AWS Whitepaper

Pengantar DevOps tentang AWS

Pengantar DevOps tentang AWS: AWS Whitepaper

Table of Contents

Pengantar DevOps tentang AWS

Tanggal publikasi: 7 April 2023 () Revisi Dokumen

Saat ini, lebih dari sebelumnya, perusahaan memulai perjalanan transformasi digital mereka untuk membangun koneksi yang lebih dalam dengan pelanggan mereka, untuk mencapai nilai bisnis yang berkelanjutan dan abadi. Organizations dari segala bentuk dan ukuran mengganggu pesaing mereka dan memasuki pasar baru dengan berinovasi lebih cepat dari sebelumnya. Untuk organisasi-organisasi ini, penting untuk fokus pada inovasi dan gangguan perangkat lunak, sehingga penting untuk merampingkan pengiriman perangkat lunak mereka. Organisasi-organisasi yang mempersingkat waktu mereka dari ide ke produksi menjadikan kecepatan dan kelincahan sebagai prioritas bisa menjadi pengganggu masa depan.

Meskipun ada beberapa faktor yang perlu dipertimbangkan dalam menjadi disruptor digital berikutnya, whitepaper ini berfokus pada DevOps, dan layanan dan fitur di platform Amazon Web Services (AWS) yang akan membantu meningkatkan kemampuan organisasi untuk memberikan aplikasi dan layanan dengan kecepatan tinggi.

Pengantar

DevOps adalah kombinasi filosofi budaya, praktik teknik, dan alat yang meningkatkan kemampuan organisasi untuk memberikan aplikasi dan layanan dengan kecepatan tinggi dan kualitas yang lebih baik. Seiring waktu, beberapa praktik penting telah muncul ketika mengadopsi DevOps: integrasi berkelanjutan (CI), pengiriman berkelanjutan (CD), Infrastruktur sebagai Kode (IAc), dan pemantauan dan pencatatan.

Paper ini menyoroti AWS kemampuan yang membantu Anda mempercepat DevOps perjalanan Anda, dan bagaimana AWS layanan dapat membantu menghilangkan angkat berat yang tidak terdiferensiasi yang terkait dengan DevOps adaptasi. Ini juga menjelaskan bagaimana membangun integrasi berkelanjutan dan kemampuan pengiriman tanpa mengelola server atau membangun node, dan bagaimana menggunakan IAC untuk menyediakan dan mengelola sumber daya cloud Anda secara konsisten dan berulang.

- Integrasi berkelanjutan: Praktik pengembangan perangkat lunak di mana pengembang secara teratur menggabungkan perubahan kode mereka ke dalam repositori pusat, setelah itu build dan pengujian otomatis dijalankan.
- Pengiriman berkelanjutan: Praktik pengembangan perangkat lunak di mana perubahan kode secara otomatis dibangun, diuji, dan disiapkan untuk rilis ke produksi.

- Infrastruktur sebagai Kode: Praktik di mana infrastruktur disediakan dan dikelola menggunakan teknik pengembangan kode dan perangkat lunak, seperti kontrol versi, dan integrasi berkelanjutan.
- Pemantauan dan pencatatan: Memungkinkan organisasi untuk melihat bagaimana kinerja aplikasi dan infrastruktur berdampak pada pengalaman pengguna akhir produk mereka.
- Komunikasi dan kolaborasi: Praktik dibuat untuk membawa tim lebih dekat dan dengan membangun alur kerja dan mendistribusikan tanggung jawab untuk. DevOps
- Keamanan: Harus menjadi perhatian lintas pemotongan. Pipeline integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) dan layanan terkait harus dilindungi dan izin kontrol akses yang tepat harus disiapkan.

Pemeriksaan masing-masing prinsip ini mengungkapkan hubungan yang erat dengan penawaran yang tersedia dari. AWS

Apakah Anda sudah Well-Architected?

AWS Well-Architected Framework membantu Anda memahami pro dan kontra dari keputusan yang Anda buat saat membangun sistem di cloud. Enam pilar dari Kerangka Kerja ini memungkinkan Anda mempelajari praktik terbaik arsitektural untuk merancang dan mengoperasikan sistem yang andal, aman, efisien, hemat biaya, dan berkelanjutan. Menggunakan AWS Well-Architected Tool, tersedia tanpa biaya di AWS Management Console, Anda dapat meninjau beban kerja Anda terhadap praktik terbaik ini dengan menjawab serangkaian pertanyaan untuk setiap pilar.

Integrasi berkelanjutan

Continuous Integration (CI) adalah praktik pengembangan perangkat lunak di mana pengembang secara teratur menggabungkan perubahan kode mereka ke dalam repositori kode pusat, setelah itu build dan pengujian otomatis dijalankan. CI membantu menemukan dan mengatasi bug lebih cepat, meningkatkan kualitas perangkat lunak, dan mengurangi waktu yang diperlukan untuk memvalidasi dan merilis pembaruan perangkat lunak baru.

AWS menawarkan layanan berikut untuk integrasi berkelanjutan:

Topik

- AWS CodeCommit
- AWS CodeBuild
- AWS CodeArtifact

AWS CodeCommit

AWS CodeCommitadalah layanan kontrol sumber terkelola yang aman, sangat skalabel, dan dikelola yang menampung repositori git pribadi. CodeCommit mengurangi kebutuhan bagi Anda untuk mengoperasikan sistem kontrol sumber Anda sendiri dan tidak ada perangkat keras untuk penyediaan dan skala atau perangkat lunak untuk menginstal, mengkonfigurasi, dan mengoperasikan. Anda dapat menggunakan CodeCommit untuk menyimpan apa pun mulai dari kode hingga binari, dan mendukung fungsionalitas standar GitHub, memungkinkannya bekerja dengan mulus dengan alat berbasis Git yang ada. Tim Anda juga dapat menggunakan CodeCommit alat kode online untuk menelusuri, mengedit, dan berkolaborasi dalam proyek. AWS CodeCommit memiliki beberapa manfaat:

- Kolaborasi — AWS CodeCommit dirancang untuk pengembangan perangkat lunak kolaboratif. Anda dapat dengan mudah melakukan komit, cabang, dan menggabungkan kode Anda, yang membantu Anda dengan mudah mempertahankan kontrol proyek tim Anda. CodeCommit juga mendukung permintaan tarik, yang menyediakan mekanisme untuk meminta tinjauan kode dan mendiskusikan kode dengan kolaborator.
- Enkripsi — Anda dapat mentransfer file Anda ke dan dari AWS CodeCommit menggunakan HTTPS atau SSH, sesuai keinginan Anda. Repositori Anda juga secara otomatis dienkripsi saat istirahat melalui AWS Key Management Service(AWS KMS) menggunakan kunci khusus pelanggan.

- Kontrol akses — AWS CodeCommit menggunakan AWS Identity and Access Management(IAM) untuk mengontrol dan memantau siapa yang dapat mengakses data Anda selain bagaimana, kapan, dan di mana mereka dapat mengaksesnya. CodeCommit juga membantu Anda memantau repositori Anda melalui dan AWS CloudTrailAmazon. CloudWatch

  Ketersediaan dan daya tahan tinggi - AWS CodeCommit menyimpan repositori Anda di Amazon Simple Storage Service (Amazon S3) dan Amazon DynamoDB. Data terenkripsi Anda disimpan secara berlebihan di beberapa fasilitas. Arsitektur ini meningkatkan ketersediaan dan daya tahan data repositori Anda.

- Notifikasi dan skrip kustom — Anda sekarang dapat menerima pemberitahuan untuk peristiwa yang memengaruhi repositori Anda. Pemberitahuan akan datang sebagai pemberitahuan Amazon Simple Notification Service (Amazon SNS). Setiap pemberitahuan akan menyertakan pesan status serta tautan ke sumber daya yang acaranya menghasilkan pemberitahuan itu. Selain itu, dengan menggunakan isyarat AWS CodeCommit repositori, Anda dapat mengirim notifikasi dan membuat webhook HTTP dengan Amazon SNS atau AWS Lambdamenjalankan fungsi sebagai respons terhadap peristiwa repositori yang Anda pilih.

