AWS Whitepaper

Ikhtisar Opsi Penerapan di AWS

Ikhtisar Opsi Penerapan di AWS: AWS Whitepaper

Table of Contents

Ikhtisar Opsi Penerapan di AWS

Tanggal publikasi: 31 Mei 2024 () Revisi Dokumen

Abstrak

Amazon Web Services (AWS) menawarkan beberapa opsi untuk penyediaan infrastruktur dan penerapan aplikasi Anda. Baik arsitektur aplikasi Anda adalah aplikasi web tiga tingkat sederhana atau serangkaian beban kerja yang kompleks, AWS menawarkan layanan penerapan untuk memenuhi persyaratan aplikasi dan organisasi Anda.

Whitepaper ini ditujukan untuk individu yang mencari ikhtisar tentang berbagai layanan penerapan yang ditawarkan oleh AWS. Ini menjabarkan fitur umum yang tersedia dalam layanan penyebaran ini, dan mengartikulasikan strategi dasar untuk menyebarkan dan memperbarui tumpukan aplikasi.

Pengantar

Merancang solusi penerapan untuk aplikasi Anda adalah bagian penting dalam membangun aplikasi yang dirancang dengan baik di AWS. Berdasarkan sifat aplikasi Anda dan layanan dasar yang diperlukan, Anda dapat menggunakan layanan AWS untuk membuat solusi penerapan fleksibel yang dapat disesuaikan dengan kebutuhan aplikasi dan organisasi Anda.

Katalog layanan AWS yang terus berkembang tidak hanya mempersulit proses memutuskan layanan mana yang akan menyusun arsitektur aplikasi Anda, tetapi juga proses memutuskan bagaimana Anda akan membuat, mengelola, dan memperbarui aplikasi Anda. Saat merancang solusi penerapan di AWS, Anda harus mempertimbangkan bagaimana solusi Anda akan menangani kemampuan berikut:

- Penyediaan - Buat infrastruktur mentah atau infrastruktur layanan terkelola yang diperlukan untuk aplikasi Anda.
- Konfigurasi - Sesuaikan infrastruktur Anda berdasarkan lingkungan, runtime, keamanan, ketersediaan, kinerja, jaringan, atau persyaratan aplikasi lainnya.
- Deploy - Instal atau perbarui komponen aplikasi Anda ke sumber daya infrastruktur dan kelola transisi dari versi aplikasi sebelumnya ke versi aplikasi baru.
- Skala - Secara proaktif atau reaktif menyesuaikan jumlah sumber daya yang tersedia untuk aplikasi Anda berdasarkan serangkaian kriteria yang ditentukan pengguna.
- Monitor - Memberikan visibilitas ke sumber daya yang diluncurkan sebagai bagian dari arsitektur aplikasi Anda. Lacak penggunaan sumber daya, keberhasilan atau kegagalan penerapan, kesehatan aplikasi, log aplikasi, penyimpangan konfigurasi, dan banyak lagi.

Whitepaper ini menyoroti layanan penerapan yang ditawarkan oleh AWS dan menguraikan strategi untuk merancang arsitektur penerapan yang sukses untuk semua jenis aplikasi.

Layanan Penerapan AWS

Tugas merancang solusi penyebaran yang dapat diskalakan, efisien, dan hemat biaya tidak boleh terbatas pada bagaimana Anda akan memperbarui versi aplikasi Anda, tetapi juga harus mempertimbangkan bagaimana Anda akan mengelola infrastruktur pendukung di seluruh siklus hidup aplikasi yang lengkap. Penyediaan sumber daya, manajemen konfigurasi, penyebaran aplikasi, pembaruan perangkat lunak, pemantauan, kontrol akses, dan masalah lainnya adalah semua faktor penting yang perlu dipertimbangkan saat merancang solusi penerapan.

Layanan AWS dapat menyediakan kemampuan manajemen untuk satu atau beberapa aspek siklus hidup aplikasi Anda. Bergantung pada keseimbangan kontrol yang Anda inginkan (manajemen sumber daya secara manual) versus kenyamanan (manajemen sumber daya AWS) dan jenis aplikasi, layanan ini dapat digunakan sendiri atau digabungkan untuk membuat solusi penyebaran yang kaya fitur. Bagian ini akan memberikan gambaran umum tentang layanan AWS yang dapat digunakan untuk memungkinkan organisasi membangun dan mengirimkan aplikasi dengan lebih cepat dan andal.

AWS CloudFormation

AWS CloudFormationadalah layanan yang memungkinkan pelanggan untuk menyediakan dan mengelola hampir semua sumber daya AWS menggunakan bahasa templat khusus yang diekspresikan dalam YAMG atau JSON. AWS CloudFormation Template membuat sumber daya infrastruktur dalam grup yang disebut tumpukan, dan memungkinkan Anda untuk menentukan dan menyesuaikan semua komponen yang diperlukan untuk mengoperasikan aplikasi Anda sambil mempertahankan kontrol penuh atas sumber daya ini. Menggunakan template memperkenalkan kemampuan untuk menerapkan kontrol versi pada infrastruktur Anda, dan kemampuan untuk mereplikasi infrastruktur Anda dengan cepat dan andal.

AWS CloudFormation menawarkan kontrol granular atas penyediaan dan pengelolaan semua komponen infrastruktur aplikasi, dari komponen tingkat rendah seperti tabel rute atau konfigurasi subnet, hingga komponen tingkat tinggi seperti distribusi. CloudFront AWS CloudFormation umumnya digunakan dengan layanan penerapan AWS lainnya atau alat pihak ketiga, digabungkan AWS CloudFormation dengan layanan penyebaran yang lebih khusus untuk mengelola penerapan kode aplikasi ke komponen infrastruktur.

AWS menawarkan ekstensi ke CloudFormation layanan selain fitur dasarnya:

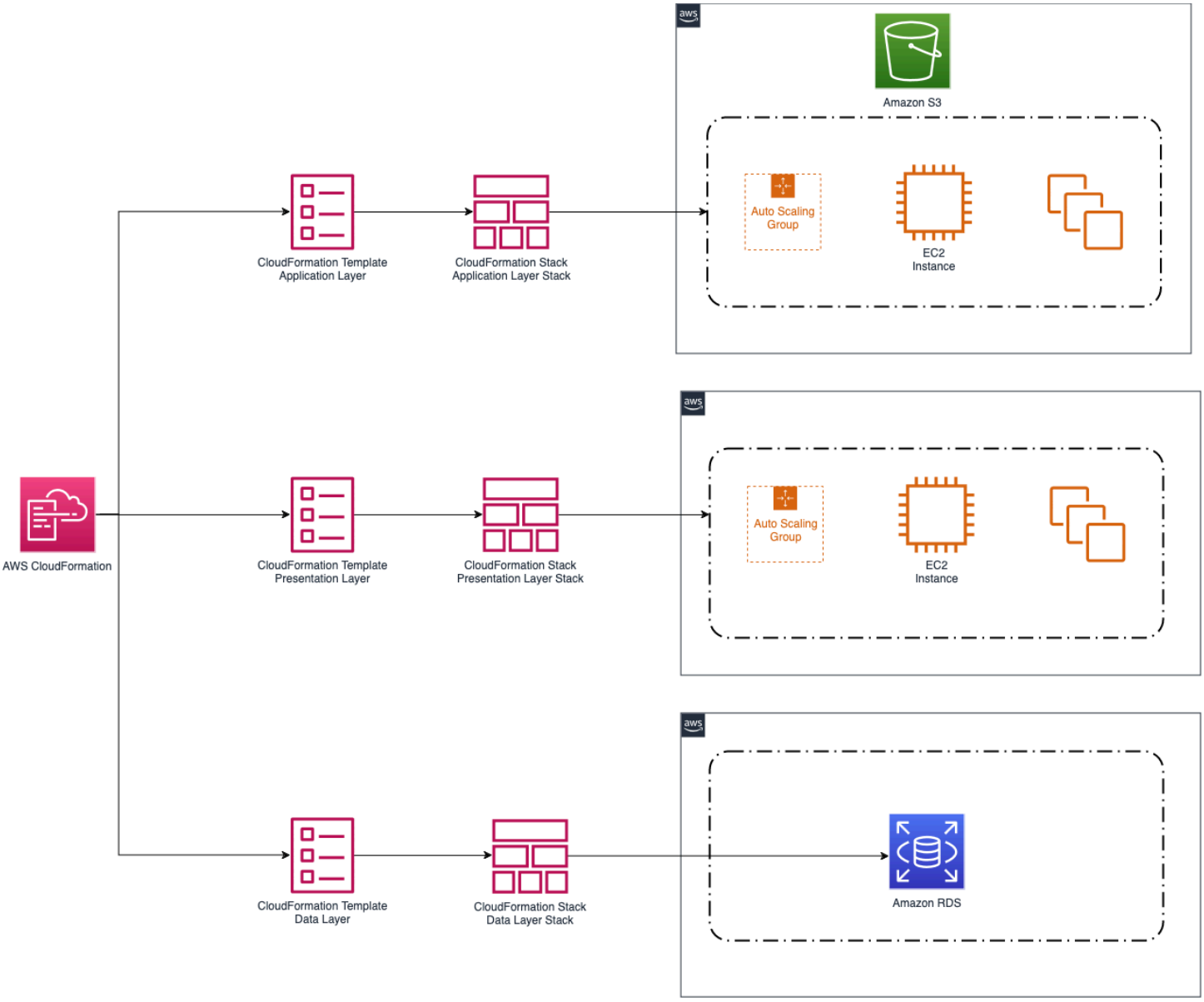
- AWS Cloud Development Kit (AWS CDK)adalah kit pengembangan perangkat lunak open source (SDK) untuk memodelkan infrastruktur AWS secara terprogram dengan TypeScript,, JavaScript Python, Java, atau C#/.NET.
- AWS Serverless Application Model(AWS SAM) adalah kerangka kerja open source untuk menyederhanakan pembuatan aplikasi tanpa server di AWS. Ini menyediakan sintaks singkatan untuk mengekspresikan fungsi, database APIs, dan pemetaan sumber peristiwa.

