Laporan Resmi AWS

Gambaran Umum Keamanan AWS Lambda

Gambaran Umum Keamanan AWS Lambda: Laporan Resmi AWS

Table of Contents

Gambaran Umum Keamanan AWS Lambda

Tanggal publikasi: 12 Februari 2021 (Revisi dokumen)

Abstrak

Laporan resmi ini menyajikan penjelasan mendalam tentang layanan AWS Lambda melalui sudut pandang keamanan. Laporan ini memberikan gambaran lengkap tentang layanan, yang berguna untuk pengadopsi baru, dan memperdalam pemahaman Lambda untuk pengguna yang sudah ada.

Sasaran audiens laporan resmi ini adalah Kepala Petugas Keamanan Informasi (Chief Information Security Officer/CISO), rekayasawan keamanan informasi, arsitek korporasi, tim kepatuhan, dan siapa pun yang tertarik untuk memahami dasar-dasar AWS Lambda.

Pengantar

Saat ini, semakin banyak beban kerja menggunakan AWS Lambda untuk mencapai skalabilitas, performa, dan efisiensi biaya, tanpa perlu mengelola infrastruktur yang mendasarinya. Beban kerja ini menskalakan hingga ribuan permintaan konkuren per detik. Lambda adalah salah satu dari sekian banyak layanan penting yang ditawarkan oleh AWS saat ini. Lambda digunakan oleh ratusan ribu pelanggan Amazon Web Services (AWS) untuk melayani triliunan permintaan setiap bulan.

Lambda cocok untuk aplikasi bermisi kritis di banyak industri. Berbagai macam pelanggan, mulai dari media dan hiburan hingga layanan keuangan dan industri yang diatur lainnya, memanfaatkan Lambda. Pelanggan mengurangi waktu masuk pasar, mengoptimalkan biaya, dan meningkatkan ketangkasan dengan berfokus pada hal yang paling penting: menjalankan bisnis.

Model lingkungan waktu aktifterkelola memungkinkan Lambda mengelola banyak detail implementasi beban kerja nirserver yang sedang berjalan. Model ini semakin mengurangi permukaan serangan sekaligus menyederhanakan keamanan cloud. Laporan resmi ini menyajikan dasar-dasar model tersebut, bersama dengan praktik terbaik, kepada developer, analis keamanan, tim keamanan dan kepatuhan, serta pemangku kepentingan lainnya.

Tentang AWS Lambda

AWS Lambda adalah layanan komputasi nirserver yang didorong peristiwa yang memperluas layanan AWS lainnya dengan logika kustom, atau membuat layanan backend lainnya yang beroperasi dengan skala, performa, dan keamanan. Lambda dapat menjalankan kode secara otomatis sebagai respons terhadap beberapa peristiwa, seperti permintaan HTTP melalui Amazon API Gateway, modifikasi objek di bucket Amazon S3, pembaruan tabel di Amazon DynamoDB, dan transisi status di AWS Step Functions. Anda juga dapat menjalankan kode langsung dari web atau aplikasi seluler mana pun. Lambda menjalankan kode pada infrastruktur komputasi dengan ketersediaan yang sangat baik, dan melakukan semua administrasi platform yang mendasarinya, termasuk pemeliharaan server dan sistem operasi, penyediaan kapasitas dan penskalaan otomatis, penerapan patch, pemantauan kode, serta pencatatan log.

Dengan Lambda, cukup unggah kode dan konfigurasi waktu pemanggilannya; sisanya Lambda akan mengatasi segala hal yang diperlukan untuk menjalankan kode Anda dengan ketersediaan tinggi. Lambda terintegrasi dengan banyak layanan AWS lainnya dan memungkinkan Anda membuat aplikasi nirserver atau layanan backend, mulai dari tugas otomatisasi sederhana yang dipicu secara berkala hingga aplikasi layanan mikro lengkap.

Lambda juga dapat dikonfigurasikan untuk mengakses sumber daya dalam Amazon Virtual Private Cloud, terlebih lagi, sumber daya on-premise.

Anda dapat melengkapi Lambda dengan postur keamanan yang kuat menggunakan AWS Identity and Access Management (IAM), dan teknik lain yang dibahas dalam laporan resmi ini untuk mempertahankan tingginya tingkat keamanan dan audit, serta memenuhi keperluan terkait kepatuhan.

Topik

Manfaat Lambda

Pelanggan yang ingin berkreasi dan mempercepat organisasi pengembangan tanpa mengganggu kemampuan tim IT mereka untuk memberikan infrastruktur terkelola yang efisien biaya dan dapat diskalakan mengakui bahwa AWS Lambda membantu mereka mengubah kerumitan operasional menjadi ketangkasan dan harga yang lebih baik, tanpa mengorbankan skala atau keandalan.

Lambda menawarkan banyak manfaat, termasuk berikut ini:

Tidak perlu mengelola server.

Lambda menjalankan kode Anda pada infrastruktur yang toleran terhadap kesalahan dan sangat tersedia yang tersebar di beberapa Zona Ketersediaan (AZ) di satu Wilayah, mempermudah deployment kode dan menyediakan semua administrasi, pemeliharaan, serta patch infrastruktur. Lambda juga menyediakan pencatatan dan pemantauan bawaan, termasuk integrasi dengan Amazon CloudWatch, CloudWatch Logs, serta AWS CloudTrail.

Penskalaan berkelanjutan

Lambda mengelola penskalaan fungsi (atau aplikasi) Anda dengan menjalankan kode peristiwa yang dipicu secara paralel, dan memproses setiap peristiwa secara terpisah.

Pengukuran milidetik

Dengan AWS Lambda, Anda dikenakan biaya pengoperasian kode per 1 milidetik (md), dan berapa kali kode dipicu. Anda membayar throughput yang konsisten atau durasi eksekusi, bukan berdasarkan unit server.

