano di controllo e nel piano dati. Ogni piano ricopre un ruolo distinto nel servizio. Il piano di controllo fornisce le API di gestione (ad esempio `CreateFunction`, `UpdateFunctionCode`, `PublishLayerVersion` e così via) e gestisce le integrazioni con tutti i servizi AWS. Le comunicazioni con il piano di controllo di Lambda sono protette in transito tramite TLS. Tutti i dati degli utenti memorizzati nel piano di controllo di Lambda sono protetti dalla crittografia dei dati a riposo attraverso l'uso di AWS KMS, progettato per proteggerli da divulgazioni non autorizzate o manomissioni.

Il piano dati è l'API Invoke di Lambda che attiva l'invocazione delle funzioni Lambda. Quando viene richiamata una funzione Lambda, il piano dati alloca un ambiente di esecuzionesu un AWS Lambda Worker (o semplicemente Worker, un tipo di istanza Amazon EC2) a quella versione della funzione oppure sceglie un ambiente di esecuzione esistente che è già stato configurato per quella versione di funzione, che poi viene utilizzato per completare l'invocazione. Per ulteriori informazioni, consultare la sezione "MicroVM e Worker di AWS Lambda" di questo documento.

Ambienti di esecuzione Lambda

Ogni chiamata viene instradata dal servizio Invoke di Lambda a un ambiente di esecuzione su un Worker in grado di soddisfare la richiesta. Salvo che attraverso il piano dati, i clienti e gli altri utenti non possono avviare direttamente comunicazioni di rete in ingresso verso un ambiente di esecuzione. Questo aiuta a garantire che le comunicazioni con l'ambiente di esecuzione siano autenticate e autorizzate.

Gli ambienti di esecuzione sono riservati per una versione di funzione specifica e non possono essere riutilizzati tra versioni di funzioni, funzioni o account AWS. Ciò significa che una singola funzione che presenta due versioni diverse comporta la presenza di almeno due ambienti di esecuzione distinti.

Ogni ambiente di esecuzione può essere utilizzato solo per una chiamata simultanea alla volta e può essere riutilizzato su più invocazioni della stessa versione di funzione per motivi prestazionali. A seconda di una serie di fattori (ad esempio, ritmo delle invocazioni, configurazione delle funzioni e

così via), possono esistere uno o più ambienti di esecuzione per una data versione di funzione. Con questo approccio, Lambda è in grado di garantire agli utenti l'isolamento a livello di versione delle funzioni.

Lambda attualmente non isola le invocazioni all'interno dell'ambiente di esecuzione di una versione di funzione. Ciò significa che una chiamata può concludersi lasciando uno stato che può influenzare la successiva invocazione (ad esempio, file scritti in /tmp o dati in memoria). Per assicurarsi che un'invocazione non possa influire su un'altra successiva, Lambda consiglia di creare funzioni distinte aggiuntive. Ad esempio, è possibile creare funzioni distinte per operazioni di analisi complesse che sono maggiormente soggette a errori e riutilizzare funzioni che non eseguono operazioni sensibili dal punto di vista della sicurezza. Lambda attualmente non limita il numero di funzioni che gli utenti possono creare. Per ulteriori informazioni sui limiti, consultare la pagina relativa alle quote di Lambda.

Gli ambienti di esecuzione sono costantemente monitorati e gestiti da Lambda e possono essere creati o distrutti per una serie di motivi, tra cui, a titolo esemplificativo ma non esaustivo:

- Arrivo di una nuova invocazione senza la disponibilità di un ambiente di esecuzione adatto
- Occorrenza di un'implementazione interna del runtime o del software di un Worker
- Pubblicazione di una nuova configurazione di concorrenza con provisioning
- Avvicinamento o superamento della durata massima del tempo di lease dell'ambiente di esecuzione o del Worker
- Altri processi interni di ribilanciamento del carico di lavoro

I clienti possono gestire il numero di ambienti di esecuzione con pre-provisioning esistenti per una versione di funzione configurando la concorrenza con provisioning nella configurazione delle funzioni. Se configurato per farlo, Lambda creerà, gestirà e assicurerà che il numero configurato di ambienti di esecuzione sia sempre disponibile. Ciò garantisce ai clienti un maggiore controllo sulle prestazioni di avvio delle loro applicazioni serverless su qualsiasi scala.