AWS CodeBuild

AWS CodeBuildadalah layanan integrasi berkelanjutan yang dikelola sepenuhnya yang mengkompilasi kode sumber, menjalankan pengujian, dan menghasilkan paket perangkat lunak yang siap digunakan. Anda tidak perlu menyediakan, mengelola, dan menskalakan server build Anda sendiri. CodeBuild dapat menggunakan salah satu dari GitHub, GitHub Enterprise BitBucket, AWS CodeCommit,, atau Amazon S3 sebagai penyedia sumber.

CodeBuild skala terus menerus dan dapat memproses beberapa build secara bersamaan. CodeBuild menawarkan berbagai lingkungan pra-konfigurasi untuk berbagai versi Microsoft Windows dan Linux. Pelanggan juga dapat membawa lingkungan build khusus mereka sebagai wadah Docker. CodeBuild juga terintegrasi dengan alat open source seperti Jenkins dan Spinnaker.

CodeBuild juga dapat membuat laporan untuk pengujian unit, fungsional, atau integrasi. Laporan ini memberikan pandangan visual tentang berapa banyak kasus pengujian yang dijalankan dan berapa banyak yang lulus atau gagal. Proses pembuatan juga dapat dijalankan di dalam Amazon Virtual Private Cloud (Amazon VPC) yang dapat membantu jika layanan integrasi atau database Anda digunakan di dalam VPC.

AWS CodeArtifact

AWS CodeArtifactadalah layanan repositori artefak yang dikelola sepenuhnya yang dapat digunakan oleh organisasi untuk menyimpan, menerbitkan, dan berbagi paket perangkat lunak dengan aman yang digunakan dalam proses pengembangan perangkat lunak mereka. CodeArtifact dapat dikonfigurasi untuk secara otomatis mengambil paket perangkat lunak dan dependensi dari repositori artefak publik sehingga pengembang memiliki akses ke versi terbaru.

Tim pengembangan perangkat lunak semakin mengandalkan paket open-source untuk melakukan tugas-tugas umum dalam paket aplikasi mereka. Hal ini telah menjadi penting bagi tim pengembangan perangkat lunak untuk mempertahankan kontrol pada versi tertentu dari perangkat lunak open-source untuk memastikan perangkat lunak bebas dari kerentanan. Dengan CodeArtifact, Anda dapat mengatur kontrol untuk menegakkan ini.

CodeArtifact bekerja dengan manajer paket yang umum digunakan dan membangun alat seperti Maven, Gradle, npm, yarn, twine, dan pip, sehingga mudah untuk diintegrasikan ke dalam alur kerja pengembangan yang ada.

Pengiriman berkelanjutan

Continuous delivery (CD) adalah praktik pengembangan perangkat lunak di mana perubahan kode secara otomatis disiapkan untuk rilis ke produksi. Pilar pengembangan aplikasi modern, pengiriman berkelanjutan memperluas integrasi berkelanjutan dengan menerapkan semua perubahan kode ke lingkungan pengujian dan/atau lingkungan produksi setelah tahap pembuatan. Ketika diterapkan dengan benar, pengembang akan selalu memiliki artefak build siap penerapan yang telah melewati proses pengujian standar.

Pengiriman berkelanjutan memungkinkan pengembang mengotomatiskan pengujian lebih dari sekadar pengujian unit sehingga mereka dapat memverifikasi pembaruan aplikasi di berbagai dimensi sebelum diterapkan ke pelanggan.

Pengujian ini mungkin termasuk pengujian UI, pengujian beban, pengujian integrasi, pengujian keandalan API, dan banyak lagi. Ini membantu pengembang memvalidasi pembaruan secara lebih menyeluruh dan menemukan masalah terlebih dahulu. Menggunakan cloud, mudah dan hemat biaya untuk mengotomatiskan pembuatan dan replikasi beberapa lingkungan untuk pengujian, yang sebelumnya sulit dilakukan di tempat.

AWS menawarkan layanan berikut untuk pengiriman berkelanjutan:

- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

Topik

- AWS CodeDeploy
- AWS CodePipeline

AWS CodeDeploy

AWS CodeDeployadalah layanan penyebaran terkelola penuh yang mengotomatiskan penerapan perangkat lunak ke berbagai layanan komputasi seperti Amazon Elastic Compute Cloud (Amazon EC2), AWS Fargate, AWS Lambda dan server lokal Anda. AWS CodeDeploy memudahkan Anda untuk merilis fitur baru dengan cepat, membantu Anda menghindari downtime selama penerapan

aplikasi, dan menangani kompleksitas memperbarui aplikasi Anda. Anda dapat menggunakannya CodeDeploy untuk mengotomatiskan penerapan perangkat lunak, mengurangi kebutuhan akan operasi manual yang rawan kesalahan. Skala layanan agar sesuai dengan kebutuhan penerapan Anda.

CodeDeploy memiliki beberapa manfaat yang selaras dengan DevOps prinsip penerapan berkelanjutan:

- Penerapan otomatis — CodeDeploy sepenuhnya mengotomatiskan penerapan perangkat lunak, memungkinkan Anda untuk menerapkan dengan andal dan cepat.
- Kontrol terpusat — CodeDeploy memungkinkan Anda untuk dengan mudah meluncurkan dan melacak status penerapan aplikasi Anda melalui atau. Konsol Manajemen AWS AWS CLI CodeDeploymemberi Anda laporan terperinci yang memungkinkan Anda melihat kapan dan ke mana setiap revisi aplikasi diterapkan. Anda juga dapat membuat pemberitahuan push untuk menerima pembaruan langsung tentang penerapan Anda.
- Minimalkan waktu henti — CodeDeploy membantu memaksimalkan ketersediaan aplikasi Anda selama proses penyebaran perangkat lunak. Ini memperkenalkan perubahan secara bertahap dan melacak kesehatan aplikasi sesuai dengan aturan yang dapat dikonfigurasi. Penerapan perangkat lunak dapat dengan mudah dihentikan dan diputar kembali jika ada kesalahan.
- Mudah diadopsi - CodeDeploy bekerja dengan aplikasi apa pun, dan memberikan pengalaman yang sama di berbagai platform dan bahasa. Anda dapat dengan mudah menggunakan kembali kode pengaturan yang ada. CodeDeploy juga dapat berintegrasi dengan proses rilis perangkat lunak yang ada atau rantai alat pengiriman berkelanjutan (misalnya,, AWS CodePipeline GitHub, Jenkins).

AWS CodeDeploy mendukung beberapa opsi penerapan. Untuk informasi selengkapnya, lihat bagian Strategi penyebaran dokumen ini.

AWS CodePipeline

AWS CodePipelineadalah layanan pengiriman berkelanjutan yang dapat Anda gunakan untuk memodelkan, memvisualisasikan, dan mengotomatiskan langkah-langkah yang diperlukan untuk merilis perangkat lunak Anda. Dengan AWS CodePipeline, Anda memodelkan proses rilis lengkap untuk membangun kode Anda, menerapkan ke lingkungan pra-produksi, menguji aplikasi Anda, dan merilisnya ke produksi. AWS CodePipeline kemudian membangun, menguji, dan menerapkan aplikasi Anda sesuai dengan alur kerja yang ditentukan setiap kali ada perubahan kode. Anda dapat

mengintegrasikan alat mitra dan alat kustom Anda sendiri ke dalam setiap tahap proses rilis untuk membentuk solusi pengiriman end-to-end berkelanjutan.

AWS CodePipeline memiliki beberapa manfaat yang selaras dengan DevOps prinsip penerapan berkelanjutan:

- Pengiriman cepat — AWS CodePipeline mengotomatiskan proses rilis perangkat lunak Anda, memungkinkan Anda merilis fitur baru dengan cepat kepada pengguna Anda. Dengan CodePipeline, Anda dapat dengan cepat mengulangi umpan balik dan mendapatkan fitur baru untuk pengguna Anda lebih cepat.
- Peningkatan kualitas — Dengan mengotomatiskan proses pembuatan, pengujian, dan rilis, AWS CodePipeline memungkinkan Anda meningkatkan kecepatan dan kualitas pembaruan perangkat lunak dengan menjalankan semua perubahan baru melalui serangkaian pemeriksaan kualitas yang konsisten.
- Mudah diintegrasikan - AWS CodePipeline dapat dengan mudah diperluas untuk beradaptasi dengan kebutuhan spesifik Anda. Anda dapat menggunakan plugin pra-bangun atau plugin kustom Anda sendiri di setiap langkah proses rilis Anda. Misalnya, Anda dapat menarik kode sumber dari GitHub, menggunakan server build Jenkins lokal, menjalankan pengujian pemuatan menggunakan layanan pihak ketiga, atau meneruskan informasi penerapan ke dasbor operasi kustom Anda.
- Alur kerja yang dapat dikonfigurasi — AWS CodePipeline memungkinkan Anda memodelkan berbagai tahapan proses rilis perangkat lunak Anda menggunakan antarmuka konsol, AWS AWS CLICloudFormation, atau AWS. SDKs Anda dapat dengan mudah menentukan pengujian yang akan dijalankan dan menyesuaikan langkah-langkah untuk menerapkan aplikasi Anda dan dependensinya.