Meningkatkan inovasi

Lambda mengambil alih pengelolaan infrastruktur sehingga sumber daya Anda bisa bebas dan lebih fokus pada inovasi dan pengembangan logika bisnis.

Modernisasikan aplikasi Anda

Lambda memungkinkan Anda untuk menggunakan fungsi dengan model machine learning yang sudah terlatih untuk memasukkan kecerdasan buatan ke dalam aplikasi dengan mudah. Permintaan antarmuka pemrograman aplikasi (API) tunggal dapat mengklasifikasikan gambar, menganalisis video, mengonversi pidato ke teks, melakukan pemrosesan bahasa alami, dan banyak lagi.

Ekosistem yang kaya

Lambda mendukung developer melalui AWS Serverless Application Repository untuk menemukan, men-deploy, dan menerbitkan aplikasi nirserver, AWS Serverless Application Model untuk membuat aplikasi nirserver dan integrasi dengan berbagai lingkungan pengembangan terintegrasi (IDE) seperti AWS Cloud9, AWS Toolkit for Visual Studio, AWS Tools for Visual Studio Team Services, dan

beberapa lainnya. Lambda terintegrasi dengan layanan AWS lainnya untuk memberikan ekosistem yang kaya untuk membuat aplikasi nirserver.

Biaya untuk menjalankan aplikasi berbasis Lambda

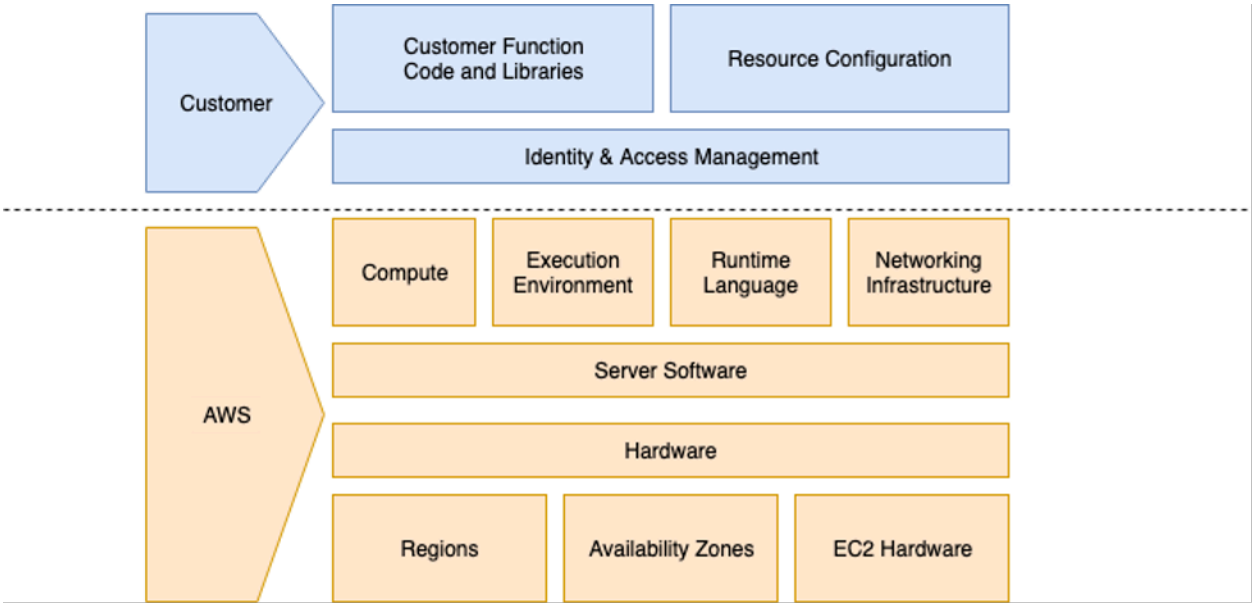
Lambda menawarkan model penentuan harga bayar sesuai penggunaan yang terperinci. Dengan model ini, Anda dikenakan biaya berdasarkan jumlah pemanggilan fungsi dan durasinya (waktu yang diperlukan untuk menjalankan kode). Selain model harga fleksibel ini, Lambda juga menawarkan 1 juta permintaan gratis setiap bulan, yang memungkinkan banyak pelanggan untuk mengotomatiskan proses tanpa biaya apa pun.

Model Tanggung Jawab Bersama

Keamanan dan Kepatuhan merupakan tanggung jawab bersama antara AWS dan pelanggan. Model tanggung jawab bersama ini dapat membantu meringankan beban operasional Anda, karena AWS mengoperasikan, mengelola, dan mengontrol komponen dari sistem operasi host dan lapisan virtualisasi hingga keamanan fisik pada fasilitas tempat layanan tersebut beroperasi.

Untuk AWS Lambda, AWS mengelola layanan dasar dan infrastruktur yang mendasari, sistem operasi, serta platform aplikasi. Anda bertanggung jawab atas keamanan kode serta identity and access management (IAM) Anda ke layanan Lambda dan dalam fungsi Anda.

Gambar 1 menunjukkan model tanggung jawab bersama berlaku untuk komponen umum dan khusus dari AWS Lambda. Tanggung jawab AWS ditunjukkan dengan garis putus-putus berwarna oranye, dan tanggung jawab pelanggan ditunjukkan dengan garis putus-putus berwarna biru.



Gambar 1 - Model tanggung jawab bersama untuk AWS Lambda

Fungsi dan Lapisan Lambda

Dengan Lambda, Anda dapat menjalankan kode secara virtual tanpa administrasi infrastruktur yang mendasarinya. Anda hanya bertanggung jawab atas kode yang Anda berikan kepada Lambda, dan konfigurasi tentang cara Lambda menjalankan kode tersebut atas nama Anda. Saat ini, Lambda mendukung dua jenis sumber daya kode: Fungsi dan Lapisan.