A parte la configurazione della concorrenza con provisioning, i clienti non possono controllare in modo deterministico il numero di ambienti di esecuzione creati o gestiti da Lambda in risposta alle chiamate.

Ruolo di esecuzione

Ogni funzione Lambda deve inoltre essere configurata con un ruolo di esecuzione, che è un ruolo IAM che viene assunto dal servizio Lambda durante l'esecuzione di operazioni relative al piano di

controllo e al piano dati relative alla funzione. Il servizio Lambda assume questo ruolo per recuperare credenziali di sicurezza temporanee che sono poi disponibili come variabili di ambiente durante l'invocazione di una funzione. Per motivi prestazionali, il servizio Lambda memorizzerà nella cache tali credenziali e potrà riutilizzarle in diversi ambienti di esecuzione che utilizzano lo stesso ruolo di esecuzione.

Per garantire il rispetto del principio del privilegio minimo, Lambda raccomanda che ogni funzione disponga del proprio ruolo unico e che sia configurata con il set minimo di autorizzazioni richieste.

Il servizio Lambda può anche assumere il ruolo di esecuzione per eseguire specifiche operazioni del piano di controllo come quelle relative alla creazione e alla configurazione di interfacce di rete elastiche (ENI) per funzioni VPC, invio di log ad Amazon CloudWatch Application Insights, invio di tracciamenti a AWS X-Ray o altre operazioni correlate differenti dalle invocazioni. Gli utenti possono sempre esaminare ed eseguire l'audit di tali casi d'uso esaminando i log di audit in AWS CloudTrail.

Per ulteriori informazioni su questo argomento, consultare la pagina della documentazione sul ruolo di esecuzione di AWS Lambda.

Lambda MicroVM e Worker

Lambda creerà i suoi ambienti di esecuzione su un parco istanze Amazon EC2 chiamate AWS Lambda Worker. I Worker sono istanze Nitro EC2 bare metal che vengono avviate e gestite da Lambda in un account AWS isolato separato che non è visibile agli utenti. I Worker dispongono di una o più Micro Virtual Machines (MVM) virtualizzate tramite hardware create da Firecracker. Firecracker è un Virtual Machine Monitor (VMM) open source che utilizza la macchina virtuale basata sul kernel Linux (KVM) per creare e gestire MVM. È progettato appositamente per la creazione e la gestione di container e servizi basati su funzioni sicuri e multi-tenant che forniscono modelli operativi serverless. Per ulteriori informazioni sul modello di sicurezza di Firecracker, consultare il sito web del progetto Firecracker.

Come parte del modello di responsabilità condivisa, Lambda è responsabile del mantenimento della configurazione di sicurezza, dei controlli e del livello di applicazione delle patch dei Worker. Il team Lambda utilizza Amazon Inspector per scoprire potenziali problemi di sicurezza noti, nonché altri meccanismi di notifica dei problemi di sicurezza personalizzati ed elenchi di pre-divulgazione, in modo che gli utenti non debbano gestire il livello di sicurezza sottostante del loro ambiente di esecuzione.

Figura 3: Modello di isolamento per i Worker di AWS Lambda

I Worker sono soggetti a una durata massima del tempo di lease di 14 ore. Quando un Worker si avvicina al tempo massimo di lease, non vengono instradate verso di esso ulteriori invocazioni, le MVM vengono regolarmente arrestate e l'istanza Worker sottostante viene terminata. Lambda monitora e genera allarmi in modo continuo relativamente alle attività del ciclo di vita del suo parco istanze.