Tabel 1: AWS CloudFormation fitur penyebaran

Kemampuan	Deskripsi
Ketentuan	CloudFormation akan secara otomatis membuat dan memperbarui komponen infrastru ktur yang didefinisikan dalam template. Lihat Praktik AWS CloudFormation Terbaik untuk detail selengkapnya tentang membuat infrastruktur menggunakan AWS CloudForm ation templat.
Konfigurasi	AWS CloudFormation template menawarkan fleksibilitas yang luas untuk menyesuaikan dan memperbarui semua komponen infrastruktur. Lihat Anatomi AWS CloudFormation Template untuk detail lebih lanjut tentang menyesuaikan templat.
Deploy	Perbarui AWS CloudFormation template Anda untuk mengubah sumber daya dalam tumpukan. Bergantung pada arsitektur aplikasi Anda, Anda mungkin memerlukan layanan penyebaran tambahan untuk memperbarui versi aplikasi yang berjalan pada infrastruktur Anda. Lihat Menerapkan Aplikasi di Amazon EC2 dengan AWS CloudFormation detail selengkap

Kemampuan	Deskripsi
	nya tentang bagaimana AWS CloudFormation dapat digunakan sebagai solusi penerapan.
Penskalaan	AWS CloudFormation tidak akan secara otomatis menangani penskalaan infrastru ktur atas nama Anda; namun, Anda dapat mengonfigurasi kebijakan penskalaan otomatis untuk sumber daya Anda dalam templat AWS CloudFormation .
Memantau	AWS CloudFormation menyediakan pemantaua n asli atas keberhasilan atau kegagalan pembaruan infrastruktur yang ditentukan dalam templat, serta deteksi drift untuk memantau ketika sumber daya yang ditentukan dalam templat tidak memenuhi spesifikasi. Solusi pemantauan tambahan perlu tersedia untuk pemantauan dan metrik tingkat aplikasi. Lihat Memantau Kemajuan Pembaruan Tumpukan untuk detail selengkapnya tentang cara AWS CloudFormation memantau pembaruan infrastruktur.

Diagram berikut menunjukkan kasus penggunaan umum untuk AWS CloudFormation. Di sini, AWS CloudFormation template dibuat untuk menentukan semua komponen infrastruktur yang diperlukan untuk membuat aplikasi web tiga tingkat sederhana. Dalam contoh ini, kami menggunakan skrip bootstrap yang didefinisikan AWS CloudFormation untuk menyebarkan versi terbaru aplikasi kami ke EC2 instance Amazon; Namun, ini juga merupakan praktik umum untuk menggabungkan layanan penyebaran tambahan dengan AWS CloudFormation (AWS CloudFormation hanya menggunakan untuk manajemen infrastruktur dan kemampuan penyediaannya). Perhatikan bahwa lebih dari satu AWS CloudFormation template digunakan untuk membuat infrastruktur. Dalam diagram, AWS CloudFormation digunakan untuk membuat semua komponen infrastruktur termasuk peran IAM, subnet VPCs, tabel rute, grup keamanan, dan kebijakan bucket Amazon S3. AWS CloudFormation Template terpisah digunakan untuk membangun setiap domain arsitektur aplikasi.



AWS CloudFormation kasus penggunaan

AWS Elastic Beanstalk

AWS Elastic Beanstalkadalah easy-to-use layanan untuk menyebarkan dan menskalakan aplikasi dan layanan web yang dikembangkan dengan Java, .NET, .NET Core, PHP, Node.js, Python, Ruby, Go, atau Docker pada server yang sudah dikenal seperti Apache, Nginx, Passenger, dan IIS. Elastic Beanstalk adalah solusi manajemen aplikasi yang lengkap, dan mengelola semua tugas infrastruktur dan platform atas nama Anda.

Dengan Elastic Beanstalk, Anda dapat dengan cepat menyebarkan, mengelola, dan menskalakan aplikasi tanpa beban operasional dalam mengelola infrastruktur. Elastic Beanstalk mengurangi kompleksitas manajemen untuk aplikasi web, menjadikannya pilihan yang baik untuk organisasi yang baru mengenal AWS atau ingin menerapkan aplikasi web secepat mungkin.

Saat menggunakan Elastic Beanstalk sebagai solusi penyebaran Anda, cukup unggah kode sumber Anda dan Elastic Beanstalk akan menyediakan dan mengoperasikan semua infrastruktur yang diperlukan, termasuk server, database, penyeimbang beban, jaringan, dan grup penskalaan otomatis. Meskipun sumber daya ini dibuat atas nama Anda, Anda tetap memegang kendali penuh atas sumber daya ini, memungkinkan pengembang untuk menyesuaikan sesuai kebutuhan. Elastic Beanstalk memenuhi kriteria kepatuhan ISO, PCI, SOC 1, SOC 2, dan SOC 3 beserta kriteria kelayakan HIPAA. Ini berarti aplikasi yang berjalan pada Elastic Beanstalk dapat memproses data keuangan yang diatur atau informasi kesehatan yang dilindungi (PHI).

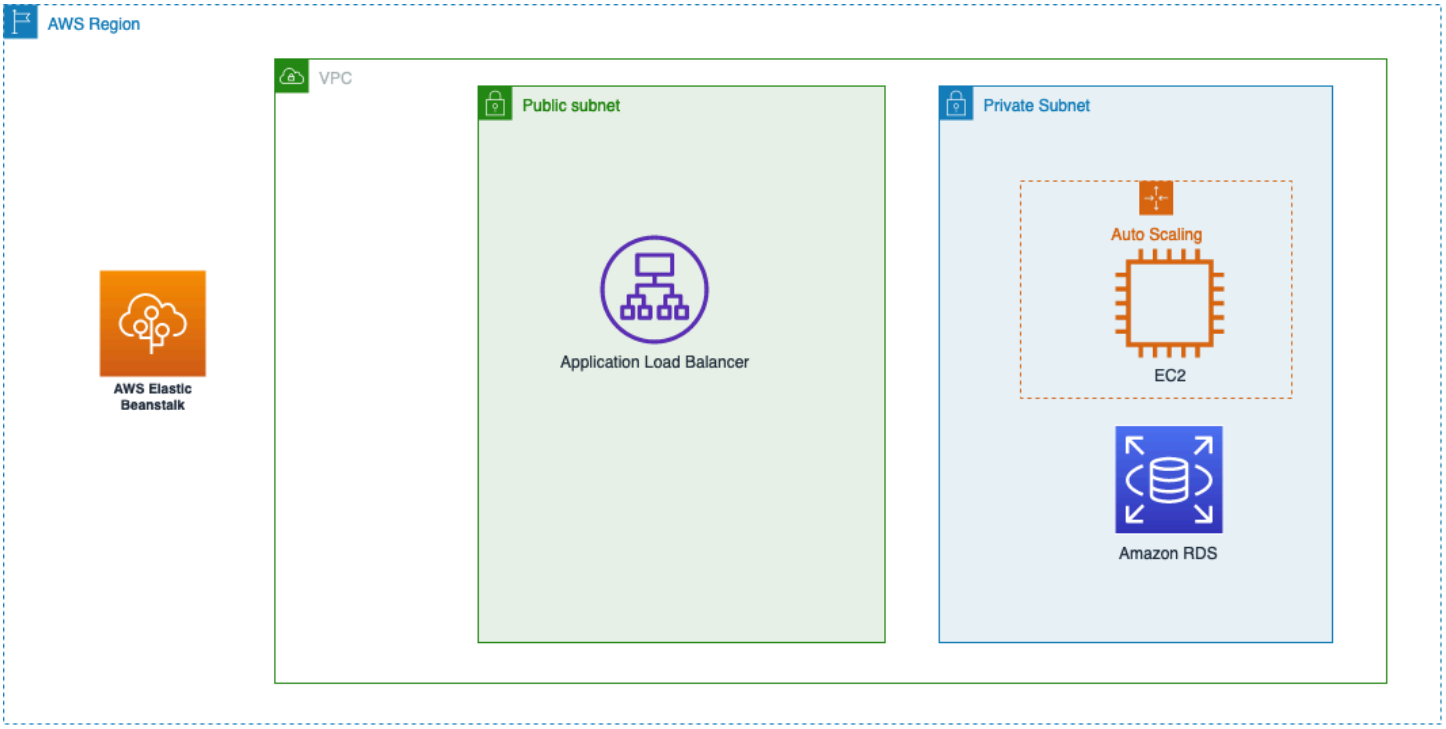
Tabel 2: AWS Elastic Beanstalk Fitur Penerapan

Kemampuan	Deskripsi
Ketentuan	Elastic Beanstalk akan menciptakan semua komponen infrastruktur yang diperlukan untuk mengoperasikan aplikasi web atau layanan yang berjalan pada salah satu platform yang didukung. Jika Anda membutuhkan infrastru ktur tambahan, ini harus dibuat di luar Elastic Beanstalk. Lihat Platform Elastic Beanstalk untuk detail lebih lanjut tentang platform aplikasi web yang didukung oleh Elastic Beanstalk.
Konfigurasi	Elastic Beanstalk menyediakan berbagai pilihan untuk menyesuaikan sumber daya di lingkunga n Anda. Lihat Mengonfigurasi lingkungan Elastic Beanstalk untuk informasi lebih lanjut tentang menyesuaikan sumber daya yang dibuat oleh Elastic Beanstalk.