Strategi penyebaran

Strategi penyebaran menentukan bagaimana Anda ingin mengirimkan perangkat lunak Anda. Organizations mengikuti strategi penyebaran yang berbeda berdasarkan model bisnis mereka. Beberapa memilih untuk memberikan perangkat lunak yang sepenuhnya diuji, dan yang lain mungkin ingin pengguna mereka memberikan umpan balik dan membiarkan pengguna mereka mengevaluasi di bawah fitur pengembangan (seperti rilis Beta). Bagian berikut membahas berbagai strategi penyebaran.

Penerapan di tempat

Dalam strategi ini, versi aplikasi sebelumnya pada setiap sumber daya komputasi dihentikan, aplikasi terbaru diinstal, dan versi baru aplikasi dimulai dan divalidasi. Hal ini memungkinkan penerapan aplikasi untuk melanjutkan dengan gangguan minimal pada infrastruktur yang mendasarinya. Dengan penerapan di tempat, Anda dapat menerapkan aplikasi Anda tanpa membuat infrastruktur baru; Namun, ketersediaan aplikasi Anda dapat terpengaruh selama penerapan ini. Pendekatan ini juga meminimalkan biaya infrastruktur dan overhead manajemen yang terkait dengan menciptakan sumber daya baru. Anda dapat menggunakan penyeimbang beban sehingga setiap instance dideregistrasi selama penerapannya dan kemudian dikembalikan ke layanan setelah penerapan selesai. Penerapan di tempat dapat all-at-once, dengan asumsi pemadaman layanan, atau dilakukan sebagai pembaruan bergulir. AWS CodeDeploy dan AWS Elastic Beanstalk menawarkan konfigurasi penerapan one-at-a-time untuk,, dan. half-at-a-time all-at-once

Deployment blue/green

Penerapan biru/hijau, kadang-kadang disebut sebagai red/black deployment, is a technique for releasing applications by shifting traffic between two identical environments running differing versions of the application. Blue/green penerapan membantu Anda meminimalkan waktu henti selama pembaruan aplikasi, mengurangi risiko seputar waktu henti dan fungsionalitas rollback.

Penerapan biru/hijau memungkinkan Anda meluncurkan versi baru (hijau) aplikasi Anda di samping versi lama (biru), dan memantau dan menguji versi baru sebelum Anda mengalihkan lalu lintas ke sana, memutar kembali pada deteksi masalah.

Penyebaran kenari

Tujuan dari penyebaran kenari adalah untuk mengurangi risiko penerapan versi baru yang berdampak pada beban kerja. Metode ini secara bertahap akan menyebarkan versi baru, membuatnya terlihat oleh pengguna baru dengan cara yang lambat. Saat Anda mendapatkan kepercayaan dalam penerapan, Anda akan menerapkannya untuk menggantikan versi saat ini secara keseluruhan.

Penyebaran linier

Penerapan linier berarti lalu lintas digeser dalam peningkatan yang sama dengan jumlah menit yang sama antara setiap kenaikan. Anda dapat memilih dari opsi linier yang telah ditentukan sebelumnya yang menentukan persentase lalu lintas yang bergeser dalam setiap kenaikan dan jumlah menit di antara setiap kenaikan.

All-at-once penyebaran

All-at-oncepenyebaran berarti semua lalu lintas digeser dari lingkungan asli ke lingkungan pengganti sekaligus.

Matriks strategi penyebaran

Matriks berikut mencantumkan strategi penerapan yang didukung untuk Amazon Elastic Container Service (Amazon ECS) AWS Lambda, dan Amazon/on-premises. EC2

- Amazon ECS adalah layanan orkestrasi yang dikelola sepenuhnya.
- AWS Lambda memungkinkan Anda menjalankan kode tanpa menyediakan atau mengelola server.
- Amazon EC2 memungkinkan Anda menjalankan kapasitas komputasi yang aman dan dapat diubah ukurannya di cloud.

Strategi penyebaran	Amazon ECS	AWS Lambda	EC2Amazon/ lokal	
Di tempat	✓	✓	✓	
Biru/hijau	✓	✓	✓*	
Kenari	✓	✓	X	
Linear	✓	✓	X	
A ll-at-once	✓	✓	X	

> **Note**
>
> Blue/green deployment with EC2/on-premis hanya berfungsi dengan EC2 instance.

AWS Elastic Beanstalk strategi penyebaran

AWS Elastic Beanstalkmendukung jenis strategi penyebaran berikut:

- A ll-at-once Melakukan penerapan di tempat pada semua instance.
- Rolling Membagi instance menjadi batch dan menyebarkan ke satu batch pada satu waktu.
- Bergulir dengan batch tambahan Membagi penerapan menjadi beberapa batch tetapi untuk batch pertama membuat instance baru alih-alih menerapkan pada EC2 instance yang ada. EC2

- Immutable Jika Anda perlu menerapkan dengan instance baru alih-alih menggunakan instance yang sudah ada.
- Pemisahan lalu lintas Melakukan penyebaran yang tidak dapat diubah dan kemudian meneruskan persentase lalu lintas ke instance baru untuk durasi waktu yang telah ditentukan sebelumnya. Jika instans tetap sehat, maka teruskan semua lalu lintas ke instance baru dan matikan instance lama.

Infrastruktur sebagai kode

Prinsip dasarnya DevOps adalah memperlakukan infrastruktur dengan cara yang sama seperti pengembang memperlakukan kode. Kode aplikasi memiliki format dan sintaks yang ditentukan. Jika kode tidak ditulis sesuai dengan aturan bahasa pemrograman, aplikasi tidak dapat dibuat. Kode disimpan dalam manajemen versi atau sistem kontrol sumber yang mencatat riwayat pengembangan kode, perubahan, dan perbaikan bug. Ketika kode dikompilasi atau dibangun ke dalam aplikasi, kami mengharapkan aplikasi yang konsisten dibuat, dan build dapat diulang dan dapat diandalkan.

Mempraktikkan infrastruktur sebagai kode berarti menerapkan ketelitian pengembangan kode aplikasi yang sama untuk penyediaan infrastruktur. Semua konfigurasi harus didefinisikan dengan cara deklaratif dan disimpan dalam sistem kontrol sumber seperti AWS CodeCommit, sama seperti kode aplikasi. Penyediaan infrastruktur, orkestrasi, dan penyebaran juga harus mendukung penggunaan infrastruktur sebagai kode.

Infrastruktur secara tradisional disediakan menggunakan kombinasi skrip dan proses manual. Terkadang skrip ini disimpan dalam sistem kontrol versi atau didokumentasikan langkah demi langkah dalam file teks atau run-book. Seringkali orang yang menulis buku run bukanlah orang yang sama yang mengeksekusi skrip ini atau mengikuti buku run-book. Jika skrip atau runbook ini tidak sering diperbarui, skrip tersebut berpotensi menjadi penghenti pertunjukan dalam penerapan. Ini menghasilkan penciptaan lingkungan baru yang tidak selalu dapat diulang, dapat diandalkan, atau konsisten.

Sebaliknya, AWS menyediakan cara yang DevOps terfokus untuk menciptakan dan memelihara infrastruktur. Mirip dengan cara pengembang perangkat lunak menulis kode aplikasi, AWS menyediakan layanan yang memungkinkan pembuatan, penyebaran, dan pemeliharaan infrastruktur dengan cara terprogram, deskriptif, dan deklaratif. Layanan ini memberikan ketelitian, kejelasan, dan keandalan. AWS Layanan yang dibahas dalam paper ini merupakan inti dari DevOps metodologi dan membentuk dasar-dasar dari berbagai prinsip dan praktik tingkat yang lebih tinggi AWS DevOps .