Tutte le comunicazioni del piano dati con i Worker sono crittografate utilizzando Advanced Encryption Standard with Galois/Counter Mode (AES-GCM). Oltre alle operazioni sul piano dati, i clienti non possono interagire direttamente con un Worker in quanto è ospitato in un Amazon VPC isolato in rete gestito da Lambda negli account di servizio di Lambda.

Quando un Worker deve creare un nuovo ambiente di esecuzione, gli viene concessa un'autorizzazione limitata nel tempo per accedere agli artefatti delle funzioni del cliente. Questi artefatti sono ottimizzati specificamente per l'ambiente di esecuzione e i Worker di Lambda. Il codice funzione che viene caricato utilizzando il formato ZIP viene ottimizzato una sola volta e quindi viene archiviato in un formato crittografato utilizzando una chiave gestita da AWS e AES-GCM.

Anche le funzioni caricate su Lambda utilizzando il formato immagine container sono ottimizzate. L'immagine del container viene prima scaricata dalla fonte originale, ottimizzata in blocchi distinti e quindi archiviata sotto forma di blocchi crittografati utilizzando un metodo di crittografia convergente autenticato che utilizza una combinazione di AES-CTR, AES-GCM e MAC SHA-256. Il metodo di crittografia convergente consente a Lambda di deduplicare in modo sicuro i blocchi crittografati. Tutte le chiavi necessarie per decrittare i dati degli utenti sono protette tramite Customer Master Key AWS KMS (CMK) gestite dal cliente. L'utilizzo di CMK da parte del servizio Lambda è disponibile per gli utenti nei log di AWS CloudTrail a fini di monitoraggio e auditing.

Tecnologie di isolamento Lambda

Lambda utilizza una varietà di tecnologie di isolamento open source e proprietarie per proteggere i Worker e gli ambienti di esecuzione. Ogni ambiente di esecuzione contiene una copia dedicata dei seguenti elementi:

- Il codice della specifica versione della funzione
- Qualsiasi livello AWS Lambda selezionato per la versione della tua funzione
- Il runtime della funzione scelta (ad esempio, Java 11, NodeJS 12, Python 3.8 e così via) o il runtime personalizzato della funzione
- Una directory /tmp scrivibile
- Uno spazio utente Linux minimo basato su Amazon Linux 2

Gli ambienti di esecuzione sono isolati l'uno dall'altro utilizzando diverse tecnologie simili a container integrate nel kernel Linux, insieme alle tecnologie di isolamento proprietarie di AWS. Queste tecnologie includono:

- cgroups: usati per limitare l'accesso della funzione alla CPU e alla memoria.
- namespaces: ogni ambiente di esecuzione viene eseguito in uno spazio dei nomi dedicato. Tale obiettivo è raggiunto utilizzando ID di processo di gruppo, ID utente, interfacce di rete e altre risorse univoci gestiti dal kernel Linux.
- seccomp-bpf: per limitare le chiamate di sistema (syscalls) che possono essere utilizzate dall'ambiente di esecuzione.
- iptables e tabelle di routing: per impedire le comunicazioni di rete in ingresso e isolare le connessioni di rete tra MVM.
- chroot: fornisce accesso con un ambito limitato al filesystem sottostante.
- Configurazione di Firecracker: utilizzata per limitare il throughput del dispositivo a blocchi e dei dispositivi di rete.
- Funzionalità di sicurezza di Firecracker: per ulteriori informazioni sull'attuale progetto di sicurezza di Firecracker, consultare il documento di progettazione di Firecracker più recente.

Insieme alle tecnologie di isolamento proprietarie di AWS, questi meccanismi forniscono un forte isolamento tra gli ambienti di esecuzione.