Fungsi adalah sumber daya yang dapat dipanggil untuk menjalankan kode Anda di Fungsi Lambda dapat mencakup sumber daya bersama yang umum yang disebut Lapisan. Lapisan dapat digunakan untuk membagikan kode atau data umum di berbagai fungsi atau akun AWS. Anda bertanggung jawab atas pengelolaan semua kode yang ada dalam fungsi atau lapisan Anda. Saat menerima kode lapisan atau fungsi dari pelanggan, Lambda melindungi akses ke sana dengan enkripsi at-rest menggunakan AWS Key Management Service (AWS KMS) dan in-transit menggunakan TLS 1.2+.

Anda dapat mengelola akses ke fungsi dan lapisan melalui kebijakan AWS Lambda, atau melalui izin berbasis sumber daya. Lihat Layanan AWS yang berkonfigurasi dengan IAM untuk mengetahui daftar lengkap fitur IAM yang didukung di IAM.

Anda juga dapat mengendalikan keseluruhan siklus hidup fungsi dan lapisan melalui API bidang kendali Lambda. Contohnya, Anda dapat menghapus fungsi dengan menelepon `DeleteFunction`, atau mencabut izin dari akun lain dengan menelepon `RemovePermission`.

Mode Pemanggilan Lambda

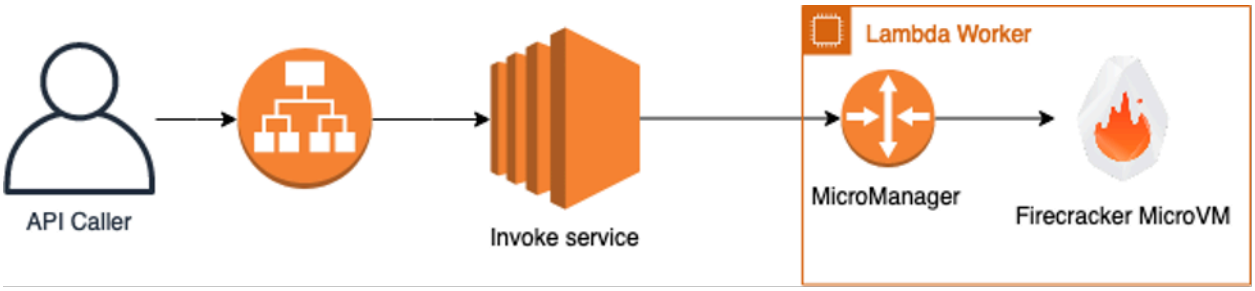
API Pemanggilan dapat dipanggil dalam dua mode: mode peristiwa dan mode permintaan respons.

- Modeperistiwa mengantrekan muatan untuk pemanggilan asinkron.
- Mode permintaan-respons memanggil fungsi dengan muatan yang disediakan dengan sinkron dan mengembalikan respons secepat mungkin.

Dalam keduanya, eksekusi fungsi selalu dilakukan dalam lingkungan eksekusi Lambda, tetapi muatan mengambil jalur yang berbeda. Untuk informasi lebih lanjut, lihat "Lingkungan Eksekusi Lambda" di dokumen ini.

Anda juga dapat menggunakan layanan AWS lain yang melakukan pemanggilan atas nama Anda. Mode pemanggilan mana yang digunakan bergantung pada layanan AWS yang Anda gunakan, dan cara konfigurasinya. Untuk informasi lainnya terkait bagaimana layanan AWS lain berintegrasi dengan Lambda, lihat Menggunakan AWS Lambda dengan layanan lain.

Saat Lambda menerima panggilan permintaan respons, maka akan langsung diteruskan ke layanan pemanggilan. Jika layanan pemanggilan tidak tersedia, penelepon dapat mengantrekan sementara muatan sisi klien untuk mencoba kembali pemanggilan beberapa kali. Jika layanan pemanggilan menerima muatan, layanan kemudian mencoba untuk mengidentifikasi lingkungan eksekusi yang tersedia untuk permintaan, dan meneruskan muatan ke lingkungan eksekusi tersebut untuk menyelesaikan pemanggilan. Jika lingkungan eksekusi tidak ada atau tidak sesuai, maka satu lingkungan akan dibuat sebagai respons dari permintaan. Sedangkan secara in-transit, muatan pemanggilan yang dikirim ke layanan pemanggilan diamankan dengan TLS 1.2 +. Lalu lintas dalam layanan Lambda (dari penyeimbang beban) melewati virtual private cloud (VPC) internal yang terisolasi, yang dimiliki oleh layanan Lambda, dalam Wilayah AWS tempat permintaan dikirim.



Gambar 2 - Model pemanggilan untuk AWS Lambda respons permintaan

Muatan modepemanggilan peristiwa selalu antre untuk diproses sebelum dipanggil. Semua muatan diantrekan untuk diproses dalam antrean Amazon Simple Queue Service (Amazon SQS). Peristiwa antrean selalu diamankan secara in-transit dengan TLS 1.2+, tetapi saat ini tidak dienkripsi at-rest. Antrean Amazon SQS yang digunakan oleh Lambda dikelola oleh layanan Lambda, dan tidak terlihat oleh Anda sebagai pelanggan. Peristiwa yang diantrekan dapat disimpan dalam antrean bersama, tetapi dapat dimigrasi atau ditetapkan ke antrean khusus, tergantung pada sejumlah faktor yang tidak dapat dikendalikan secara langsung oleh pelanggan (misalnya, tingkat pemanggilan, ukuran peristiwa, dan sebagainya).

Peristiwa yang diantrekan diambil dalam batch oleh armada poller Lambda. Armada poller adalah sekelompok instans EC2 yang bertujuan untuk memproses pemanggilan antrean peristiwa yang belum diproses. Armada poller mengambil antrean peristiwa yang perlu diproses dengan meneruskannya ke layanan pemanggilan, seperti yang dilakukan pelanggan dalam pemanggilan mode permintaan respons.