AWS menawarkan layanan berikut untuk mendefinisikan infrastruktur sebagai kode.

Layanan

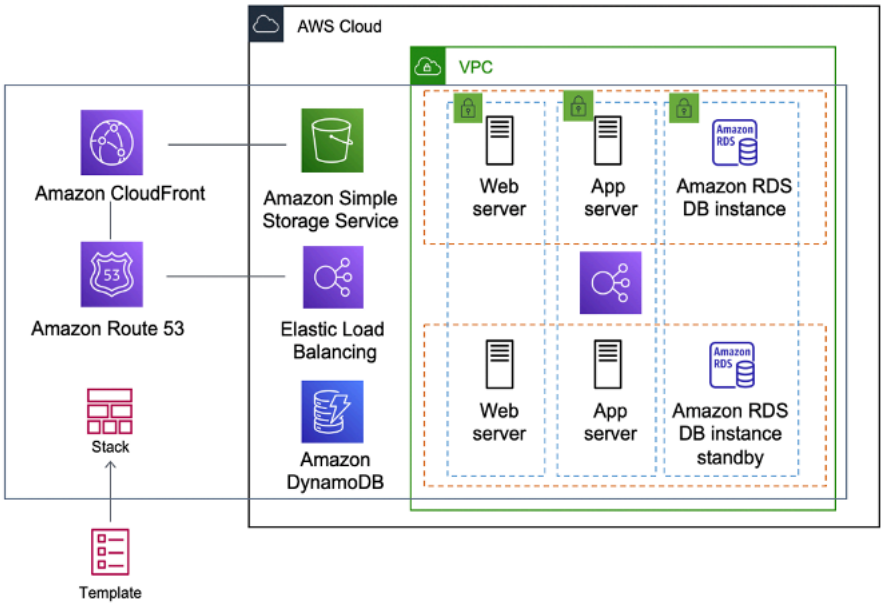
- CloudFormation
- AWS Serverless Application Model
- AWS Cloud Development Kit (AWS CDK)

- AWS Cloud Development Kit untuk Kubernetes
- AWS Cloud Development Kit untuk Terraform
- AWS Cloud Control API

CloudFormation

AWS CloudFormation adalah layanan yang memungkinkan pengembang untuk membuat AWS sumber daya secara teratur dan dapat diprediksi. Sumber daya ditulis dalam file teks menggunakan format JSON atau YAML. Templat memerlukan sebuah sintaks dan struktur tertentu sesuai dengan jenis sumber daya yang sedang dibuat dan dikelola. Anda membuat sumber daya Anda di JSON atau YAMB dengan editor kode apa pun seperti AWS Cloud9, memeriksanya ke dalam sistem kontrol versi, dan kemudian CloudFormation membangun layanan yang ditentukan dengan cara yang aman dan dapat diulang.

CloudFormation Template diterapkan ke AWS lingkungan sebagai tumpukan. Anda dapat mengelola tumpukan melalui Konsol Manajemen AWS, AWS Command Line Interface, atau CloudFormation APIs. Jika Anda perlu membuat perubahan pada sumber daya yang berjalan di tumpukan, Anda memperbarui tumpukan. Sebelum membuat perubahan pada sumber daya Anda, Anda dapat membuat kumpulan perubahan, yang merupakan ringkasan perubahan yang Anda ajukan. Set perubahan memungkinkan Anda melihat bagaimana perubahan dapat memengaruhi sumber daya yang sedang berjalan, terutama untuk sumber daya penting, sebelum menerapkannya.



AWS CloudFormation membuat seluruh lingkungan (tumpukan) dari satu template

Anda dapat menggunakan satu templat untuk membuat dan memperbarui seluruh lingkungan, atau templat terpisah untuk mengelola beberapa lapisan dalam suatu lingkungan. Hal ini memungkinkan template untuk dimodulasi, dan juga menyediakan lapisan tata kelola yang penting bagi banyak organisasi.

Saat Anda membuat atau memperbarui tumpukan di CloudFormation konsol, acara akan ditampilkan, menunjukkan status konfigurasi. Jika terjadi kesalahan, secara default tumpukan digulung kembali ke keadaan sebelumnya. Amazon SNS menyediakan pemberitahuan tentang acara. Misalnya, Anda dapat menggunakan Amazon SNS untuk melacak kemajuan pembuatan dan penghapusan tumpukan menggunakan email dan berintegrasi dengan proses lain secara terprogram.

AWS CloudFormation memudahkan untuk mengatur dan menyebarkan kumpulan AWS sumber daya, dan memungkinkan Anda menjelaskan dependensi apa pun atau meneruskan parameter khusus saat tumpukan dikonfigurasi.

Dengan CloudFormation template, Anda dapat bekerja dengan serangkaian AWS layanan yang luas, seperti Amazon S3, Auto Scaling, Amazon, Amazon DynamoDB, CloudFront Amazon, Amazon,, ELB EC2, ElastiCache IAM AWS Elastic Beanstalk OpsWorks, AWS, dan Amazon VPC. Untuk daftar sumber daya yang didukung terbaru, lihat referensi jenis AWS sumber daya dan properti.

AWS Serverless Application Model

AWS Serverless Application Model (AWS SAM) adalah kerangka kerja sumber terbuka yang dapat Anda gunakan untuk membangun aplikasi nirserver pada AWS.

AWS SAM terintegrasi dengan AWS layanan lain, sehingga membuat aplikasi tanpa server dengan AWS SAM memberikan manfaat berikut:

- Konfigurasi penerapan tunggal — AWS SAM memudahkan untuk mengatur komponen dan sumber daya terkait, dan beroperasi pada satu tumpukan. Anda dapat menggunakan AWS SAM untuk berbagi konfigurasi (seperti memori dan batas waktu) antar sumber daya, dan menyebarkan semua sumber daya terkait bersama-sama sebagai entitas tunggal berversi.
- Perpanjangan CloudFormation — Karena AWS SAM merupakan perpanjangan dari CloudFormation, Anda mendapatkan kemampuan penyebaran yang andal. CloudFormation Anda dapat menentukan sumber daya dengan menggunakan CloudFormation dalam AWS SAM template Anda.

- Praktik terbaik bawaan - Anda dapat menggunakan AWS SAM untuk menentukan dan menyebarkan IAc Anda. Hal ini memungkinkan Anda untuk menggunakan dan menerapkan praktik terbaik seperti ulasan kode.

AWS Cloud Development Kit (AWS CDK)

AWS Cloud Development Kit (AWS CDK)Ini adalah kerangka pengembangan perangkat lunak open source untuk memodelkan dan menyediakan sumber daya aplikasi cloud Anda menggunakan bahasa pemrograman yang sudah dikenal. AWS CDK memungkinkan Anda untuk memodelkan infrastruktur aplikasi menggunakan TypeScript, Python, Java, dan .NET. Pengembang dapat memanfaatkan Integrated Development Environment (IDE) yang ada, menggunakan alat seperti pelengkapan otomatis dan dokumentasi in-line untuk mempercepat pengembangan infrastruktur.

AWS CDK memanfaatkan CloudFormation di latar belakang untuk menyediakan sumber daya dengan cara yang aman dan berulang. Konstruksi adalah blok bangunan dasar kode CDK. Sebuah konstruksi mewakili komponen cloud dan merangkum semua yang CloudFormation dibutuhkan untuk membuat komponen. AWS CDK Termasuk AWS Construct Library, yang berisi konstruksi yang mewakili banyak AWS layanan. Dengan menggabungkan konstruksi bersama-sama, Anda dapat dengan cepat dan mudah membuat arsitektur kompleks untuk penerapan. AWS

AWS Cloud Development Kit untuk Kubernetes

AWS Cloud Development Kit for Kubernetes adalah kerangka kerja pengembangan perangkat lunak open-source untuk mendefinisikan aplikasi Kubernetes menggunakan bahasa pemrograman tujuan umum.