Archiviazione e stato

Gli ambienti di esecuzione non vengono mai riutilizzati tra diverse versioni di funzioni o clienti, ma un singolo ambiente può essere riutilizzato tra invocazioni della stessa versione di funzione. Ciò significa che i dati e lo stato possono persistere tra le invocazioni. I dati e/o lo stato possono continuare a persistere per ore prima di essere distrutti come parte della normale gestione del ciclo di vita dell'ambiente di esecuzione. Per motivi prestazionali, le funzioni possono trarre vantaggio da questo comportamento per migliorare l'efficienza mantenendo e riutilizzando cache locali o connessioni di lunga durata tra le chiamate. All'interno di un ambiente di esecuzione, queste invocazioni multiple sono gestite da un singolo processo quindi, se la chiamata si verifica in un ambiente di esecuzione riutilizzato, qualsiasi stato a livello di processo (come uno stato statico in Java) può essere disponibile per il riutilizzo nelle successive invocazioni.

Ogni ambiente di esecuzione Lambda include anche un filesystem scrivibile, disponibile a `/tmp`. Questo spazio di archiviazione non è accessibile o condiviso tra gli ambienti di esecuzione. Come per lo stato del processo, i file scritti su `/tmp` rimangono persistenti per tutta la vita dell'ambiente di esecuzione. Ciò consente di ammortizzare su più invocazioni costose operazioni di trasferimento, come il download di modelli di machine learning (ML). Le funzioni che non intendono mantenere i dati tra le invocazioni non devono scrivere su /tmp o devono cancellare i loro file da /tmp tra le invocazioni. La directory `/tmp` è supportata da un archivio istanza Amazon EC2 ed è soggetto a crittografia dei dati a riposo.

Gli utenti che desiderano mantenere i dati nel file system al di fuori dell'ambiente di esecuzione dovrebbero prendere in considerazione l'utilizzo dell'integrazione di Lambda con Amazon Elastic File System (Amazon EFS). Per ulteriori informazioni, consultare Utilizzo di Amazon EFS con AWS Lambda.

Se gli utenti non desiderano mantenere i dati o lo stato tra le chiamate, Lambda consiglia di non utilizzare il contesto di esecuzione o l'ambiente di esecuzione per archiviare dati o stato. Se gli utenti desiderano prevenire attivamente la fuga di dati o stato tra le chiamate, Lambda consiglia di creare funzioni distinte per ogni stato. Lambda non consiglia ai clienti di utilizzare o archiviare uno stato sensibile dal punto di vista della sicurezza nell'ambiente di esecuzione, in quanto potrebbe subire variazioni tra le invocazioni. Consigliamo invece di ricalcolare lo stato a ogni invocazione.

Manutenzione del runtime in Lambda

Lambda fornisce supporto per questi runtime analizzando e implementando continuamente aggiornamenti e patch di sicurezza compatibili ed eseguendo altre attività di manutenzione in fase di runtime. Ciò consente agli utenti di concentrarsi solo sulla manutenzione e sulla sicurezza del codice incluso nella loro funzione e livello. Il team Lambda utilizza Amazon Inspector per scoprire problemi di sicurezza noti, nonché altri meccanismi di notifica dei problemi di sicurezza personalizzati ed elenchi di pre-divulgazione per garantire che i linguaggi di runtime e l'ambiente di esecuzione rimangano aggiornati. Se vengono identificate nuove patch o aggiornamenti, Lambda testa e implementa gli aggiornamenti di runtime senza alcun coinvolgimento da parte degli utenti. Per ulteriori informazioni sul programma di conformità di Lambda, consultare la sezione "Lambda e conformità" di questo documento.

In genere, non è richiesta alcuna operazione per l'applicazione delle patch più recenti per i runtime Lambda supportati, ma a volte potrebbe essere necessari un passaggio di test delle patch prima che vengano implementate (ad esempio, patch di runtime con incompatibilità note). Nel caso in cui sia necessaria un'operazione da parte degli utenti, Lambda li contatterà tramite il Personal Health Dashboard, tramite l'e-mail dell'account AWS o tramite altri mezzi, con l'indicazione delle operazioni specifiche da intraprendere.