Jika pemanggilan tidak dapat dilakukan, armada poller akan menyimpan peristiwa sementara, dalam memori, pada host sampai berhasil menyelesaikan eksekusi, atau sampai jumlah upaya coba lagi eksekusi terlampaui. Tidak ada data muatan yang pernah ditulis ke disk pada armada poller itu sendiri. Armada polling dapat ditugaskan di seluruh pelanggan AWS, memungkinkan waktu pemanggilan paling singkat. Untuk informasi lebih lanjut tentang layanan mana yang mungkin menggunakan mode pemanggilan peristiwa, lihat Menggunakan AWS Lambda dengan layanan lain.

Eksekusi Lambda

Saat menjalankan fungsi atas perintah Anda, Lambda mengelola penyediaan dan konfigurasi sistem dasar yang diperlukan untuk menjalankan kode Anda. Hal ini memungkinkan developer untuk fokus pada logika bisnis dan menulis kode, bukan membuat administrasi dan mengelola sistem yang mendasarinya.

Layanan Lambda dibagi menjadi bidang kendali dan bidang data. Setiap bidang mewakili tujuan yang berbeda dalam layanan. Bidang kendali memberikan API manajemen (misalnya, `CreateFunction`, `UpdateFunctionCode`, `PublishLayerVersion`, dan sebagainya), serta mengelola integrasi dengan semua layanan AWS. Komunikasi ke bidang kendali Lambda dilindungi secara in-transit oleh TLS. Semua data pelanggan yang disimpan dalam bidang kendali Lambda dienkripsi at-rest menggunakan AWS KMS, yang didesain untuk melindungi dari gangguan atau pengungkapan yang tidak sah.

Bidang data adalah API Panggilan Lambda yang memicu pemanggilan fungsi Lambda. Saat fungsi Lambda dipanggil, bidang data mengalokasikan lingkungan eksekusipada Worker AWS Lambda (atau Worker saja, sejenis instans Amazon EC2) ke versi fungsi, atau memilih lingkungan eksekusi yang sudah ada yang telah diatur untuk versi fungsi tersebut, yang kemudian digunakan untuk menyelesaikan pemanggilan. Untuk informasi selengkapnya, lihat bagian "MicroVM dan Worker AWS Lambda" pada dokumen ini.

Lingkungan eksekusi Lambda

Setiap pemanggilan dirutekan oleh layanan panggilan Lambda ke lingkungan eksekusi pada Worker yang mampu melayani permintaan. Selain melalui bidang data, pelanggan dan pengguna lain tidak dapat langsung memulai komunikasi jaringan inbound/ingress dengan lingkungan eksekusi. Hal ini membantu memastikan bahwa komunikasi ke lingkungan eksekusi Anda diautentikasi dan sah.

Lingkungan eksekusi disimpan untuk versi fungsi tertentu dan tidak dapat digunakan kembali di versi fungsi, fungsi, atau akun AWS lainnya. Artinya, satu fungsi yang mungkin memiliki dua versi yang berbeda akan menghasilkan setidaknya dua lingkungan eksekusi yang unik.

Setiap lingkungan eksekusi hanya dapat digunakan untuk satu pemanggilan konkuren sekaligus, dan dapat digunakan kembali di beberapa pemanggilan versi fungsi yang sama untuk alasan performa. Tergantung pada beberapa faktor (misalnya, tingkat pemanggilan, konfigurasi fungsi, dan sebagainya), satu atau beberapa lingkungan eksekusi mungkin ada untuk versi fungsi tertentu. Dengan pendekatan ini, Lambda mampu memberikan isolasi tingkat versi fungsi bagi pelanggannya.

Saat ini, Lambda tidak mengisolasi panggilan dalam lingkungan eksekusi versi fungsi. Artinya, satu pemanggilan dapat meninggalkan situasi yang dapat memengaruhi pemanggilan berikutnya (misalnya, file yang ditulis ke/tmp atau data dalam memori). Jika Anda ingin memastikan bahwa satu pemanggilan tidak berpengaruh pada pemanggilan lain, Lambda merekomendasikan agar Anda membuat fungsi tambahan yang berbeda. Misalnya, Anda dapat membuat fungsi yang berbeda untuk operasi parsing yang rumit dan lebih rentan terhadap kesalahan, serta menggunakan kembali fungsi yang tidak melakukan operasi sensitif keamanan. Saat ini, Lambda tidak membatasi jumlah fungsi yang dapat pelanggan buat. Untuk informasi lebih lanjut tentang batasan, lihat halaman Kuota Lambda.

Lingkungan eksekusi terus dipantau dan dikelola oleh Lambda, dan lingkungan tersebut dapat dibuat atau dihancurkan karena sejumlah alasan termasuk, namun tidak terbatas pada:

- Ada pemanggilan baru dan tidak ada lingkungan eksekusi yang cocok
- Terjadi deployment perangkat lunak Worker waktu aktif internal
- Konfigurasi konkurensi yang disediakan diterbitkan
- Waktu sewa di lingkungan eksekusi, atau Worker, mendekati atau telah melampaui masa pakai maksimum
- Proses penyeimbangan ulang beban kerja internal lainnya

Pelanggan dapat mengelola jumlah lingkungan eksekusi yang sudah disediakan yang ada untuk versi fungsi dengan mengonfigurasi konkurensi yang disediakan pada konfigurasi fungsi mereka. Jika dikonfigurasi demikian, Lambda akan membuat, mengelola dan memastikan jumlah lingkungan eksekusi yang dikonfigurasi selalu ada. Hal ini memastikan bahwa pelanggan memiliki kontrol yang lebih besar atas performa perusahaan rintisan dari aplikasi nirserver pada berbagai skala.