Setelah Anda mendefinisikan aplikasi Anda dalam bahasa pemrograman (pada tanggal publikasi ini, hanya Python dan TypeScript didukung), cdk8s akan mengonversi deskripsi aplikasi Anda menjadi YAMB pra-Kubernetes. File YAMB ini kemudian dapat digunakan oleh cluster Kubernetes mana pun yang berjalan di mana saja. Karena struktur didefinisikan dalam bahasa pemrograman, Anda dapat menggunakan fitur kaya yang disediakan oleh bahasa pemrograman. Anda dapat menggunakan fitur abstraksi bahasa pemrograman untuk membuat kode boilerplate Anda sendiri, dan menggunakannya kembali di semua penerapan.

AWS Cloud Development Kit untuk Terraform

Dibangun di atas perpustakaan JSII open source, CDK for Terraform (CDKTF) memungkinkan Anda untuk menulis konfigurasi Terraform sesuai pilihan C#, Python,, Java TypeScript, atau Go dan masih mendapat manfaat dari ekosistem penuh penyedia dan modul Terraform. Anda dapat mengimpor penyedia atau modul yang ada dari Terraform Registry ke dalam aplikasi Anda, dan CDKTF akan menghasilkan kelas sumber daya bagi Anda untuk berinteraksi dalam bahasa pemrograman target Anda.

Dengan CDKTF, pengembang dapat mengatur IAC mereka tanpa beralih konteks dari bahasa pemrograman yang mereka kenal, menggunakan perkakas dan sintaks yang sama untuk menyediakan sumber daya infrastruktur yang mirip dengan logika bisnis aplikasi. Tim dapat berkolaborasi dalam sintaks yang sudah dikenal, sambil tetap menggunakan kekuatan ekosistem Terraform dan menerapkan konfigurasi infrastruktur mereka melalui pipeline penerapan Terraform yang sudah ada.

AWS Cloud Control API

AWS Cloud Control APIadalah AWS kemampuan baru yang memperkenalkan satu set umum Create, Read, Update, Delete, and List (CRUDL) APIs untuk membantu pengembang mengelola infrastruktur cloud mereka dengan cara yang mudah dan konsisten. Cloud Control API yang umum APIs memungkinkan pengembang untuk mengelola siklus hidup AWS dan layanan pihak ketiga secara seragam.

Sebagai pengembang, Anda mungkin lebih suka menyederhanakan cara Anda mengelola siklus hidup semua sumber daya Anda. Anda dapat menggunakan model konfigurasi sumber daya seragam Cloud Control API dengan format yang telah ditentukan sebelumnya untuk menstandarisasi konfigurasi sumber daya cloud Anda. Selain itu, Anda akan mendapat manfaat dari perilaku API yang seragam (elemen respons dan kesalahan) saat mengelola sumber daya Anda.

Misalnya, Anda akan merasa mudah untuk men-debug kesalahan selama operasi CRUDL melalui kode kesalahan seragam yang muncul oleh Cloud Control API yang independen dari sumber daya tempat Anda beroperasi. Menggunakan Cloud Control API, Anda juga akan merasa mudah untuk mengonfigurasi dependensi lintas sumber daya. Anda juga tidak perlu lagi membuat dan memelihara kode khusus di beberapa alat vendor dan menggunakan AWS dan APIs sumber daya pihak ketiga bersama-sama.

Otomatisasi dan perkakas

Filosofi dan praktik inti lainnya DevOps adalah otomatisasi. Otomasi berfokus pada pengaturan, konfigurasi, penyebaran, dan dukungan infrastruktur dan aplikasi yang berjalan di atasnya. Dengan menggunakan otomatisasi, Anda dapat mengatur lingkungan lebih cepat dengan cara standar dan berulang. Penghapusan proses manual adalah kunci untuk DevOps strategi yang sukses. Secara historis, konfigurasi server dan penerapan aplikasi sebagian besar merupakan proses manual. Lingkungan menjadi tidak standar, dan mereproduksi lingkungan ketika masalah muncul sulit.

Penggunaan otomatisasi sangat penting untuk mewujudkan manfaat penuh dari cloud. Secara internal, AWS sangat bergantung pada otomatisasi untuk menyediakan fitur inti elastisitas dan skalabilitas.

Proses manual rawan kesalahan, tidak dapat diandalkan, dan tidak memadai untuk mendukung bisnis yang gesit. Seringkali, organisasi dapat mengikat sumber daya yang sangat terampil untuk menyediakan konfigurasi manual, ketika waktu dapat dihabiskan dengan lebih baik untuk mendukung kegiatan lain yang lebih penting, dan bernilai lebih tinggi dalam bisnis.

Lingkungan operasi modern umumnya mengandalkan otomatisasi penuh untuk menghilangkan intervensi manual atau akses ke lingkungan produksi. Ini termasuk semua pelepasan perangkat lunak, konfigurasi mesin, penambalan sistem operasi, pemecahan masalah, atau perbaikan bug. Banyak tingkat praktik otomatisasi dapat digunakan bersama untuk menyediakan proses end-to-end otomatis tingkat yang lebih tinggi.

Otomasi memiliki manfaat utama berikut:

- Perubahan cepat
- Peningkatan produktivitas
- Konfigurasi yang dapat diulang
- Lingkungan yang dapat direproduksi
- Elastisitas
- Penskalaan Otomatis
- Pengujian otomatis

Otomasi adalah landasan dengan AWS layanan dan didukung secara internal di semua layanan, fitur, dan penawaran.

Topik

- AWS OpsWorks
- AWS Elastic Beanstalk
- EC2 Image Builder
- AWS Proton
- AWS Service Catalog
- AWS Cloud9
- AWS CloudShell
- Amazon CodeGuru

AWS OpsWorks

AWS OpsWorksmengambil prinsip-prinsip DevOps bahkan lebih jauh dari AWS Elastic Beanstalk. Ini dapat dianggap sebagai layanan manajemen aplikasi daripada hanya wadah aplikasi. OpsWorks menyediakan lebih banyak tingkat otomatisasi, dengan fitur tambahan seperti integrasi dengan perangkat lunak manajemen konfigurasi (Chef) dan manajemen siklus hidup aplikasi. Anda dapat menggunakan manajemen siklus hidup aplikasi untuk menentukan kapan sumber daya disiapkan, dikonfigurasi, diterapkan, tidak digunakan, atau diakhiri.

Untuk fleksibilitas tambahan, Anda AWS OpsWorks harus menentukan aplikasi Anda dalam tumpukan yang dapat dikonfigurasi. Anda juga dapat memilih tumpukan aplikasi yang telah ditentukan. Tumpukan aplikasi berisi semua penyediaan untuk sumber daya AWS yang diperlukan aplikasi Anda, termasuk server aplikasi, server web, database, dan penyeimbang beban.

Tumpukan aplikasi diatur ke dalam lapisan arsitektur sehingga tumpukan dapat dipertahankan secara mandiri. Contoh lapisan dapat mencakup tingkat web, tingkat aplikasi, dan tingkat basis data. Di luar kotak, AWS OpsWorks juga menyederhanakan pengaturan grup AWS Auto Scaling dan penyeimbang beban Elastic Load Balancing (ELB), yang selanjutnya menggambarkan prinsip otomatisasi. DevOps Sama seperti AWS Elastic Beanstalk OpsWorks , AWS mendukung pembuatan versi aplikasi, penerapan berkelanjutan, dan manajemen konfigurasi infrastruktur

OpsWorks menampilkan DevOps fitur dan arsitektur

AWS OpsWorks juga mendukung DevOps praktik pemantauan dan pencatatan (tercakup dalam bagian berikutnya). Dukungan pemantauan disediakan oleh Amazon CloudWatch. Semua peristiwa siklus hidup dicatat, dan log Chef terpisah mendokumentasikan resep Chef apa pun yang dijalankan, bersama dengan pengecualian apa pun.

AWS Elastic Beanstalk

AWS Elastic Beanstalk adalah sebuah layanan untuk melakukan deployment dan menskalakan aplikasi web dan layanan yang dikembangkan dengan Java, NET, PHP, Node.js, Python, Ruby, Go, dan Docker pada server-server yang sudah dikenal seperti Apache, NGINX, Passenger, dan IIS.