Kemampuan	Deskripsi
Deploy	Elastic Beanstalk secara otomatis menangani penerapan aplikasi, dan menciptakan lingkungan yang menjalankan versi baru aplikasi Anda tanpa memengaruhi pengguna yang sudah ada. Lihat Deploying Applications AWS Elastic Beanstalk untuk detail selengkapnya tentang penerapan aplikasi dengan Elastic Beanstalk.
Penskalaan	Elastic Beanstalk menggunakan Elastic Load Balancing dan Auto Scaling untuk secara otomatis menskalakan aplikasi Anda berdasarkan kebutuhan spesifiknya. Beberapa zona ketersediaan memberi Anda opsi untuk meningkatkan keandalan dan ketersediaan aplikasi. Lihat Auto Scaling Group untuk Lingkungan Elastic Beanstalk Anda untuk detail selengkapnya tentang penskalaan otomatis dengan Elastic Beanstalk.
Memantau	Elastic Beanstalk menawarkan pemantauan lingkungan bawaan untuk aplikasi termasuk keberhasilan/kegagalan penerapan, kesehatan lingkungan, kinerja sumber daya, dan log aplikasi. Lihat Monitoring an Environment untuk detail lebih lanjut tentang pemantauan full-stack dengan Elastic Beanstalk.

Kemampuan	Deskripsi
Dukungan graviton	Prosesor berbasis AWS Graviton arm64 memberikan kinerja harga terbaik untuk beban kerja cloud Anda yang berjalan di Amazon. EC2 Dengan AWS Graviton di Elastic Beanstalk, Anda dapat memilih jenis instans EC2 Amazon untuk memenuhi kebutuhan pengoptimalan beban kerja Anda dan mendapatkan keuntungan dari peningkatan kinerja harga dibandingkan prosesor berbasis x86 yang sebanding.

Elastic Beanstalk memudahkan aplikasi web untuk digunakan dan dikelola dengan cepat di AWS. Contoh berikut menunjukkan kasus penggunaan umum untuk Elastic Beanstalk karena digunakan untuk menyebarkan aplikasi web sederhana. Semua infrastruktur aplikasi (termasuk grup keamanan, peran IAM, dan CloudWatch alarm) dibuat dan dikelola oleh Elastic Beanstalk. EC2 Instans Amazon secara otomatis disediakan dengan lingkungan runtime dan paket penerapan. Lingkungan Elastic Beanstalk dapat berintegrasi dengan sumber daya seperti Amazon Relational Database Service (Amazon RDS) yang dibuat di luar Elastic Beanstalk.



AWS Elastic Beanstalk kasus penggunaan

AWS CodeDeploy

AWS CodeDeployadalah layanan penyebaran terkelola penuh yang mengotomatiskan penerapan aplikasi untuk menghitung layanan seperti Amazon, Amazon Elastic Container Service (EC2Amazon ECS), atau server lokal. AWS Lambda Organizations dapat digunakan CodeDeploy untuk mengotomatiskan penerapan aplikasi dan menghapus operasi manual rawan kesalahan dari proses penyebaran. CodeDeploy dapat digunakan dengan berbagai macam konten aplikasi termasuk kode, fungsi tanpa server, file konfigurasi, dan banyak lagi.

CodeDeploy dimaksudkan untuk digunakan sebagai layanan blok bangunan yang berfokus pada membantu pengembang aplikasi menyebarkan dan memperbarui perangkat lunak yang berjalan pada infrastruktur yang ada. Ini bukan solusi manajemen end-to-end aplikasi, dan dimaksudkan untuk digunakan bersama dengan layanan penerapan AWS lainnya seperti AWS CodeStar, Alat Pengembang AWS lainnya AWS CodePipeline, dan layanan pihak ketiga (lihat Integrasi AWS CodeDeploy Produk untuk daftar lengkap integrasi produk) sebagai bagian dari pipeline CI/CD lengkap. Selain itu, CodeDeploy tidak mengelola pembuatan sumber daya atas nama pengguna.

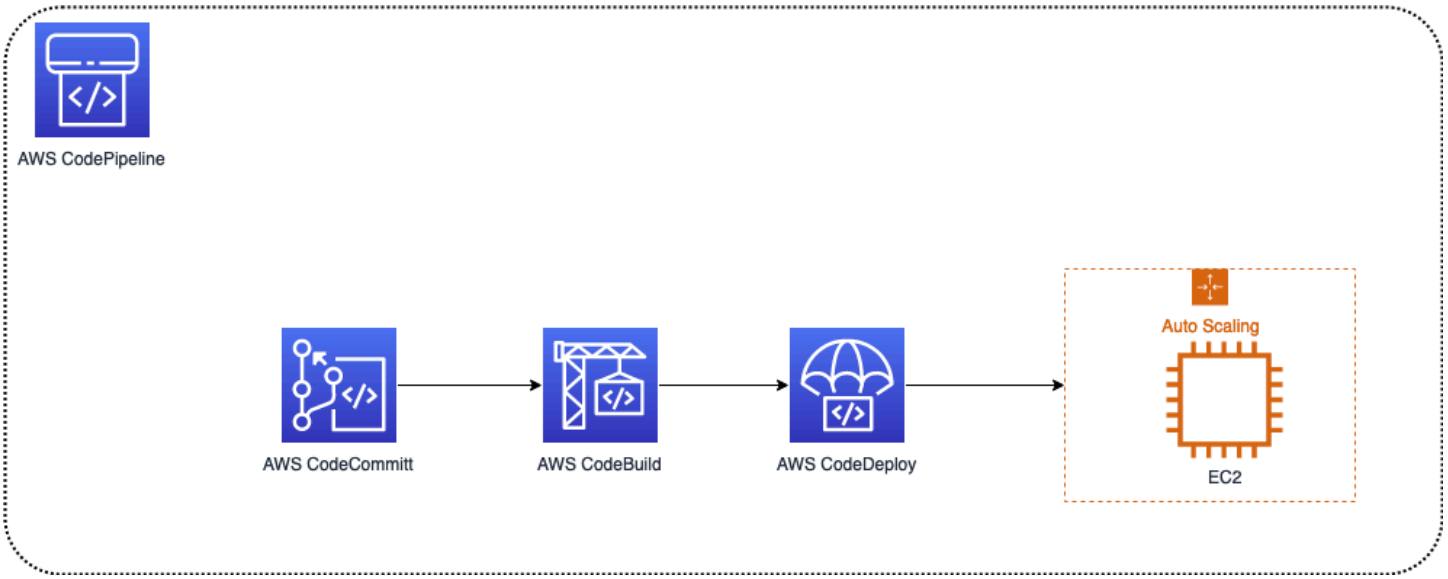
Tabel 3: AWS CodeDeploy fitur penyebaran

Kemampuan	Deskripsi
Ketentuan	CodeDeploy dimaksudkan untuk digunakan dengan sumber daya komputasi yang ada dan tidak membuat sumber daya atas nama Anda. CodeDeploy membutuhkan sumber daya komputasi untuk diatur ke dalam konstruksi yang disebut grup penyebaran untuk menyebarkan konten aplikasi. Lihat Bekerja dengan Grup Penerapan CodeDeploy untuk detail selengkapnya tentang penautan CodeDeploy ke sumber daya komputasi.

Kemampuan	Deskripsi
Konfigurasi	CodeDeploy menggunakan file spesifikasi aplikasi untuk menentukan kustomisasi untuk sumber daya komputasi. Lihat Referensi CodeDeploy AppSpec File untuk detail lebih lanjut tentang kustomisasi sumber daya dengan. CodeDeploy
Deploy	Bergantung pada jenis sumber daya komputasi yang CodeDeploy digunakan, CodeDeplo y menawarkan strategi berbeda untuk menerapkan aplikasi Anda. Lihat Bekerja dengan Penerapan di CodeDeplo y untuk detail selengkapnya tentang jenis proses penerapan yang didukung.
Penskalaan	CodeDeploy tidak mendukung penskalaa n infrastruktur aplikasi dasar Anda; namun, tergantung pada konfigurasi penerapan Anda, itu mungkin membuat sumber daya tambahan untuk mendukung penerapan biru/hijau.
Memantau	CodeDeploy dapat memantau keberhasilan atau kegagalan penerapan dan menawarka n riwayat semua penerapan, tetapi tidak memberikan metrik kinerja atau tingkat aplikasi. Lihat Monitoring Deployment di CodeDeplo y untuk detail lebih lanjut tentang jenis kemampuan pemantauan yang ditawarkan oleh CodeDeploy

Diagram berikut menggambarkan kasus penggunaan umum CodeDeploy sebagai bagian dari solusi CI/CD lengkap. Dalam contoh ini, CodeDeploy digunakan bersama dengan Alat Pengembang AWS tambahan, yaitu AWS CodePipeline (mengotomatiskan pipeline CI/CD), AWS

CodeBuild(membangun dan menguji komponen aplikasi), dan AWS CodeCommit(repositori kode sumber) untuk menyebarkan aplikasi ke sekelompok instance Amazon. EC2 CodeDeploy digunakan dengan alat lain sebagai bagian dari pipa CI/CD lengkap. CodeDeploy mengelola penyebaran komponen aplikasi ke sumber daya komputasi yang merupakan bagian dari grup penyebaran. Semua komponen infrastruktur dibuat di luar CodeDeploy.



AWS CodeDeploy kasus penggunaan

AWS CodeDeploy untuk AWS Lambda

AWS CodeDeploy for AWS Lambda memungkinkan Anda mengotomatiskan penerapan tanpa server Anda, memberi Anda kontrol dan visibilitas yang lebih besar atas rilis aplikasi Anda. Anda dapat menggunakan CodeDeploy untuk menyebarkan versi baru fungsi tanpa server Anda ke sebagian kecil pengguna atau lalu lintas dan secara bertahap meningkatkan lalu lintas saat Anda mendapatkan kepercayaan pada versi baru. Dengan CodeDeploy, Anda dapat menentukan grup penerapan, yang mewakili sekumpulan fungsi Lambda yang menerima lalu lintas dari sumber peristiwa yang sama. Misalnya, Anda dapat membuat grup penerapan untuk sekumpulan fungsi Lambda yang diprakarsai oleh API Gateway atau aturan Amazon. EventBridge Anda kemudian dapat membuat penerapan menggunakan CodeDeploy, yang menyebarkan versi baru fungsi erverless Anda ke grup penerapan tertentu.