I clienti possono utilizzare altri linguaggi di programmazione in Lambda implementando un runtime personalizzato. Per i runtime personalizzati, la manutenzione del runtime, inclusa la verifica che il runtime personalizzato includa le patch di sicurezza più recenti, diventa responsabilità dell'utente. Per ulteriori informazioni, consultare Runtime personalizzati di AWS Lambda nella Guida per lo sviluppatore di AWS Lambda.

Quando i manutentori del linguaggio runtime a monte contrassegnano il loro linguaggio come in stato End-Of-Life (EOL), Lambda rispetta la decisione non supportando più il runtime relativo alla specifica versione del linguaggio. Quando le versioni del runtime sono contrassegnate come deprecate in Lambda, Lambda interrompe il supporto a creazione di nuove funzioni e aggiornamenti di funzioni esistenti create nel runtime deprecato. Per avvisare il cliente di imminenti deprecazioni del runtime, Lambda invia notifiche agli utenti indicando l'imminente data di deprecazione e le conseguenze che possono aspettarsi. Lambda non fornirà inoltre aggiornamenti di sicurezza, supporto tecnico o aggiornamenti rapidi per i runtime deprecati e riserva il diritto di disabilitare le invocazioni di funzioni configurate per essere eseguite in un runtime deprecato in qualsiasi momento. Se gli utenti desiderano continuare a eseguire versioni di runtime deprecate o non supportate, possono creare il

proprio runtime AWS Lambda personalizzato. Per dettagli su quando i runtime diventano deprecati, consultare la policy di supporto dei runtime di AWS Lambda.

Monitoraggio e auditing delle funzioni Lambda

È possibile monitorare ed eseguire l'auditing delle funzioni Lambda con molti servizi e metodi AWS, inclusi i seguenti servizi.

Amazon CloudWatch

AWS Lambda monitora automaticamente le funzioni Lambda per conto degli utenti. Tramite Amazon CloudWatch, segnala parametri quali il numero di richieste, la durata dell'esecuzione per richiesta e il numero di richieste che generano un errore. Questi parametri sono esposti a livello di funzione e possono quindi essere sfruttati per impostare gli allarmi di CloudWatch. Per un elenco dei parametri esposti da Lambda, consultare Parametri di AWS Lambda.

Amazon CloudTrail

Utilizzando AWS CloudTrail, è possibile implementare la governance, la conformità, il controllo operativo e il controllo dei rischi dell'intero account AWS, incluso Lambda. CloudTrail consente di registrare, monitorare e mantenere continuamente le attività dell'account relative alle operazioni sull'infrastruttura AWS, fornendo una cronologia completa degli eventi delle operazioni intraprese tramite AWS Management Console, SDK AWS, strumenti a riga di comando e altri servizi AWS. Utilizzando CloudTrail, è possibile facoltativamente crittografare i file di log utilizzando AWS KMS e sfruttare anche la convalida dell'integrità dei file di log di CloudTrail per un'asserzione positiva.

AWS X-Ray

Utilizzando AWS X-Ray, è possibile analizzare ed eseguire il debug di applicazioni di produzione e distribuite basate su Lambda. In questo modo è possibile comprendere le prestazioni dell'applicazione e dei relativi servizi sottostanti, in modo da poter identificare e risolvere la causa principale dei problemi prestazionali e degli errori. La vista end-to-end delle richieste durante il percorso attraverso l'applicazione offerta da X-Ray mostra una mappa dei componenti sottostanti dell'applicazione, in modo da poter analizzare le applicazioni durante lo sviluppo e in produzione.