Elastic Beanstalk adalah abstraksi di atas Amazon EC2, Auto Scaling, dan menyederhanakan penerapan dengan memberikan fitur tambahan seperti kloning, penerapan, Elastic Beanstalk Command Line Interface (EB CLI) dan integrasi dengan AWS Toolkit for Visual Studio, Visual Studio Code, Eclipse, dan IntelliSite IntelliJ untuk meningkatkan produktivitas pengembang. blue/green

EC2 Image Builder

EC2 Image Builder adalah AWS layanan terkelola penuh yang membantu Anda mengotomatiskan pembuatan, pemeliharaan, validasi, berbagi, dan penyebaran AMI kustom up-to-date Linux atau

Windows yang disesuaikan, aman, dan. EC2 Image Builder juga dapat digunakan untuk membuat gambar kontainer. Anda dapat menggunakan Konsol Manajemen AWS, AWS CLI, atau APIs untuk membuat gambar khusus di AWS akun Anda.

EC2 Image Builder secara signifikan mengurangi upaya menjaga gambar up-to-date dan keamanan dengan menyediakan antarmuka grafis sederhana, otomatisasi bawaan, dan pengaturan keamanan AWS yang disediakan. Dengan EC2 Image Builder, tidak ada langkah manual untuk memperbarui gambar dan Anda juga tidak perlu membuat pipeline otomatisasi Anda sendiri.

AWS Proton

AWS Protonmemungkinkan tim platform untuk menghubungkan dan mengoordinasikan semua alat berbeda yang dibutuhkan tim pengembangan Anda untuk penyediaan infrastruktur, penerapan kode, pemantauan, dan pembaruan. AWS Proton memungkinkan infrastruktur otomatis sebagai penyediaan kode dan penyebaran aplikasi tanpa server dan berbasis kontainer.

AWS Proton memungkinkan tim platform untuk menentukan infrastruktur dan alat penyebaran mereka, sambil memberikan pengalaman layanan mandiri kepada pengembang untuk mendapatkan infrastruktur dan menyebarkan kode. Melalui AWS Proton, tim platform menyediakan sumber daya bersama dan menentukan tumpukan aplikasi, termasuk CI/CD saluran pipa dan alat observabilitas. Anda kemudian dapat mengelola infrastruktur dan fitur penyebaran yang tersedia untuk pengembang.

AWS Service Catalog

AWS Service Catalogmemungkinkan organisasi untuk membuat dan mengelola katalog layanan TI yang disetujui. AWS Layanan TI ini dapat mencakup semuanya mulai dari gambar mesin virtual, server, perangkat lunak, basis data, dan lainnya hingga arsitektur aplikasi multi-tier yang lengkap. AWS Service Catalog memungkinkan Anda mengelola layanan, aplikasi, sumber daya, dan metadata TI yang diterapkan secara terpusat untuk mencapai tata kelola templat IAC Anda yang konsisten.

Dengan AWS Service Catalog, Anda dapat memenuhi persyaratan kepatuhan Anda sambil memastikan pelanggan Anda dapat dengan cepat menyebarkan layanan TI yang disetujui yang mereka butuhkan. Pengguna akhir dapat dengan cepat men-deploy hanya layanan IT yang disetujui yang mereka butuhkan, mengikuti batasan yang ditetapkan oleh organisasi Anda.

AWS Cloud9

AWS Cloud9adalah IDE berbasis cloud yang memungkinkan Anda menulis, menjalankan, dan men-debug kode Anda hanya dengan browser. Ini termasuk editor kode, debugger, dan terminal. AWS Cloud9 dilengkapi dengan alat penting untuk bahasa pemrograman populer, termasuk, JavaScript Python, PHP, dan banyak lagi, sehingga Anda tidak perlu menginstal file atau mengkonfigurasi mesin pengembangan Anda untuk memulai proyek baru. Karena AWS Cloud9 IDE Anda berbasis cloud, Anda dapat mengerjakan proyek dari kantor, rumah, atau di mana saja menggunakan mesin yang terhubung ke internet.

AWS CloudShell

AWS CloudShelladalah shell berbasis browser yang membuatnya lebih mudah untuk mengelola, mengeksplorasi, dan berinteraksi dengan aman dengan sumber daya Anda. AWS AWS CloudShell sudah diautentikasi sebelumnya dengan kredensil konsol Anda. Alat pengembangan dan operasi umum sudah diinstal sebelumnya, jadi tidak perlu menginstal atau mengkonfigurasi perangkat lunak pada mesin lokal Anda.

Amazon CodeGuru

Amazon CodeGuru adalah alat pengembang yang memberikan rekomendasi cerdas untuk meningkatkan kualitas kode dan mengidentifikasi baris kode aplikasi yang paling mahal. Integrasikan CodeGuru ke dalam alur kerja pengembangan perangkat lunak Anda yang ada untuk mengotomatiskan tinjauan kode selama pengembangan aplikasi dan terus memantau kinerja aplikasi dalam produksi dan memberikan rekomendasi dan petunjuk visual tentang cara meningkatkan kualitas kode, kinerja aplikasi, dan mengurangi biaya keseluruhan. CodeGuru memiliki dua komponen:

- Amazon CodeGuru Reviewer — Amazon CodeGuru Reviewer adalah layanan peninjauan kode otomatis yang mengidentifikasi cacat kritis dan penyimpangan dari praktik terbaik pengkodean untuk kode Java dan Python. Ini memindai baris kode dalam permintaan tarik dan memberikan rekomendasi cerdas berdasarkan standar yang dipelajari dari proyek sumber terbuka utama serta basis kode Amazon.
- Amazon CodeGuru Profiler — Amazon CodeGuru Profiler menganalisis profil runtime aplikasi dan memberikan rekomendasi dan visualisasi cerdas yang memandu pengembang tentang cara meningkatkan kinerja bagian paling relevan dari kode mereka.

Pemantauan dan observabilitas

Komunikasi dan kolaborasi merupakan hal mendasar dalam sebuah DevOps filosofi. Untuk memfasilitasi ini, umpan balik sangat penting. Umpan balik ini disediakan oleh rangkaian layanan pemantauan dan observabilitas kami.

AWS menyediakan layanan berikut untuk pemantauan dan pencatatan:

Topik

- CloudWatch Metrik Amazon
- CloudWatch Alarm Amazon
- CloudWatch Log Amazon
- Wawasan CloudWatch Log Amazon
- CloudWatch Acara Amazon
- Amazon EventBridge
- AWS CloudTrail
- DevOpsGuru Amazon
- AWS X-Ray
- Amazon Managed Service for Prometheus
- Amazon Managed Grafana

CloudWatch Metrik Amazon

CloudWatch Metrik Amazon secara otomatis mengumpulkan data dari AWS layanan seperti EC2 instans Amazon, volume Amazon EBS, dan instans database Amazon RDS (DB). Metrik ini kemudian dapat diatur sebagai dasbor dan alarm atau peristiwa dapat dibuat untuk memicu peristiwa atau melakukan tindakan Auto Scaling.

CloudWatch Alarm Amazon

Anda dapat mengatur alarm menggunakan CloudWatch alarm Amazon berdasarkan metrik yang dikumpulkan oleh metrik Amazon. CloudWatch Alarm kemudian dapat mengirim pemberitahuan ke topik Amazon SNS, atau memulai tindakan Auto Scaling. Alarm membutuhkan periode (lamanya

waktu untuk mengevaluasi metrik), periode evaluasi (jumlah titik data terbaru), dan titik data untuk alarm (jumlah titik data dalam periode evaluasi).

CloudWatch Log Amazon

Amazon CloudWatch Logs adalah layanan agregasi dan pemantauan log. AWS CodeBuild, CodeCommit, CodeDeploy dan CodePipeline menyediakan integrasi dengan CloudWatch log sehingga semua log dapat dipantau secara terpusat. Selain itu, layanan yang disebutkan sebelumnya berbagai AWS layanan lain menyediakan integrasi langsung dengan CloudWatch.

Dengan CloudWatch Log Anda dapat:

- Kueri data log Anda
- Pantau log dari EC2 instans Amazon
- Pantau peristiwa yang AWS CloudTrail dicatat
- Tentukan kebijakan penyimpanan log

Wawasan CloudWatch Log Amazon

Amazon CloudWatch Logs Insights memindai log Anda dan memungkinkan Anda melakukan kueri dan visualisasi interaktif. Ini memahami berbagai format log dan menemukan bidang otomatis dari log JSON.