CodeDeploy juga memungkinkan Anda untuk menentukan konfigurasi penerapan, yang menentukan pengaturan untuk penerapan, seperti jenis penerapan, strategi penerapan, dan aturan pemindahan lalu lintas. Anda dapat menggunakan strategi penyebaran Canary untuk menyebarkan versi baru fungsi tanpa server Anda ke sebagian kecil lalu lintas dan memantau kesehatan dan kinerja versi baru sebelum meningkatkan lalu lintas ke sana.

Dengan menggunakan CodeDeploy for serverless, Anda dapat mengotomatiskan proses penerapan, mengurangi waktu dan upaya yang diperlukan untuk merilis versi baru aplikasi Anda, dan meningkatkan stabilitas dan keandalan fungsi tanpa server Anda.

Amazon Elastic Container Service

Amazon Elastic Container Service (Amazon ECS) adalah layanan orkestrasi kontainer yang dikelola sepenuhnya yang mendukung kontainer Docker dan memungkinkan Anda menjalankan aplikasi dengan mudah di cluster terkelola. Amazon ECS menghilangkan kebutuhan untuk menginstal, mengoperasikan, dan menskalakan infrastruktur manajemen kontainer, dan menyederhanakan pembuatan lingkungan dengan fitur inti AWS yang sudah dikenal seperti Grup Keamanan, Elastic Load Balancing, AWS Identity and Access Managementdan (IAM).

Saat menjalankan aplikasi di Amazon ECS, Anda dapat memilih untuk menyediakan daya komputasi yang mendasari container Anda dengan EC2 instans Amazon atau dengan AWS Fargate, mesin komputasi tanpa server untuk kontainer. Dalam kedua kasus tersebut, Amazon ECS secara otomatis menempatkan dan menskalakan container Anda ke klaster Anda sesuai dengan konfigurasi yang ditentukan oleh pengguna. Meskipun Amazon ECS tidak membuat komponen infrastruktur seperti Load Balancers atau peran IAM atas nama Anda, layanan Amazon ECS menyediakan sejumlah APIs untuk menyederhanakan pembuatan dan penggunaan sumber daya ini dalam cluster Amazon ECS.

Amazon ECS memungkinkan pengembang untuk memiliki kontrol langsung dan halus atas semua komponen infrastruktur, memungkinkan pembuatan arsitektur aplikasi khusus. Selain itu, Amazon ECS mendukung strategi penerapan yang berbeda untuk memperbarui gambar wadah aplikasi Anda.

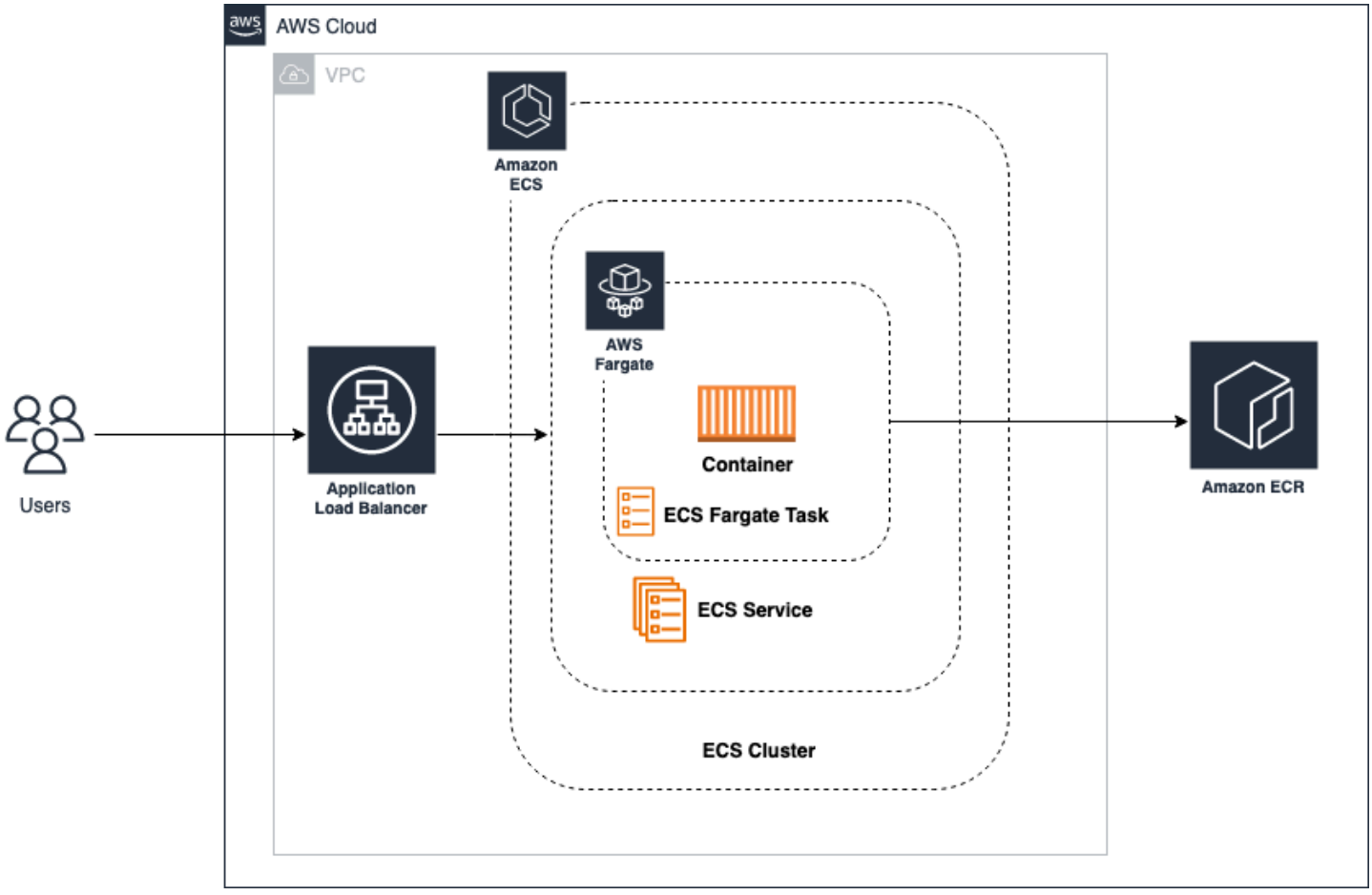
Tabel 4: Fitur penyebaran Amazon ECS

Kemampuan	Deskripsi
Ketentuan	Amazon ECS akan menyediakan instans penampung aplikasi baru dan menghitung sumber daya berdasarkan kebijakan penskalaan dan konfigurasi Amazon ECS. Sumber daya infrastruktur seperti Load Balancer perlu dibuat di luar Amazon ECS.

Kemampuan	Deskripsi
	Lihat Memulai Amazon ECS untuk detail selengkapnya tentang jenis sumber daya yang dapat dibuat dengan Amazon ECS.
Konfigurasi	Amazon ECS mendukung penyesuaian sumber daya komputasi yang dibuat untuk menjalankan aplikasi kontainer, serta kondisi runtime wadah aplikasi (misalnya, variabel lingkungan, port terbuka, memori cadangan/CPU). Kustomisasi sumber daya komputasi yang mendasarinya hanya tersedia jika menggunakan EC2 instans Amazon. Lihat Membuat Cluster untuk detail selengkapnya tentang cara menyesuaikan klaster Amazon ECS untuk menjalankan aplikasi kontainer.
Deploy	Amazon ECS mendukung beberapa strategi penerapan untuk aplikasi kontainer Anda. Lihat Jenis Penerapan Amazon ECS untuk detail selengkapnya tentang jenis proses penerapan yang didukung.
Penskalaan	Amazon ECS dapat digunakan dengan kebijakan penskalaan otomatis untuk secara otomatis menyesuaikan jumlah kontainer yang berjalan di kluster Amazon ECS Anda. Lihat Auto Scaling Layanan untuk detail selengkapnya tentang mengonfigurasi penskalaan otomatis untuk aplikasi kontainer Anda di Amazon ECS.

Kemampuan	Deskripsi
Memantau	Amazon ECS mendukung pemantauan sumber daya komputasi dan wadah aplikasi dengan. CloudWatch Lihat Monitoring Amazon ECS untuk detail selengkapnya tentang jenis kemampuan pemantauan yang ditawarkan oleh Amazon ECS.

Diagram berikut menggambarkan Amazon ECS yang digunakan untuk mengelola aplikasi kontainer sederhana. Dalam contoh ini, komponen infrastruktur dibuat di luar Amazon ECS, dan Amazon ECS digunakan untuk mengelola penyebaran dan pengoperasian wadah aplikasi di cluster.



Kasus penggunaan Amazon ECS

> **Note**
>
> - Infrastruktur aplikasi (termasuk repositori Amazon Elastic Container Registry (Amazon ECR), konfigurasi Amazon ECS, dan Load Balancers) disediakan dan dikelola di luar penerapan Amazon ECS Anda.
> - Amazon ECS mengelola penyebaran wadah aplikasi yang berjalan di dalam layanan Amazon ECS sebagai tugas yang bersumber dari registri kontainer seperti Amazon ECR.