AWS Config

Con AWS Config, è possibile tenere traccia delle modifiche alla configurazione delle funzioni Lambda (incluse le funzioni eliminate), degli ambienti di runtime, dei tag, del nome del gestore,

della dimensione del codice, dell'allocazione della memoria, delle impostazioni di timeout e delle impostazioni di concorrenza, insieme alle associazioni tra Lambda e ruolo di esecuzione, sottorete e gruppo di sicurezza IAM. Ciò offre una visione olistica del ciclo di vita della funzione Lambda e consente di far emergere tali dati per potenziali requisiti di audit e conformità.

Progettazione e funzionamento delle funzioni Lambda

Questa sezione illustra l'architettura e il funzionamento di Lambda. Per informazioni sulle best practice standard per le applicazioni serverless, consultare il whitepaper Serverless Applications Lens, che definisce ed esplora i pilastri dell'AWS Well-Architected Framework in un contesto serverless.

- Pilastro dell'eccellenza operativa: la capacità di eseguire e monitorare sistemi per offrire valore aziendale e migliorare in modo continuo il supporto di processi e procedure.
- Pilastro della sicurezza: la capacità di proteggere le informazioni, i sistemi e le risorse offrendo valore aziendale attraverso strategie di valutazione e mitigazione dei rischi.
- Pilastro dell'affidabilità: la capacità di un sistema di ripristinarsi in seguito a interruzione nel funzionamento dell'infrastruttura o del servizio, di acquisire in modo dinamico nuove risorse di calcolo per rispondere alla richiesta e di mitigare problemi quali configurazioni errate o problemi di rete transitori.
- Pilastro dell'efficienza delle prestazioni: l'utilizzo efficiente delle risorse di calcolo per soddisfare i requisiti e la manutenzione di tale efficienza in risposta ai cambiamenti della domanda e all'evoluzione delle tecnologie.
- Pilastro dell'ottimizzazione dei costi: il processo continuo di perfezionamento e miglioramento per garantire che i risultati aziendali vengano raggiunti riducendo al minimo i costi man mano che la domanda cambia e le tecnologie si evolvono.

Il whitepaper Serverless Applications Lens include argomenti come logging di parametri e allarmi, throttling e i limitazioni, assegnazione delle autorizzazioni alle funzioni Lambda e l'accesso ai dati sensibili da parte delle funzioni Lambda.

Lambda e conformità

Come indicato nella sezione "Modello di responsabilità condivisa", l'utente è responsabile della determinazione di quale regime di conformità deve essere applicato ai suoi dati. Dopo aver determinato le esigenze del regime di conformità, è possibile utilizzare le varie funzionalità Lambda per soddisfare tali controlli. Per ricevere assistenza è possibile contattare gli esperti di AWS (come Solution Architect, esperti del dominio, Technical Account Manager e altre risorse umane). Tuttavia, AWS non può consigliare ai clienti se o quali regimi di conformità siano applicabili a un particolare caso d'uso.

A partire da novembre 2020, Lambda rientra nell'ambito dei report SOC 1, SOC 2 e SOC 3, che sono report di valutazioni indipendenti di terze parti che dimostrano come AWS rispetti i principali controlli e obiettivi di conformità. Per un elenco aggiornato di informazioni sulla conformità, consultare la pagina Servizi AWS coperti dal programma di compliance.

A causa della natura sensibile di alcuni report di conformità, questi non possono essere condivisi pubblicamente. Per accedere a tali report, è possibile autenticarsi su AWS Management Console e utilizzare AWS Artifact, un portale self-service gratuito per l'accesso on demand ai report di conformità di AWS.

Fonti eventi Lambda

Lambda si integra con oltre 140 servizi AWS tramite integrazione diretta e il bus eventi di Amazon EventBridge. Le fonti eventi Lambda comunemente utilizzate sono:

- Amazon API Gateway
- Amazon CloudWatch Events
- Amazon CloudWatch Logs
- Amazon DynamoDB Streams
- Amazon EventBridge
- Amazon Kinesis Data Streams
- Amazon S3
- Amazon SNS
- Amazon SQS
- AWS Step Functions

Con queste fonti eventi è possibile:

- Utilizzare AWS Identity and Access Management per gestire l'accesso al servizio e alle risorse in modo sicuro.