CloudWatch Acara Amazon

Amazon CloudWatch Events memberikan aliran peristiwa sistem yang mendekati waktu nyata yang menjelaskan perubahan AWS sumber daya. Dengan menggunakan aturan sederhana yang dapat Anda siapkan dengan cepat, Anda dapat mencocokkan peristiwa dan merutekannya ke satu atau beberapa fungsi atau pengaliran target.

CloudWatch Peristiwa menjadi sadar akan perubahan operasional saat terjadi. CloudWatch Peristiwa merespons perubahan operasional ini dan mengambil tindakan korektif seperlunya, dengan mengirim pesan untuk merespons lingkungan, mengaktifkan fungsi, membuat perubahan, dan menangkap informasi negara.

Anda dapat mengonfigurasi aturan di Amazon CloudWatch Events untuk mengingatkan Anda tentang perubahan dalam AWS layanan dan mengintegrasikan peristiwa ini dengan sistem pihak ketiga

lainnya menggunakan Amazon EventBridge. Berikut ini adalah layanan AWS DevOps terkait yang memiliki integrasi dengan CloudWatch Acara.

- Acara Application Auto Scaling
- CodeBuildAcara
- CodeCommitAcara
- CodeDeploy Peristiwa
- CodePipeline Peristiwa

Amazon EventBridge

> **Note**
>
> Amazon CloudWatch Events dan EventBridge layanan dasar dan API yang sama, bagaimanapun, EventBridge menyediakan lebih banyak fitur.

Amazon EventBridge adalah bus acara tanpa server yang memungkinkan integrasi antara AWS layanan, Perangkat Lunak sebagai layanan (SaaS), dan aplikasi Anda. Selain membangun aplikasi berbasis acara, EventBridge dapat digunakan untuk memberi tahu tentang peristiwa dari layanan seperti CodeBuild,, CodeDeploy CodePipeline, dan CodeCommit.

AWS CloudTrail

Untuk merangkul DevOps prinsip-prinsip kolaborasi, komunikasi, dan transparansi, penting untuk memahami siapa yang membuat modifikasi pada infrastruktur Anda. Dalam AWS, transparansi ini disediakan oleh AWS CloudTrail. Semua AWS interaksi ditangani melalui panggilan AWS API yang dipantau dan dicatat oleh. AWS CloudTrail Semua file log yang dihasilkan disimpan dalam bucket Amazon S3 yang Anda tentukan. File log dienkripsi menggunakan enkripsi sisi server Amazon S3 (SSE). Semua panggilan API dicatat apakah mereka datang langsung dari pengguna atau atas nama pengguna oleh AWS layanan. Banyak grup dapat memperoleh manfaat dari CloudTrail log, termasuk tim operasi untuk dukungan, tim keamanan untuk tata kelola, dan tim keuangan untuk penagihan.

DevOpsGuru Amazon

Amazon DevOps Guru adalah layanan yang didukung oleh pembelajaran mesin (ML) yang dirancang untuk memudahkan peningkatan kinerja dan ketersediaan operasional aplikasi. DevOps Guru membantu mendeteksi perilaku yang menyimpang dari pola operasi normal, sehingga Anda dapat mengidentifikasi masalah operasional jauh sebelum berdampak pada pelanggan Anda.

DevOps Guru menggunakan model ML yang diinformasikan oleh tahun Amazon.com dan keunggulan AWS operasional untuk membantu mengidentifikasi perilaku aplikasi anomali (misalnya, peningkatan latensi, tingkat kesalahan, kendala sumber daya, dan lainnya) dan memunculkan masalah kritis yang dapat menyebabkan potensi pemadaman atau gangguan layanan.

Ketika DevOps Guru mengidentifikasi masalah kritis, ia menghemat waktu debugging dengan mengambil informasi yang relevan dan spesifik dari sejumlah besar sumber data dan secara otomatis mengirimkan peringatan dan memberikan ringkasan anomali terkait, dan konteks kapan dan di mana masalah terjadi.

AWS X-Ray

AWS X-Raymembantu pengembang menganalisis dan men-debug produksi, aplikasi terdistribusi, seperti yang dibangun menggunakan arsitektur microservices. Dengan X-Ray, Anda dapat memahami kinerja aplikasi dan layanan yang mendasarinya untuk mengidentifikasi dan memecahkan masalah akar penyebab masalah kinerja dan kesalahan. X-Ray memberikan end-to-end tampilan permintaan saat mereka melakukan perjalanan melalui aplikasi Anda, dan menunjukkan peta komponen dasar aplikasi Anda. X-Ray memudahkan Anda untuk:

- Buat peta layanan — Dengan melacak permintaan yang dibuat untuk aplikasi Anda, X-Ray dapat membuat peta layanan yang digunakan oleh aplikasi Anda. Ini memberi Anda tampilan koneksi antar layanan dalam aplikasi Anda, dan memungkinkan Anda membuat pohon ketergantungan, mendeteksi latensi atau kesalahan saat bekerja di seluruh AWS Availability Zone atau Regions, membidik layanan yang tidak beroperasi seperti yang diharapkan, dan sebagainya.
- Identifikasi kesalahan dan bug — X-Ray dapat secara otomatis menyorot bug atau kesalahan dalam kode aplikasi Anda dengan menganalisis kode respons untuk setiap permintaan yang dibuat untuk aplikasi Anda. Ini memungkinkan debugging kode aplikasi yang mudah tanpa mengharuskan Anda mereproduksi bug atau kesalahan.

- Buat aplikasi analisis dan visualisasi Anda sendiri — X-Ray menyediakan serangkaian kueri yang dapat APIs Anda gunakan untuk membuat aplikasi analisis dan visualisasi Anda sendiri yang menggunakan data yang direkam X-Ray.

Amazon Managed Service for Prometheus

Amazon Managed Service for Prometheus adalah layanan pemantauan tanpa server untuk metrik yang kompatibel dengan Prometheus sumber terbuka, sehingga memudahkan Anda untuk memantau dan memperingatkan lingkungan kontainer dengan aman. Amazon Managed Service untuk Prometheus mengurangi beban berat yang diperlukan untuk memulai pemantauan aplikasi di Amazon Elastic Kubernetes Service, Amazon Elastic Container Service, dan, serta cluster Kubernetes yang dikelola sendiri. AWS Fargate

Amazon Managed Grafana

Grafana yang Dikelola Amazon adalah layanan yang dikelola sepenuhnya dengan visualisasi data interaktif yang kaya untuk membantu pelanggan menganalisis, memantau, dan alarm pada metrik, log, dan jejak di berbagai sumber data. Anda dapat membuat dasbor interaktif dan membagikannya kepada siapa pun di organisasi Anda dengan layanan yang diskalakan secara otomatis, sangat tersedia, dan aman untuk perusahaan.

Komunikasi dan Kolaborasi

Apakah Anda mengadopsi DevOps Budaya di organisasi Anda atau melalui transformasi DevOps budaya, komunikasi dan kolaborasi adalah bagian penting dari pendekatan Anda. Di Amazon, kami telah menyadari bahwa ada kebutuhan untuk membawa perubahan pada pola pikir tim kami dan dengan demikian mengadopsi konsep Tim Dua Pizza.

“Kami mencoba menciptakan tim yang tidak lebih besar dari yang bisa diberi makan oleh dua pizza,” kata Bezos. “Kami menyebutnya aturan tim dua pizza.”

Semakin kecil tim, semakin baik kolaborasinya. Kolaborasi sangat penting, karena rilis perangkat lunak bergerak lebih cepat dari sebelumnya. Dan kemampuan tim untuk memberikan perangkat lunak dapat menjadi faktor pembeda bagi organisasi Anda terhadap pesaing Anda. Bayangkan situasi di mana fitur produk baru perlu dirilis atau bug perlu diperbaiki. Anda ingin ini terjadi secepat mungkin, sehingga Anda dapat memiliki go-to-market waktu yang lebih kecil. Anda tidak ingin transformasi menjadi proses yang bergerak lambat; Anda menginginkan pendekatan yang gesit di mana gelombang perubahan mulai membuat dampak.

Komunikasi antar tim juga penting saat Anda bergerak menuju model tanggung jawab bersama dan mulai bergerak keluar dari pendekatan pengembangan siloed. Ini membawa konsep kepemilikan ke tim, dan menggeser perspektif mereka untuk melihat proses sebagai sebuah end-to-end usaha. Tim Anda seharusnya tidak menganggap lingkungan produksi Anda sebagai kotak hitam di mana mereka tidak memiliki visibilitas.