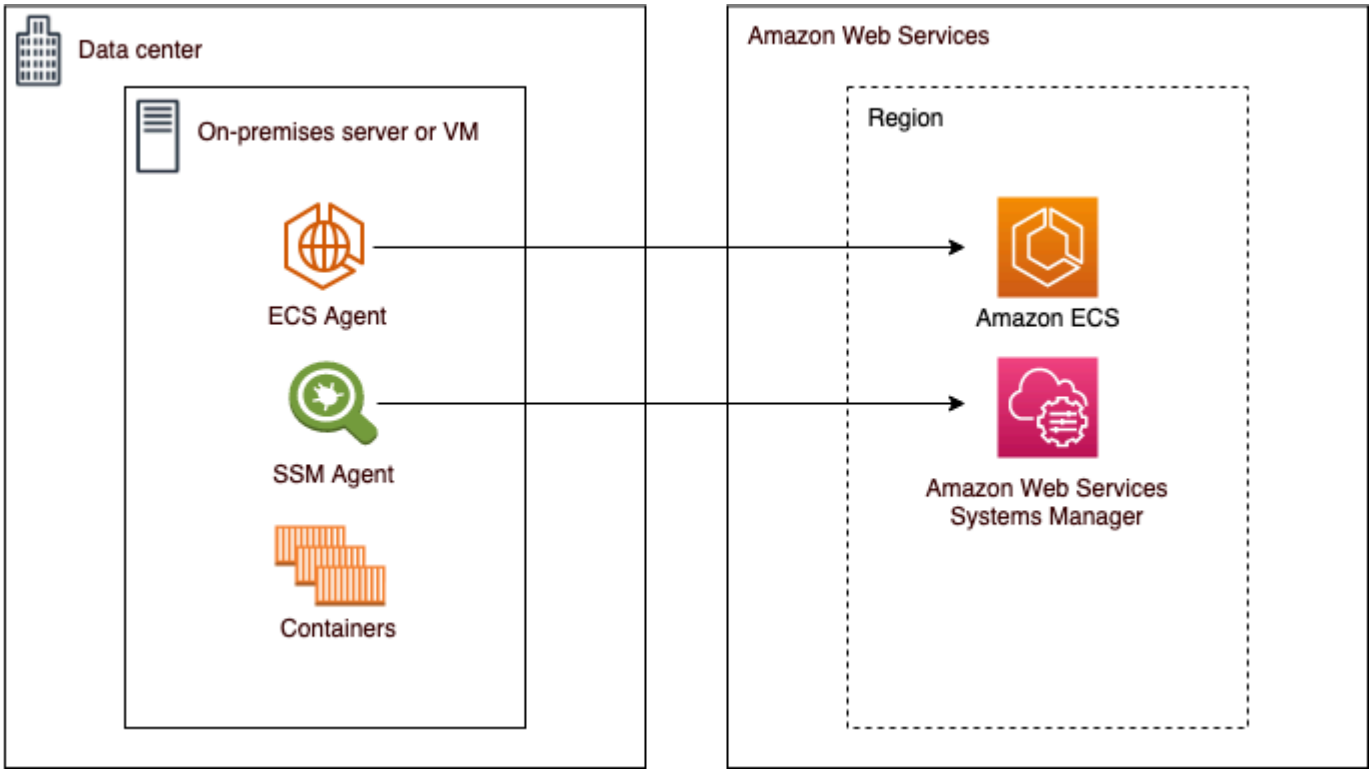
Amazon ECS mendukung beberapa jenis instans kontainer seperti Linux dan Windows, serta jenis instans eksternal seperti mesin virtual lokal (VM) dengan Amazon ECS Anywhere.

Amazon ECS Anywhere

Amazon ECS Anywhere memungkinkan Anda menjalankan tugas Amazon ECS di mana saja, baik di tempat maupun di lingkungan cloud lainnya. Dengan Amazon ECS Anywhere, Anda dapat dengan mudah menerapkan dan mengelola aplikasi kontainer di seluruh infrastruktur hybrid, sekaligus mempertahankan pengalaman operasional yang konsisten. Layanan ini bekerja dengan memperluas platform Amazon ECS ke lingkungan apa pun, termasuk pusat data lokal, kantor jarak jauh, dan lingkungan cloud lainnya. Ini memungkinkan Anda untuk menggunakan Amazon ECS APIs dan perkakas yang sama untuk menyebarkan dan mengelola kontainer di semua lingkungan Anda, tanpa harus khawatir tentang infrastruktur yang mendasarinya.

Amazon ECS Anywhere menggunakan agen Amazon ECS untuk mengelola penerapan dan siklus hidup kontainer, memungkinkan Anda menggunakan definisi tugas Amazon ECS dan file konfigurasi yang sama dengan yang Anda gunakan di. AWS Cloud Ini dapat membantu menyederhanakan proses penyebaran dan pengelolaan kontainer di seluruh infrastruktur hybrid Anda, dan mengurangi waktu dan upaya yang diperlukan untuk konfigurasi dan manajemen manual.

Dengan Amazon ECS Anywhere, Anda juga dapat memanfaatkan layanan AWS lainnya, seperti IAM, dan Amazon ECR AWS CloudFormation, untuk mengelola aplikasi kontainer Anda. Ini dapat membantu memastikan bahwa aplikasi Anda aman, sesuai, dan terintegrasi dengan layanan AWS lainnya.



Amazon ECS Anywhere architecture

Layanan Kontainer Elastis Amazon aktif AWS Outposts

Amazon ECS on AWS Outposts adalah layanan AWS yang dikelola sepenuhnya yang memungkinkan Anda menjalankan tugas Amazon ECS di tempat, menggunakan alat yang sama APIs dan alat yang Anda gunakan di. AWS Cloud Dengan Amazon ECS aktif AWS Outposts, Anda dapat menerapkan dan mengelola aplikasi kontainer dengan cara yang konsisten dan akrab, baik Anda menjalankannya di tempat maupun di cloud. AWS Outposts adalah layanan yang dikelola sepenuhnya yang memperluas infrastruktur, layanan APIs, dan alat AWS ke lingkungan lokal Anda. Dengan Amazon ECS aktif AWS Outposts, Anda dapat menjalankan tugas Amazon ECS pada perangkat keras yang didedikasikan untuk organisasi Anda, tanpa harus khawatir tentang infrastruktur yang mendasarinya. Ini dapat membantu memastikan bahwa aplikasi Anda digunakan dengan cara yang aman dan sesuai, sekaligus memungkinkan Anda memanfaatkan fleksibilitas dan skalabilitas cloud.

Amazon ECS AWS Outposts berfungsi dengan menerapkan serangkaian layanan AWS dan APIs ke lingkungan lokal Anda, yang memungkinkan Anda menjalankan tugas Amazon ECS pada perangkat keras khusus. Ini termasuk agen Amazon ECS, yang mengelola penerapan dan siklus hidup kontainer, dan AWS Outposts infrastruktur, yang menyediakan lingkungan yang aman dan sesuai untuk menjalankan aplikasi kontainer. Dengan Amazon ECS aktif AWS Outposts, Anda

dapat menggunakan Amazon ECS APIs dan perkakas yang sama dengan yang Anda gunakan di dalamnya AWS Cloud, sehingga mudah untuk menerapkan dan mengelola aplikasi kontainer dengan cara yang konsisten dan akrab. Ini dapat membantu mengurangi waktu dan upaya yang diperlukan untuk konfigurasi dan manajemen manual, serta meningkatkan konsistensi dan keandalan di seluruh infrastruktur hybrid Anda. Amazon ECS on AWS Outposts juga terintegrasi dengan layanan AWS lainnya, seperti IAM, AWS CloudFormation dan Amazon ECR, untuk mengelola aplikasi kontainer Anda. Ini dapat membantu memastikan bahwa aplikasi Anda aman, sesuai, dan terintegrasi dengan layanan AWS lainnya.

Amazon Elastic Kubernetes Service

Amazon Elastic Kubernetes Service (Amazon EKS) adalah layanan kesesuaian Kubernetes yang dikelola sepenuhnya dan bersertifikat yang menyederhanakan proses membangun, mengamankan, mengoperasikan, dan memelihara klaster Kubernetes di AWS. Amazon EKS terintegrasi dengan layanan AWS inti seperti CloudWatch, Grup Auto Scaling, dan IAM untuk memberikan pengalaman yang mulus dalam memantau, menskalakan, dan menyeimbangkan beban aplikasi kontainer Anda.

Amazon EKS menyediakan bidang kontrol yang dapat diskalakan dan sangat tersedia untuk beban kerja Kubernetes. Saat menjalankan aplikasi di Amazon EKS, seperti Amazon ECS, Anda dapat memilih untuk memberikan daya komputasi yang mendasari container Anda dengan instans Amazon atau dengan EC2 instans Amazon. AWS Fargate

Amazon VPC Lattice adalah layanan jaringan aplikasi yang dikelola sepenuhnya yang dibangun langsung ke infrastruktur jaringan AWS yang dapat Anda gunakan untuk menghubungkan, mengamankan, dan memantau layanan Anda di beberapa akun dan cloud pribadi virtual (). VPCs Dengan Amazon EKS, Anda dapat memanfaatkan VPC Lattice melalui penggunaan AWS Gateway API Controller, sebuah implementasi dari Kubernetes Gateway API. Menggunakan VPC Lattice, Anda dapat mengatur konektivitas lintas cluster dengan semantik Kubernetes standar dengan cara yang sederhana dan konsisten.

Anda dapat menggunakan Amazon EKS dengan salah satu opsi penerapan berikut:

- Amazon EKS Distro — Amazon EKS Distro adalah distribusi perangkat lunak dan dependensi Kubernetes open-source yang sama yang digunakan oleh Amazon EKS di cloud. Distro Amazon EKS mengikuti siklus versi rilis Kubernetes yang sama seperti Amazon EKS dan disediakan sebagai proyek sumber terbuka. Untuk mempelajari lebih lanjut, lihat Amazon EKS Distro.
- Amazon EKS AWS Outposts aktif — AWS Outposts memungkinkan layanan AWS asli, infrastruktur, dan model operasi di fasilitas lokal Anda. Amazon EKS AWS Outposts aktif, Anda

dapat memilih untuk menjalankan cluster yang diperluas atau lokal. Dengan cluster yang diperluas, bidang kontrol Kubernetes berjalan di an Wilayah AWS dan node berjalan. AWS Outposts Dengan kluster lokal, seluruh klaster Kubernetes berjalan secara lokal AWS Outposts, termasuk bidang kontrol Kubernetes dan node.