Selain melalui konfigurasi konkurensi yang disediakan, pelanggan tidak dapat berkali-kali mengontrol jumlah lingkungan eksekusi yang dibuat atau dikelola oleh Lambda sebagai respons atas pemanggilan.

Peran eksekusi

Setiap fungsi Lambda juga harus dikonfigurasi dengan peran eksekusi, yaitu IAM role yang diasumsikan oleh layanan Lambda saat melakukan operasi bidang kendali dan bidang data yang terkait dengan fungsi tersebut. Layanan Lambda mengasumsikan peran ini untuk mengambil kredensial keamanan sementara yang kemudian tersedia sebagai variabel lingkungan selama pemanggilan fungsi. Demi performa, layanan Lambda akan meng-cache kredensial ini, dan dapat

menggunakannya kembali di berbagai lingkungan eksekusi yang menggunakan peran eksekusi yang sama.

Untuk memastikan kepatuhan terhadap prinsip hak akses paling rendah, Lambda merekomendasikan agar setiap fungsi memiliki peran unik, dan dikonfigurasi dengan serangkaian izin minimum yang diperlukan.

Layanan Lambda juga dapat mengasumsikan peran eksekusi untuk melakukan operasi bidang kendali tertentu, seperti yang terkait dengan pembuatan dan konfigurasi Antarmuka jaringan elastis (ENI) untuk fungsi VPC, mengirim log ke Wawasan Aplikasi Amazon CloudWatch, mengirim pelacakan ke AWS X-Ray, atau operasi non-panggilan terkait lainnya. Pelanggan dapat terus meninjau dan mengaudit kasus penggunaan ini dengan meninjau log audit di AWS CloudTrail.

Untuk informasi lebih lanjut tentang subjek ini, lihat halaman dokumentasi Peran eksekusi AWS Lambda.

MicroVM dan Worker Lambda

Lambda akan membuat lingkungan eksekusi di armada instans Amazon EC2 yang disebut Worker AWS Lambda. Worker adalah instans EC2 Nitro bare metal yang diluncurkan dan dikelola oleh Lambda dalam akun AWS terisolasi terpisah yang tidak terlihat oleh pelanggan. Worker memiliki satu atau beberapa Mesin Virtual Mikro (MVM) dengan virtualisasi perangkat keras buatan Firecracker. Firecracker adalah Monitor Mesin Virtual (VMM) sumber terbuka yang menggunakan Mesin Virtual Berbasis Kernel (KVM) Linux untuk membuat dan mengelola MVM. Mesin ini dirancang khusus untuk membuat dan mengelola kontainer multi-penghuni dan layanan berbasis fungsi yang aman yang menyediakan model operasional nirserver. Untuk informasi lebih lanjut tentang model keamanan Firecracker, lihat situs web proyek Firecracker.

Sebagai bagian dari model tanggung jawab bersama, Lambda bertanggung jawab untuk memelihara konfigurasi keamanan, kontrol, dan tingkat patching Worker. Tim Lambda menggunakan Amazon Inspector untuk menemukan potensi masalah keamanan yang sudah umum, serta mekanisme notifikasi masalah keamanan khusus lainnya dan daftar pra-pengungkapan, sehingga pelanggan tidak perlu mengelola postur keamanan yang mendasari lingkungan eksekusi mereka.

Gambar 3 – Model isolasi untuk Worker AWS Lambda

Masa sewa maksimum Worker adalah 14 jam. Jika Worker mendekati waktu sewa maksimum, tidak akan ada pemanggilan yang akan dirutekan, MVM akan diakhiri, dan instans yang mendasari Worker

juga diakhiri. Lambda terus memantau dan memberikan alarm pada aktivitas siklus hidup dari masa pakai armada.

Semua komunikasi bidang data ke workers dienkripsi menggunakan Advanced Encryption Standard dengan Galois/Counter Mode (AES-GCM). Selain melalui bidang data, pelanggan tidak dapat berinteraksi langsung dengan worker karena di-host di jaringan terisolasi Amazon VPC yang dikelola oleh Lambda di akun layanan Lambda.

Saat Worker perlu membuat lingkungan eksekusi baru, maka akan diberi otorisasi terbatas waktu untuk mengakses artefak fungsi pelanggan. Artefak ini dioptimalkan khusus untuk lingkungan eksekusi dan worker Lambda. Kode fungsi yang diunggah menggunakan format ZIP dioptimalkan sekali, kemudian disimpan dalam format terenkripsi menggunakan AES-GCM dan kunci yang dikelola AWS.

Fungsi yang diunggah ke Lambda menggunakan format gambar kontainer juga dioptimalkan. Gambar kontainer pertama kali diunduh dari sumber aslinya, dioptimalkan menjadi potongan yang berbeda, dan kemudian disimpan sebagai potongan terenkripsi menggunakan metode enkripsi konvergen yang diautentikasi yang menggunakan kombinasi AES-CTR, AES-GCM, dan MAC SHA-256. Metode enkripsi konvergen memungkinkan Lambda untuk mendeduplikasi potongan terenkripsi dengan aman. Semua kunci yang diperlukan untuk membatalkan enkripsi data pelanggan dilindungi menggunakan AWS KMSKunci Master Pelanggan (CMK) yang dikelola pelanggan. Penggunaan CMK oleh layanan Lambda tersedia untuk pelanggan di log AWS CloudTrail untuk pelacakan dan audit.