Transformasi budaya juga penting, karena Anda mungkin membangun DevOps tim bersama atau memiliki anggota yang DevOps fokus dalam tim Anda. Kedua pendekatan ini memperkenalkan Tanggung Jawab Bersama ke dalam tim.

Keamanan

Apakah Anda akan melalui DevOps transformasi atau menerapkan DevOps prinsip-prinsip untuk pertama kalinya, Anda harus berpikir tentang Keamanan sebagai terintegrasi dalam DevOps proses Anda. Ini harus menjadi perhatian lintas batas di seluruh tahap penerapan pengujian build Anda.

Sebelum mengeksplorasi keamanan AWS, paper ini membahas AWS Shared Responsibility Model. DevOps

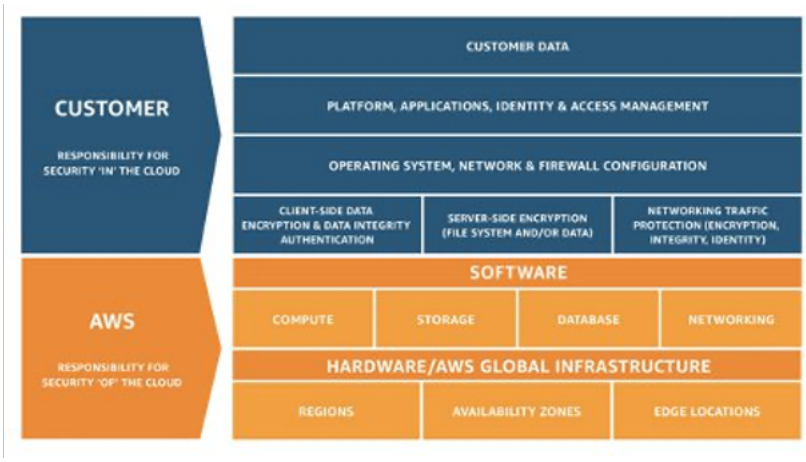
Topik

AWS Model Tanggung Jawab Bersama

Keamanan adalah tanggung jawab bersama antara AWS dan pelanggan. Bagian-bagian berbeda dari Model Tanggung Jawab Bersama adalah:

- Tanggung jawab AWS “Keamanan Cloud” - AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan semua layanan yang ditawarkan di AWS Cloud. Infrastruktur ini terdiri dari perangkat keras, perangkat lunak, jaringan, dan fasilitas yang menjalankan AWS Cloud layanan.
- Tanggung jawab pelanggan “Keamanan di Cloud” — Tanggung jawab pelanggan ditentukan oleh AWS Cloud layanan yang dipilih pelanggan. Hal ini menentukan jumlah tugas konfigurasi yang harus dilakukan pelanggan sebagai bagian dari tanggung jawab keamanan mereka.

Model bersama ini dapat membantu meringankan beban operasional pelanggan saat AWS mengoperasikan, mengelola, dan mengontrol komponen dari sistem operasi host dan lapisan virtualisasi hingga keamanan fisik fasilitas tempat layanan beroperasi. Ini sangat penting dalam kasus di mana pelanggan ingin memahami keamanan lingkungan build mereka.



Model Tanggung Jawab Bersama AWS

Untuk DevOps, tetapkan izin berdasarkan model izin hak istimewa paling sedikit. Model ini menyatakan bahwa “pengguna (atau layanan) harus memiliki hak akses yang tepat yang diperlukan untuk menyelesaikan tanggung jawab peran mereka—tidak lebih, tidak kurang.”

Izin dipertahankan di IAM. Anda dapat menggunakan IAM untuk mengontrol siapa yang diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya.

Identity and Access Management

AWS Identity and Access Management(IAM) mendefinisikan kontrol dan kebijakan yang digunakan untuk mengelola akses ke sumber daya. AWS Menggunakan IAM Anda dapat membuat pengguna dan grup dan menentukan izin untuk berbagai DevOps layanan.

Selain pengguna, berbagai layanan mungkin juga memerlukan akses ke AWS sumber daya. Misalnya, CodeBuild proyek Anda mungkin memerlukan akses untuk menyimpan gambar Docker di Amazon Elastic Container Registry (Amazon ECR) dan memerlukan izin untuk menulis ke Amazon ECR. Jenis izin ini ditentukan oleh peran tipe khusus yang dikenal sebagai peran layanan.

IAM adalah salah satu komponen infrastruktur AWS keamanan. Dengan IAM, Anda dapat mengelola grup, pengguna, peran layanan, dan kredensi keamanan secara terpusat seperti kata sandi, kunci akses, dan kebijakan izin yang mengontrol layanan AWS dan sumber daya yang dapat diakses pengguna. Kebijakan IAM memungkinkan Anda menentukan kumpulan izin. Kebijakan ini kemudian dapat dilampirkan ke peran, pengguna, atau layanan untuk menentukan izin mereka.

Anda juga dapat menggunakan IAM untuk membuat peran yang digunakan secara luas dalam DevOps strategi yang Anda inginkan. Dalam beberapa kasus, masuk akal untuk secara terprogram

AssumeRolealih-alih langsung mendapatkan izin. Ketika layanan atau pengguna mengambil peran, mereka diberikan kredensi sementara untuk mengakses layanan yang biasanya tidak dapat mereka akses.

Kesimpulan

Untuk membuat perjalanan ke cloud lancar, efisien, dan efektif, perusahaan teknologi harus merangkul DevOps prinsip dan praktik. Prinsip-prinsip ini tertanam dalam AWS, dan membentuk landasan berbagai AWS layanan, terutama yang ada dalam penawaran penyebaran dan pemantauan.

Mulailah dengan mendefinisikan infrastruktur Anda sebagai kode menggunakan layanan AWS CloudFormation atau AWS CDK. Selanjutnya, tentukan cara aplikasi Anda akan menggunakan penerapan berkelanjutan dengan bantuan layanan seperti AWS CodeBuild,, AWS CodeDeploy AWS CodePipeline, dan AWS CodeCommit. Pada tingkat aplikasi, gunakan wadah seperti, Amazon ECS AWS Elastic Beanstalk, atau Amazon Elastic Kubernetes Service (Amazon EKS). Gunakan OpsWorks untuk menyederhanakan konfigurasi arsitektur umum. Menggunakan layanan ini juga memudahkan untuk memasukkan layanan penting lainnya seperti Auto Scaling dan ELB.

Terakhir, gunakan DevOps strategi pemantauan seperti Amazon CloudWatch, dan praktik keamanan yang solid seperti IAM.

Dengan AWS sebagai mitra Anda, DevOps prinsip Anda membawa kelincahan ke bisnis dan organisasi TI Anda dan mempercepat perjalanan Anda ke cloud.

Revisi Dokumen

Untuk mengetahui jika ada perubahan pada laporan resmi ini, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Diperbarui	Diperbarui	April 7, 2023
Bagian yang diperbarui untuk menyertakan layanan baru	Bagian yang diperbarui untuk menyertakan layanan baru	16 Oktober 2020
Publikasi awal	Pertama kali laporan resmi dipublikasikan	Desember 1, 2014

Kontributor

Para kontributor untuk dokumen ini antara lain:

- Abhra Sinha, Arsitek Solusi
- Anil Nadiminti, Arsitek Solusi
- Muhammad Mansoor, Arsitek Solusi
- Ajit Zadgaonkar, Pemimpin Teknologi Seluruh Dunia, Modernisasi
- Juan Lamadrid, Arsitek Solusi
- Darren Ball, Arsitek Solusi
- Rajeswari Malladi, Arsitek Solusi
- Pallavi Nargund, Arsitek Solusi
- Bert Zahniser, Arsitek Solusi
- Abdullahi Olaoye, Arsitek Solusi Cloud
- Mohamed Kiswani, Manajer Pengembangan Perangkat Lunak
- Tara McCann, Manajer, Arsitek Solusi

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili penawaran dan praktik produk AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dari AWS dan afiliasinya, pemasok, atau pemberi lisensinya. Produk atau layanan AWS disediakan “sebagaimana adanya” tanpa jaminan, pernyataan, atau ketentuan dalam bentuk 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.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.