- Amazon EKS Anywhere — Amazon EKS Anywhere adalah opsi penerapan untuk Amazon EKS yang memungkinkan Anda membuat dan mengoperasikan klaster Kubernetes secara lokal dengan mudah. Baik Amazon EKS dan Amazon EKS Anywhere dibangun di Amazon EKS Distro. Untuk mempelajari lebih lanjut tentang Amazon EKS Anywhere, lihat Menjalankan beban kerja Hybrid Container dengan Amazon EKS Anywhere, Ikhtisar Amazon EKS Anywhere, dan Membandingkan Amazon EKS Anywhere dengan Amazon EKS.

Saat memilih opsi penerapan mana yang akan digunakan untuk klaster Kubernetes Anda, pertimbangkan hal berikut:

Tabel 5: Fitur penerapan Kubernetes

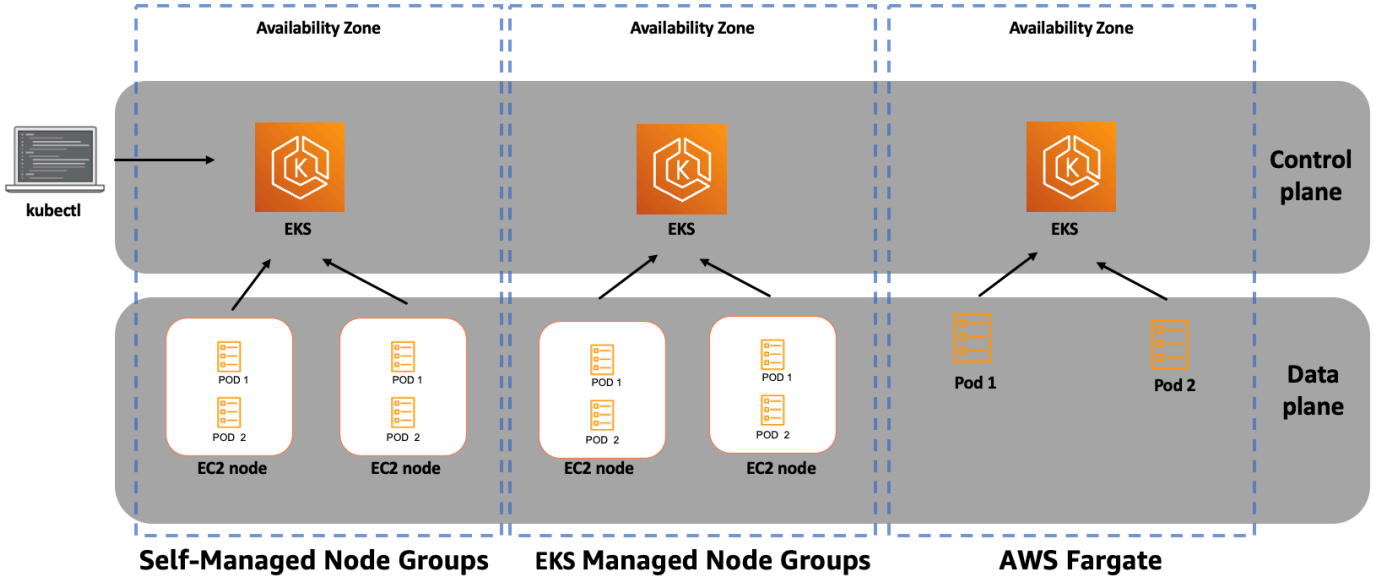
Fitur	Amazon EKS	Amazon EKS di AWS Outposts	Amazon EKS Anywhere	Distro Amazon EKS
Perangkat keras	AWS disediakan	AWS disediakan	Disediakan oleh Anda	Disediakan oleh Anda
Lokasi penyebaran	AWS Cloud	Pusat data Anda	Pusat data Anda	Pusat data Anda
Kubernetes mengontrol lokasi pesawat	AWS Cloud	AWS Cloud atau pusat data Anda	Pusat data Anda	Pusat data Anda
Lokasi pesawat data Kubernetes	AWS Cloud	Pusat data Anda	Pusat data Anda	Pusat data Anda
Dukungan	AWS dukungan	AWS dukungan	AWS dukungan	Dukungan komunitas OSS

Tabel 6: Fitur penyebaran Amazon EKS

Kemampuan	Deskripsi
Ketentuan	Amazon EKS menyediakan sumber daya tertentu untuk mendukung aplikasi kontainer: • Load Balancer, jika diperlukan • Menghitung sumber daya, atau pekerja (Amazon EKS mendukung Windows dan Linux) • Instans Kontainer Aplikasi, atau pod Lihat Memulai Amazon EKS untuk detail selengkapnya tentang penyediaan klaster Amazon EKS.
Konfigurasi	Amazon EKS mendukung penyesuaian sumber daya komputasi (pekerja) jika Anda menggunakan EC2 instans Amazon untuk memasok daya komputasi. Amazon EKS juga mendukung kustomisasi kondisi runtime dari wadah aplikasi (pod). Lihat dokumentasi Worker Nodes dan Fargate Pod Configuration untuk detail selengkapnya.
Deploy	Amazon EKS mendukung strategi penerapan yang sama dengan Kubernetes. Lihat Menulis Spesifikasi Deployment Kubernetes -> Strategy untuk detail selengkapnya.
Penskalaan	Amazon EKS menskalakan pekerja dengan Kubernetes Cluster Autoscaler, dan pod dengan Kubernetes Horizontal Pod Autoscaler dan Kubernetes Vertical Pod Autoscaler. Amazon EKS juga mendukung Karpenter, autoscaler cluster Kubernetes open source, fleksibel, dan berkinerja tinggi untuk membantu

Kemampuan	Deskripsi
	meningkatkan ketersediaan aplikasi dan efisiensi klaster Anda dengan meluncurkan sumber daya komputasi berukuran tepat secara cepat sebagai respons terhadap perubahan beban aplikasi.
Memantau	Log pesawat kontrol Amazon EKS menyediakan informasi audit dan diagnostik langsung ke CloudWatch Log. Pesawat kontrol Amazon EKS juga terintegrasi dengan AWS CloudTrail untuk merekam tindakan yang diambil di Amazon EKS. Lihat Logging dan Monitoring Amazon EKS untuk detail selengkapnya.

Amazon EKS memungkinkan organisasi untuk memanfaatkan alat dan plugin Kubernetes open source, dan dapat menjadi pilihan yang baik bagi organisasi yang bermigrasi ke AWS dengan lingkungan Kubernetes yang ada. Diagram berikut menggambarkan Amazon EKS yang digunakan untuk mengelola aplikasi kontainer umum.



Amazon EKS use case

Amazon EKS Anywhere

Amazon EKS Anywhere memungkinkan Anda membuat dan mengoperasikan klaster Kubernetes di infrastruktur Anda sendiri. Amazon EKS Anywhere dibangun di atas kekuatan Amazon EKS Distro dan menyediakan perangkat lunak sumber terbuka yang mutakhir dan ditambal sehingga Anda dapat memiliki lingkungan Kubernetes lokal yang lebih andal daripada penawaran Kubernetes yang dikelola sendiri.

Amazon EKS Anywhere membuat klaster Kubernetes lokal ke penyedia yang dipilih. Penyedia yang didukung termasuk Bare Metal (via Tinkerbell),, CloudStack dan vSphere. Untuk mengelola klaster itu, Anda dapat menjalankan perintah buat dan hapus cluster dari mesin Administratif Ubuntu atau Mac.