Teknologi Isolasi Lambda

Lambda menggunakan berbagai teknologi isolasi sumber terbuka dan eksklusif untuk melindungi Workers dan lingkungan eksekusi. Setiap lingkungan eksekusi berisi salinan khusus item berikut:

- Kode versi fungsi tertentu
- Lapisan AWS Lambda yang dipilih untuk versi fungsi Anda
- Waktu aktif fungsi yang dipilih (misalnya, Java 11, NodeJS 12, Python 3.8, dan sebagainya) waktu aktif kustom fungsi
- Sebuah direktori/tmp yang dapat ditulis
- Ruang pengguna Linux minimal berdasarkan Amazon Linux 2

Lingkungan eksekusi diisolasi satu sama lain menggunakan beberapa teknologi seperti kontainer yang dibangun ke dalam kernel Linux, bersama dengan teknologi isolasi dari AWS. Teknologi ini meliputi:

- cgroups– Digunakan untuk membatasi akses fungsi ke CPU dan memori.
- namespaces – Setiap lingkungan eksekusi berjalan di namespace khusus. Kami melakukan ini dengan menyerahkan pengelolaan ID proses grup yang unik, ID pengguna, antarmuka jaringan, dan sumber daya lainnya ke kernel Linux.
- seccomp-bpf – Untuk membatasi panggilan sistem (syscalls) yang dapat digunakan dari dalam lingkungan eksekusi.
- iptables dan tabel perutean – Untuk mencegah komunikasi jaringan masuk dan untuk mengisolasi koneksi jaringan antarMVM.
- chroot – Memberikan akses terbatas ke sistem file yang mendasarinya.
- Konfigurasi Firecracker – Digunakan untuk menilai perangkat blok batas dan throughput perangkat jaringan.
- Fitur keamanan Firecracker – Untuk informasi lebih lanjut tentang desain keamanan terbaru Firecracker, lihat dokumen desain terbaru Firecracker .

Bersama dengan teknologi isolasi dari AWS, mekanisme ini memberikan isolasi yang kuat antara lingkungan eksekusi.

Penyimpanan dan status

Lingkungan eksekusi tidak pernah digunakan kembali di berbagai versi fungsi atau pelanggan, tetapi satu lingkungan dapat digunakan kembali pada pemanggilan versi fungsi yang sama. Artinya, data dan status dapat bertahan di antara pemanggilan. Data dan/atau status dapat bertahan selama berjam-jam sebelum dihancurkan sebagai bagian dari pengelolaan siklus hidup lingkungan eksekusi normal. Demi performa, fungsi dapat memanfaatkan perilaku ini untuk meningkatkan efisiensi dengan mempertahankan dan menggunakan kembali cache lokal atau koneksi jangka panjang di antara pemanggilan. Di dalam lingkungan eksekusi, beberapa pemanggilan ini ditangani oleh satu proses, sehingga setiap status dalam proses (seperti status statis di Jawa) dapat tersedia untuk digunakan kembali untuk pemanggilan berikutnya, jika pemanggilan terjadi pada lingkungan eksekusi yang digunakan kembali.

Setiap lingkungan eksekusi Lambda juga mencakup filesystem yang dapat ditulis, yang tersedia di `/tmp`. Penyimpanan ini tidak dapat diakses atau dibagikan di antara lingkungan eksekusi. Seperti status proses, file yang ditulis ke `/tmp` tetap ada selama masa pakai lingkungan eksekusi. Hal ini memungkinkan operasi transfer yang mahal, seperti mengunduh model machine learning (ML), untuk diamortisasi di beberapa pemanggilan. Fungsi yang tidak ingin menyimpan data di antara pemanggilan sebaiknya tidak menulis ke /tmp, atau menghapus filenya dari /tmp di antara pemanggilan. `/tmp` Direktori didukung oleh Penyimpanan instans Amazon EC2 dan dienkripsi secara at-rest.

Pelanggan yang ingin menyimpan data ke sistem file di luar lingkungan eksekusi sebaiknya menggunakan integrasi Lambda dengan Amazon Elastic File System (Amazon EFS). Untuk informasi lebih lanjut, lihat Menggunakan Amazon EFS dengan AWS Lambda.

Jika pelanggan tidak ingin menyimpan data atau status seluruh pemanggilan, Lambda tidak menyarankan untuk menggunakan konteks eksekusi atau lingkungan eksekusi untuk menyimpan data atau status. Jika pelanggan ingin aktif mencegah data atau kebocoran status di seluruh pemanggilan, Lambda merekomendasikan untuk membuat fungsi yang berbeda untuk setiap status. Lambda tidak menyarankan untuk menggunakan atau menyimpan status sensitif keamanan ke dalam lingkungan eksekusi, karena dapat bermutasi antara pemanggilan. Sebagai gantinya, sebaiknya hitung ulang status pada setiap pemanggilan.

Pemeliharaan Waktu Aktif di Lambda

Lambda menyediakan dukungan untuk waktu aktif ini dengan terus memindai dan men-deploy pembaruan dan patch keamanan yang kompatibel, dan dengan melakukan aktivitas pemeliharaan waktu aktif lainnya. Hal ini memungkinkan pelanggan untuk fokus hanya pada pemeliharaan dan keamanan kode apa pun yang termasuk dalam Fungsi dan Lapisan mereka. Tim Lambda menggunakan Amazon Inspector untuk menemukan masalah keamanan yang sudah umum, serta mekanisme notifikasi masalah keamanan khusus lainnya dan daftar pra-pengungkapan untuk memastikan bahwa bahasa waktu aktif dan lingkungan eksekusi kami tetap di-patch. Jika teridentifikasi ada patch baru atau pembaruan patch, Lambda akan menguji dan men-deploy pembaruan waktu aktif tanpa melibatkan pelanggan. Untuk informasi lebih lanjut tentang program kepatuhan Lambda, lihat bagian "Lambda dan Kepatuhan" dari dokumen ini.

Biasanya, tidak perlu adanya tindakan untuk mengambil patch terbaru untuk waktu aktif Lambda yang didukung, tetapi terkadang diperlukan tindakan untuk menguji patch sebelum di-deploy (misalnya, patch waktu aktif yang tidak kompatibel). Jika ada tindakan yang diperlukan oleh pelanggan, Lambda akan menghubungi mereka melalui Personal Health Dashboard, email akun AWS, atau cara lain, dengan tindakan spesifik yang harus diambil.