- Crittografare i dati a riposo.* Tutti i servizi crittografano i dati in transito.
- Accedere dal proprio Amazon Virtual Private Cloud utilizzando gli endpoint VPC (basati su tecnologia AWS PrivateLink)
- Utilizzare Amazon CloudWatch Application Insights per raccogliere, creare report e generare allarmi sui parametri.
- Utilizzare AWS CloudTrail per registrare, monitorare e mantenere continuamente le attività dell'account relative alle operazioni sull'infrastruttura AWS, fornendo una cronologia completa degli eventi delle operazioni intraprese tramite AWS Management Console >SDK AWS, strumenti a riga di comando e altri servizi AWS.

*Al momento della pubblicazione, la crittografia dei dati a riposo non era disponibile per Amazon EventBridge. Per aggiornamenti su queste funzionalità monitorare con continuità le homepage dei servizi.

Conclusione

AWS Lambda offre un potente kit di strumenti per la creazione di applicazioni sicure e scalabili. Molte delle best practice per la sicurezza e la conformità in Lambda sono analoghe a quelle di tutti i servizi AWS, ma alcune sono specifiche di tale servizio. Questo whitepaper descrive i vantaggi di Lambda, la sua idoneità per le applicazioni e l'ambiente di runtime gestito da Lambda. Include anche informazioni su monitoraggio e audit e best practice per la sicurezza e la conformità. Nel momento in cui si riflette sulla propria prossima implementazione, è importante prendere in considerazione ciò che si e imparato su AWS Lambda e valutare questo come potrebbe migliorare la prossima soluzione per il carico di lavoro.

Collaboratori

I collaboratori di questo documento includono:

- Mayank Thakkar, Global Life Sciences Solutions Architect
- Marc Brooker, Senior Principal Engineer (Serverless)
- Osman Surkatty, Senior Security Engineer (Serverless)

Approfondimenti

Per ulteriori informazioni, consulta:

- Modello di responsabilità condivisa, che illustra il pensiero di AWS sulla sicurezza in generale.
- Best practice sulla sicurezza AWS, che include suggerimenti per il servizio AWS Identity and Access Management (IAM).
- Serverless Applications Lens descrive l'AWS Well-Architected Framework e identifica gli elementi chiave per garantire che i carichi di lavoro siano progettati in base alle best practice.
- Introduzione alla sicurezza di AWS fornisce un'ampia introduzione al modo di pensare alla sicurezza in AWS.
- Rischi e conformità in AWS fornisce una panoramica della conformità in AWS.

Revisioni del documento

Per ricevere una notifica sugli aggiornamenti di questo whitepaper, iscriviti al feed RSS.

update-history-change	update-history-description	update-history-date
Aggiornato	Aggiornamenti significativi	15 febbraio 2021
Pubblicazione originale	Prima pubblicazione del whitepaper	3 gennaio 2019

Avvisi

I clienti sono responsabili della propria valutazione autonoma delle informazioni contenute in questo documento. Questo documento: (a) è solo a scopo informativo, (b) mostra le offerte e le pratiche attuali dei prodotti AWS soggette a modifiche senza preavviso, e (c) non crea alcun impegno o garanzia da parte di AWS e dei suoi affiliati, fornitori o licenziatari. I prodotti o servizi AWS sono forniti "così come sono" senza garanzie, dichiarazioni o condizioni di alcun tipo, sia esplicite che implicite. Le responsabilità e gli obblighi di AWS verso i propri clienti sono disciplinati dagli accordi AWS e il presente documento non fa parte né modifica alcun accordo tra AWS e i suoi clienti.