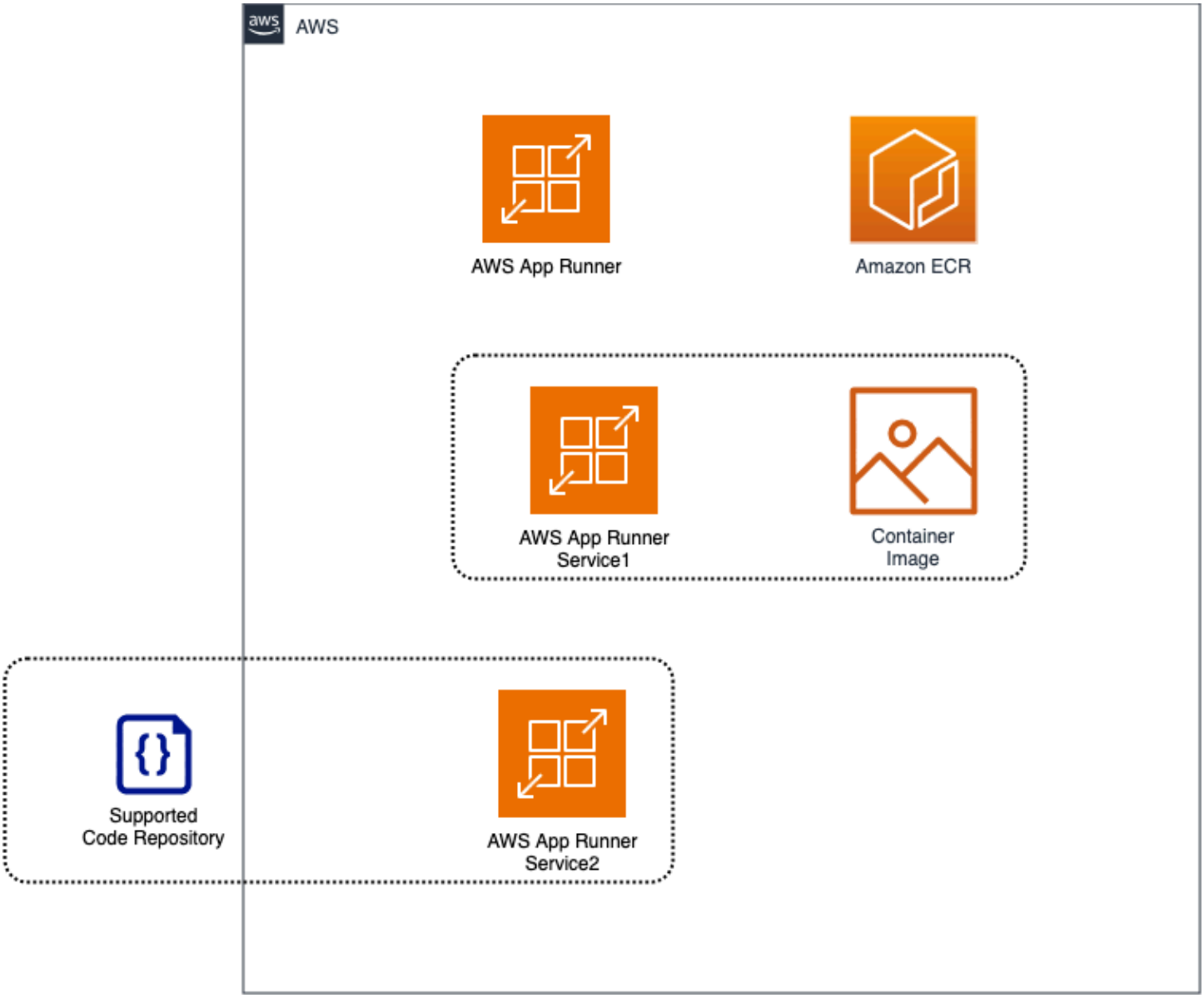
AWS App Runner

AWS App Runneradalah layanan aplikasi kontainer yang dikelola sepenuhnya yang memungkinkan Anda membangun, menyebarkan, dan menjalankan aplikasi web dan layanan API kontainer tanpa infrastruktur atau pengalaman kontainer sebelumnya. App Runner terhubung langsung ke kode atau repositori gambar Anda. Ini menyediakan integrasi otomatis dan pipa pengiriman dengan operasi yang dikelola sepenuhnya, kinerja tinggi, skalabilitas, dan keamanan.

App Runner mengambil kode sumber atau gambar sumber Anda dari repositori dan kemudian membuat dan memelihara layanan web yang sedang berjalan untuk Anda di. AWS Cloud Biasanya, Anda perlu memanggil hanya satu tindakan App Runner,`CreateService`, untuk membuat layanan Anda. Dengan repositori gambar sumber, Anda menyediakan gambar ready-to-use kontainer yang dapat diterapkan oleh App Runner untuk menjalankan layanan web Anda. Dengan repositori kode sumber, Anda memberikan kode dan instruksi untuk membangun dan menjalankan layanan web dan Anda menargetkan lingkungan runtime tertentu. App Runner mendukung beberapa platform pemrograman, masing-masing dengan satu atau lebih runtime terkelola untuk versi utama platform. App Runner mendukung gambar kontainer serta runtime dan kerangka kerja web termasuk Node.js dan Python. App Runner memantau jumlah permintaan bersamaan yang dikirim ke aplikasi Anda dan secara otomatis menambahkan instance tambahan berdasarkan volume permintaan. Jika aplikasi Anda tidak menerima permintaan masuk, App Runner akan menskalakan container ke instance yang disediakan, instance yang dibatasi CPU yang siap melayani permintaan masuk dalam milidetik.

Pada saat ini, App Runner dapat mengambil kode sumber Anda dari GitHub repositori, atau mengambil gambar sumber Anda dari Amazon ECR di Anda. Akun AWS

Diagram berikut menunjukkan ikhtisar arsitektur layanan App Runner. Dalam diagram, ada dua contoh layanan: satu menyebarkan kode sumber dari GitHub, dan yang lainnya menyebarkan gambar sumber dari Amazon ECR.



App Runner use case

App Runner mendukung pengembangan tumpukan penuh, termasuk aplikasi web frontend dan backend yang menggunakan protokol HTTP dan HTTPS. Aplikasi ini termasuk layanan API, layanan web backend, dan situs web. App Runner mendukung gambar kontainer serta runtime dan kerangka kerja web termasuk Node.js dan Python.

Amazon Lightsail

Amazon Lightsail adalah layanan cloud sederhana dan hemat biaya yang memudahkan usaha kecil, startup, dan individu untuk menyebarkan dan mengelola aplikasi mereka di cloud. Ini menyediakan antarmuka yang ramah pengguna yang mengabstraksi sebagian besar manajemen infrastruktur yang mendasarinya dan membuatnya mudah untuk meluncurkan dan menjalankan aplikasi di cloud. Dengan Lightsail, Anda dapat dengan cepat menyebarkan dan mengelola server pribadi virtual (VPS), database, dan instance penyimpanan. Layanan ini menyediakan instance pra-konfigurasi yang dioptimalkan untuk berbagai beban kerja, seperti, Drupal WordPress, dan Joomla, antara lain. Ini dapat membantu mengurangi waktu dan upaya yang diperlukan untuk mengatur dan mengkonfigurasi lingkungan Anda. Lightsail juga menyediakan penyeimbang beban terintegrasi dan penskalaan otomatis, memungkinkan Anda menangani perubahan permintaan lalu lintas tanpa intervensi manual. Layanan ini juga menyediakan pemantauan dan peringatan, sehingga Anda dapat tetap mengetahui kesehatan dan kinerja aplikasi Anda.

Salah satu manfaat utama Lightsail adalah kesederhanaan dan kemudahan penggunaannya. Layanan ini dirancang agar dapat diakses oleh pengguna dengan pengalaman komputasi awan minimal, menjadikannya pilihan yang baik untuk usaha kecil atau individu yang ingin memulai dengan cepat di cloud. Selain itu, Lightsail hemat biaya, dengan harga yang dapat diprediksi yang mencakup komputasi, penyimpanan, dan transfer data.

Wadah Amazon Lightsail

Amazon Lightsail Containers adalah AWS layanan kontainer terkelola penuh yang memudahkan penerapan dan pengelolaan aplikasi kontainer di cloud. Ini menyediakan cara sederhana dan hemat biaya untuk meluncurkan dan menjalankan kontainer menggunakan alat manajemen kontainer populer, seperti Docker dan Kubernetes.

Lightsail Containers menyediakan lingkungan terintegrasi untuk membangun, menguji, dan menyebarkan aplikasi kontainer. Ini menyederhanakan proses penyebaran dan pengelolaan kontainer dengan menyediakan antarmuka yang ramah pengguna yang mengabstraksi sebagian besar manajemen infrastruktur yang mendasarinya.

Dengan Lightsail Containers, Anda dapat menerapkan aplikasi kontainer Anda ke VPC hanya dengan beberapa klik. Layanan ini menyediakan gambar kontainer pra-konfigurasi untuk bahasa pemrograman populer, seperti Node.js, Python, Ruby, dan Java. Ini dapat membantu mengurangi waktu dan upaya yang diperlukan untuk menyiapkan dan mengonfigurasi lingkungan penampung Anda.

Lightsail Containers juga menyediakan penyeimbang beban terintegrasi yang dapat secara otomatis mendistribusikan lalu lintas di seluruh instans kontainer Anda, meningkatkan ketersediaan dan skalabilitas aplikasi. Selain itu, layanan ini menyediakan penskalaan otomatis instans kontainer, memungkinkan Anda menangani perubahan permintaan lalu lintas tanpa intervensi manual.

Dengan Lightsail Containers, Anda dapat memantau kinerja aplikasi kontainer Anda menggunakan metrik dan log bawaan. Anda juga dapat berintegrasi dengan layanan AWS lainnya, seperti Amazon S3, Amazon RDS, dan AWS CodePipeline, untuk membuat pipeline CI/CD yang sepenuhnya otomatis dan terintegrasi untuk aplikasi kontainer Anda.

Layanan OpenShift Red Hat di AWS

Layanan OpenShift Red Hat di AWS(ROSA) adalah layanan terkelola yang tersedia melalui AWS Management Console. Dengan ROSA, sebagai OpenShift pengguna Red Hat, Anda dapat membuat, menskalakan, dan mengelola aplikasi kontainer di AWS. Anda dapat menggunakan ROSA untuk membuat kluster Kubernetes menggunakan Red Hat OpenShift APIs dan alat, serta memiliki akses ke luas dan kedalaman layanan AWS. ROSA merampingkan pemindahan OpenShift beban kerja Red Hat lokal ke AWS, dan menawarkan integrasi yang ketat dengan layanan AWS lainnya. Anda juga dapat mengakses OpenShift lisensi Red Hat, penagihan, dan dukungan semuanya secara langsung melalui AWS.

Setiap cluster ROSA dilengkapi dengan bidang kontrol yang dikelola sepenuhnya dan node komputasi. Instalasi, manajemen, pemeliharaan, dan peningkatan dilakukan oleh Red Hat SRE dengan dukungan bersama Red Hat dan Amazon. Layanan cluster (seperti logging, metrik, dan pemantauan) juga tersedia. Hanya pekerja Red Hat Enterprise Linux CoreOS (RHCOS) yang didukung oleh ROSA.

ROSA akan berintegrasi dengan berbagai komputasi AWS, penyimpanan, database, analitik, pembelajaran mesin, jaringan, seluler, dan berbagai layanan aplikasi AWS, yang akan memungkinkan pelanggan mendapatkan keuntungan dari portofolio layanan AWS yang kuat yang menskalakan sesuai permintaan di seluruh dunia. Layanan asli AWS ini akan langsung dapat diakses untuk menyebarkan dan menskalakan layanan dengan cepat melalui antarmuka manajemen yang sama.