Pelanggan dapat menggunakan bahasa pemrograman lain di Lambda dengan menerapkan waktu aktif kustom. Untuk waktu aktif kustom, pemeliharaan waktu aktif menjadi tanggung jawab pelanggan, termasuk memastikan bahwa waktu aktif kustom mencakup patch keamanan terbaru. Untuk informasi lebih lanjut, lihat Waktu aktif AWS Lambda kustom dalam Panduan Developer AWS Lambda.

Saat pengelola bahasa waktu aktif upstream menandai bahasa dengan End-Of-Life (EOL), Lambda merespons dengan tidak lagi mendukung versi bahasa waktu aktif. Saat versi waktu aktif ditandai sebagai tidak digunakan lagi di Lambda, Lambda berhenti mendukung pembuatan fungsi baru dan pembaruan ke fungsi yang ada yang ditulis dalam waktu aktif yang sudah tidak berlaku. Sebagai pemberitahuan kepada pelanggan tentang berhentinya waktu aktif berikutnya, Lambda mengirim notifikasi kepada pelanggan tanggal penghentian berikutnya, dan kemungkinan hal yang akan terjadi. Lambda tidak menyediakan pembaruan keamanan, dukungan teknis, atau hotfix untuk waktu aktif yang sudah tidak berlaku, dan berhak untuk menonaktifkan pemanggilan fungsi yang dikonfigurasikan untuk berjalan pada waktu aktif yang sudah tidak berlaku kapan pun. Jika pelanggan ingin terus menjalankan versi waktu aktif yang sudah tidak berlaku atau tidak didukung, mereka dapat membuat waktu aktif AWS Lambda kustom mereka sendiri. Untuk detail terkait kapan waktu aktif sudah tidak berlaku, lihat Kebijakan dukungan Waktu Aktif AWS Lambda.

Pemantauan dan Audit Fungsi Lambda

Anda dapat memantau dan mengaudit fungsi Lambda dengan berbagai layanan dan metode AWS, termasuk layanan berikut.

Amazon CloudWatch

AWS Lambda secara otomatis memantau fungsi Lambda atas nama Anda. Melalui Amazon CloudWatch, Lambda melaporkan metrik seperti jumlah permintaan, durasi eksekusi per permintaan, dan jumlah permintaan yang menghasilkan kesalahan. Metrik ini terekspos pada tingkat fungsi, yang kemudian dapat Anda manfaatkan untuk mengatur alarm CloudWatch. Untuk daftar metrik yang diekspos oleh Lambda, lihat AWS Lambda Metrik.

Amazon CloudTrail

Dengan AWS CloudTrail, Anda dapat menerapkan tata kelola, kepatuhan, audit operasional, dan audit risiko seluruh akun AWS Anda, termasuk Lambda. CloudTrail memungkinkan Anda untuk mencatat, terus memantau, dan mempertahankan aktivitas akun yang terkait dengan tindakan di seluruh infrastruktur AWS Anda, menyediakan riwayat peristiwa lengkap tindakan yang diambil melalui AWS Management Console, AWS SDK, alat baris perintah, dan layanan AWS lainnya. Dengan CloudTrail, Anda dapat mengenkripsi berkas log menggunakan AWS KMS dan memanfaatkan integritas berkas log CloudTrail for pernyataan positif.

AWS X-Ray

Dengan AWS X-Ray, Anda dapat menganalisis dan men-debug aplikasi berbasis Lambda terdistribusi dan produksi, membantu Anda memahami kinerja aplikasi dan layanan yang mendasarinya, sehingga Anda dapat mengidentifikasi dan memecahkan penyebab masalah performa dan kesalahan. Tampilan end-to-end permintaan X-Ray saat menjelajah aplikasi Anda menunjukkan peta komponen yang mendasari aplikasi, sehingga Anda dapat menganalisis aplikasi selama pengembangan dan produksi.

AWS Config

Dengan AWS Config, Anda dapat melacak perubahan konfigurasi ke fungsi Lambda (termasuk fungsi yang dihapus), lingkungan waktu aktif, tanda, nama handler, ukuran kode, alokasi memori,

pengaturan waktu habis, dan pengaturan konkruensi, bersama dengan peran eksekusi IAM, subnet, serta asosiasi grup keamanan Lambda. Layanan ini memberi gambaran menyeluruh tentang siklus hidup fungsi Lambda dan memungkinkan Anda untuk memunculkan data tersebut untuk potensi persyaratan audit dan kepatuhan.

Perancangan dan Pengoperasian Fungsi Lambda

Bagian ini membahas arsitektur dan operasi Lambda. Untuk informasi tentang praktik terbaik standar untuk aplikasi nirserver, buka laporan resmi Serverless Applications Lens, yang mendefinisikan dan mengeksplorasi pilar-pilar AWS Well-Architected Framework dalam konteks Nirserver.

- Pilar Keunggulan Operasional – Kemampuan untuk menjalankan dan memantau sistem untuk memberikan nilai bisnis serta terus meningkatkan proses dan prosedur pendukung.
- Pilar Keamanan – Kemampuan untuk melindungi informasi, sistem, dan aset sekaligus memberikan nilai bisnis melalui penilaian risiko dan strategi mitigasi.
- Pilar Keandalan – Kemampuan sistem untuk pulih dari gangguan infrastruktur atau layanan, memperoleh sumber daya komputasi untuk memenuhi permintaan secara dinamis, dan memitigasi gangguan seperti kesalahan konfigurasi atau masalah jaringan sementara.
- Pilar Efisiensi Performa – Penggunaan sumber daya komputasi yang efisien untuk memenuhi persyaratan dan pemeliharaan efisiensi seiring perubahan permintaan dan perkembangan teknologi.
- Pilar Pengoptimalan Biaya – Proses penyempurnaan dan peningkatan yang berkelanjutan untuk memastikan bahwa hasil bisnis tercapai sekaligus meminimalkan biaya seiring perubahan permintaan dan perkembangan teknologi.

Laporan resmi Serverless Applications Lens mencakup topik seperti pencatatan metrik dan alarm, throttling dan batasan, menetapkan izin ke fungsi Lambda, serta membuat data sensitif tersedia untuk fungsi Lambda.

Lambda dan Kepatuhan

Seperti yang disebutkan di bagian "Model Tanggung Jawab Bersama", Anda bertanggung jawab untuk menentukan sistem kepatuhan yang berlaku untuk data Anda. Setelah menentukan sistem kepatuhan, Anda dapat menggunakan berbagai fitur Lambda untuk mencocokkan kontrol tersebut. Anda dapat menghubungi ahli AWS (seperti Arsitek Solusi, ahli domain, manajer akun teknis, dan sumber daya manusia lainnya) untuk mendapatkan bantuan. Namun, AWS tidak dapat memberi tahu pelanggan apakah (atau mana) sistem kepatuhan berlaku untuk kasus penggunaan tertentu.

Mulai November 2020, Lambda masuk dalam laporan SOC 1, SOC 2, dan SOC 3, yang merupakan laporan pemeriksaan pihak ketiga independen yang menunjukkan bagaimana AWS mencapai kontrol dan tujuan kepatuhan utama. Untuk daftar informasi kepatuhan terbaru, lihat halaman Layanan dalam Lingkup AWS menurut Program Kepatuhan.

Beberapa laporan kepatuhan tidak dapat dibagikan secara publik karena bersifat sensitif. Untuk mengakses laporan ini, Anda dapat masuk ke AWS Management Console dan menggunakan AWS Artifact, sebuah portal layanan mandiri gratis, untuk akses sesuai permintaan laporan kepatuhan AWS.

Sumber Peristiwa Lambda

Lambda berintegrasi dengan lebih dari 140 layanan AWS melalui integrasi langsung dan bus peristiwa Amazon EventBridge. Sumber peristiwa Lambda yang umum digunakan adalah:

- 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

Dengan sumber peristiwa ini, Anda dapat:

- Menggunakan AWS Identity and Access Management untuk mengelola akses ke layanan dan sumber daya dengan aman.

- Mengenkripsi data at-rest.* Semua layanan mengenkripsi data saat transit.
- Mengakses dari Amazon Virtual Private Cloud menggunakan VPC endpoint (didukung oleh AWS PrivateLink
- Menggunakan Wawasan Aplikasi Amazon CloudWatch untuk mengumpulkan, melaporkan, dan memberi alarm pada metrik.
- Menggunakan AWS CloudTrail untuk mencatat, terus memantau, dan mempertahankan aktivitas akun terkait tindakan di infrastruktur AWS, memberikan riwayat peristiwa lengkap tentang tindakan yang diambil melalui AWS Management Console >AWS SDK, alat baris perintah, dan Layanan AWS lainnya.

*Saat publikasi, enkripsi data at-rest tidak tersedia untuk Amazon EventBridge. Terus pantau beranda layanan untuk informasi terbaru tentang kemampuan ini.

Kesimpulan

AWS Lambda menawarkan toolkit yang kuat untuk membangun aplikasi yang aman dan dapat diskalakan. Sebagian besar praktik terbaik untuk keamanan dan kepatuhan di Lambda sama seperti yang ada di semua layanan AWS, tetapi ada beberapa yang khusus untuk Lambda. Laporan resmi ini menjelaskan manfaat Lambda, kesesuaiannya untuk aplikasi, dan lingkungan waktu aktif yang dikelola Lambda. Laporan ini juga mencakup informasi tentang pemantauan dan audit, serta praktik terbaik keamanan dan kepatuhan. Saat mempersiapkan implementasi berikutnya, pertimbangkan hal yang Anda pelajari tentang AWS Lambda, dan kontribusinya dalam meningkatkan solusi beban kerja Anda selanjutnya.

Kontributor

Kontributor dalam dokumen ini meliputi:

- Mayank Thakkar, Arsitek Solusi Ilmu Hayati Global (Global Life Sciences Solutions Architect)
- Marc Brooker, Kepala Rekayasawan Senior (Senior Principal Engineer) (Nirserver)
- Osman Surkatty, Rekayasawan Keamanan Senior (Senior Security Engineer) (Nirserver)

Sumber Bacaan Lebih Lanjut

Untuk informasi lainnya, buka:

- Model Tanggung Jawab Bersama, yang menjelaskan pandangan AWS tentang keamanan secara umum.
- Praktik Terbaik Keamanan AWS, yang mencakup rekomendasi untuk layanan AWS Identity and Access Management (IAM).
- Serverless Applications Lens mencakup AWS Well-Architected Framework mengidentifikasi elemen kunci untuk memastikan beban kerja Anda dirancang sesuai dengan praktik terbaik.
- Pengantar Keamanan AWS memberikan pengenalan luas tentang keamanan di AWS.
- Risiko dan Kepatuhan AWS memberikan gambaran umum tentang kepatuhan di AWS.

Revisi dokumen

Untuk mendapatkan notifikasi tentang pembaruan laporan resmi ini, silakan berlangganan RSS feed.

update-history-change	update-history-description	update-history-date
Diperbarui	Pembaruan signifikan	15 Februari 2021
Publikasi pertama	Laporan resmi pertama kali dipublikasikan	3 Januari 2019

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya disediakan sebagai informasi, (b) berisi penawaran produk dan praktik AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak menjadi komitmen atau jaminan apa pun dari AWS dan afiliasi, pemasok, atau pemberi lisensinya. Produk atau layanan AWS disediakan “sebagaimana adanya” tanpa jaminan, representasi, atau ketentuan apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.