AWS Local Zones

AWS Local Zone adalah perpanjangan Wilayah AWS dari jarak geografis yang dekat dengan pengguna Anda. Zona Lokal memiliki koneksinya sendiri ke internet dan mendukung AWS Direct

Connect. Sumber daya yang dibuat di Zona Lokal dapat melayani pengguna lokal dengan komunikasi latensi rendah. Zona Lokal diwakili oleh kode Wilayah diikuti oleh pengidentifikasi yang menunjukkan lokasi (misalnya, us-west-2-lax-1a).

Amazon ECS mendukung beban kerja yang menggunakan Local Zones ketika latensi rendah atau pemrosesan data lokal merupakan persyaratan. Pesawat kontrol Amazon ECS akan selalu berjalan di. Wilayah AWS

Amazon EKS mendukung sumber daya tertentu di Local Zones. Ini termasuk EC2 node Amazon yang dikelola sendiri, volume Amazon EBS, dan Application Load Balancer. Pesawat kontrol Kubernetes yang dikelola Amazon EKS selalu berjalan di. Wilayah AWS Pesawat kontrol Kubernetes yang dikelola Amazon EKS tidak dapat berjalan di Zona Lokal. Karena Local Zones muncul sebagai subnet dalam VPC Anda, Kubernetes melihat sumber daya Local Zone Anda sebagai bagian dari subnet tersebut.

AWS Wavelength

AWS Wavelengthadalah infrastruktur AWS yang memungkinkan Anda menerapkan beban kerja lebih dekat ke pengguna dan perangkat yang terhubung dengan 5G. Anda dapat menggunakan Wavelength untuk menerapkan instans Amazon, kluster EC2 Amazon EKS, dan rangkaian solusi mitra yang didukung yang tersedia di AWS Marketplace. Wavelength Zones adalah pusat data yang terisolasi secara logis dalam jaringan penyedia telekomunikasi yang terhubung kembali ke Wilayah AWS melalui konektivitas redundan, latensi rendah, dan throughput tinggi.

Beberapa fitur utama Wavelength termasuk kemampuan untuk membuat EC2 instans Amazon, volume Amazon EBS, dan subnet VPC Amazon serta gateway operator di Zona Wavelength. Anda juga dapat menggunakan layanan yang mengatur atau bekerja dengan Amazon, Amazon EBS, EC2 dan Amazon VPC seperti Amazon Auto Scaling, Amazon EKS cluster, Amazon ECS cluster, Amazon Systems Manager EC2 , Amazon Systems EC2 Manager, Amazon,,, dan Application Load Balancer. CloudWatch AWS CloudTrail AWS CloudFormation Layanan Wavelength adalah bagian dari VPC yang terhubung melalui koneksi bandwidth tinggi yang andal ke Wilayah AWS untuk memudahkan akses ke layanan termasuk Amazon DynamoDB dan Amazon Relational Database Service (Amazon RDS).

Layanan Deployment Tambahan

Amazon Simple Storage Service (Amazon S3) dapat digunakan sebagai server web untuk konten statis dan aplikasi satu halaman (SPA). Dikombinasikan dengan Amazon CloudFront untuk

meningkatkan kinerja dalam pengiriman konten statis, menggunakan Amazon S3 dapat menjadi cara sederhana dan ampuh untuk menyebarkan dan memperbarui konten statis. Rincian lebih lanjut tentang pendekatan ini dapat ditemukan di Hosting Situs Web Statis di AWS whitepaper.

AWS Proton

AWS Proton adalah layanan yang dikelola sepenuhnya yang menyederhanakan dan mengotomatiskan proses penerapan dan pengelolaan layanan mikro dan aplikasi berbasis kontainer. Ini memberikan pengalaman penyebaran terpadu dan konsisten yang terintegrasi dengan DevOps alat dan layanan populer, membuatnya lebih mudah untuk mengelola dan merampingkan pengembangan aplikasi. Proton memungkinkan pengembang untuk mendefinisikan dan membuat komponen aplikasi, seperti infrastruktur, kode, dan pipeline, sebagai templat yang dapat digunakan kembali. Template ini dapat digunakan untuk membuat beberapa lingkungan, seperti pengembangan, pengujian, dan produksi, dan dapat dibagikan di seluruh tim atau organisasi. Pendekatan ini membantu mengurangi kompleksitas penerapan dan pengelolaan layanan mikro dan aplikasi berbasis kontainer, yang dapat memakan waktu dan rawan kesalahan.

AWS Proton menyediakan template pra-bangun untuk jenis layanan mikro umum, seperti aplikasi web, dan database APIs, yang dapat disesuaikan untuk memenuhi kebutuhan spesifik. Ini juga terintegrasi dengan DevOps alat populer seperti AWS CodePipeline, AWS, dan AWS CodeCommit CodeBuild, untuk memungkinkan alur kerja integrasi dan penerapan berkelanjutan (CI/CD).

Dengan menggunakan AWS Proton, pengembang dapat mengurangi waktu dan upaya yang diperlukan untuk menerapkan dan mengelola layanan mikro dan aplikasi berbasis kontainer. Pendekatan ini memungkinkan tim untuk fokus pada pengembangan dan peningkatan aplikasi mereka, daripada menghabiskan waktu pada proses penyebaran dan manajemen.

AWS App2Container

AWS App2Containeradalah alat baris perintah untuk memigrasi dan memodernisasi aplikasi web Java dan .NET ke dalam format kontainer. App2Container menganalisis dan membangun inventaris aplikasi yang berjalan di bare metal, mesin virtual, EC2 instans Amazon, atau di cloud. Anda cukup memilih aplikasi yang ingin Anda kontainerisasi, dan App2Container mengemas artefak aplikasi dan mengidentifikasi dependensi ke dalam gambar kontainer, mengonfigurasi port jaringan, dan menghasilkan tugas ECS dan definisi pod Kubernetes. App2Container mengidentifikasi aplikasi ASP.NET dan Java yang didukung yang berjalan di mesin virtual untuk membangun inventaris komprehensif semua aplikasi di lingkungan Anda. App2Container dapat memuat aplikasi web ASP.NET yang berjalan di IIS pada Windows atau Aplikasi Java yang berjalan di Linux, mandiri

atau pada server aplikasi seperti, Apache Tomcat, Springboot, IBM JBoss Websphere, dan Oracle Weblogic.

AWS Copilot

AWS Copilot adalah antarmuka baris perintah (CLI) yang dapat Anda gunakan untuk meluncurkan dan mengelola aplikasi kontainer dengan cepat di AWS. Ini menyederhanakan menjalankan aplikasi di Amazon ECS, Fargate, dan App Runner. AWS Copilot saat ini mendukung sistem Linux, macOS, dan Windows. Copilot memungkinkan Anda untuk menggunakan pola layanan seperti layanan web load balanced untuk menyediakan infrastruktur, menyebarkan ke beberapa lingkungan seperti pengujian atau produksi, dan bahkan menggunakan pipeline AWS CodePipeline rilis untuk penerapan otomatis.

AWS Serverless Application Model

The AWS Serverless Application Model(AWS SAM) adalah kerangka kerja open source untuk membangun aplikasi tanpa server. Ini menyediakan sintaks singkatan untuk mengekspresikan fungsi, database APIs, dan pemetaan sumber peristiwa. Dengan hanya beberapa baris per sumber daya, Anda dapat menentukan aplikasi yang Anda inginkan dan memodelkannya menggunakan YAMAL. Selama penerapan, SAM mengubah dan memperluas sintaks SAM menjadi sintaks CloudFormation AWS, memungkinkan Anda membangun aplikasi tanpa server lebih cepat.

AWS SAM CLI adalah alat baris perintah open source yang memudahkan pengembangan, pengujian, dan penerapan aplikasi tanpa server di AWS. Ini adalah antarmuka baris perintah untuk membangun aplikasi tanpa server menggunakan spesifikasi AWS SAM, yang merupakan perpanjangan dari AWS. CloudFormation

AWS SAM CLI memungkinkan pengembang untuk menentukan dan menguji aplikasi tanpa server mereka secara lokal sebelum menerapkannya ke AWS. Ini menyediakan lingkungan pengujian lokal yang mensimulasikan AWS Lambda dan API Gateway, memungkinkan pengembang untuk menguji kode dan konfigurasi mereka sebelum menerapkannya ke cloud.

AWS SAM CLI juga mencakup berbagai fitur bermanfaat, seperti penerapan kode otomatis, pencatatan, dan kemampuan debugging. Ini memungkinkan pengembang untuk membangun, mengemas, dan menyebarkan aplikasi mereka dengan satu perintah, mengurangi waktu dan upaya yang diperlukan untuk menyebarkan dan mengelola aplikasi tanpa server.

Selain itu, AWS SAM CLI menyediakan dukungan untuk berbagai bahasa pemrograman, termasuk Node.js, Python, Java, dan .NET Core, antara lain. Hal ini memungkinkan pengembang untuk

menggunakan bahasa pemrograman pilihan mereka dan alat untuk membangun dan menyebarkan aplikasi tanpa server mereka.

AWS SAM CLI terintegrasi dengan layanan AWS lainnya, seperti AWS dan CodePipeline AWS CodeBuild, untuk menyediakan pipeline CI/CD yang sepenuhnya otomatis dan terintegrasi untuk aplikasi tanpa server. Ini juga memungkinkan pengembang untuk menggunakan layanan AWS lainnya, seperti Amazon S3, Amazon DynamoDB, dan Amazon SNS, sebagai bagian dari aplikasi tanpa server mereka.

AWS Cloud Development Kit (AWS CDK)

The AWS Cloud Development Kit (AWS CDK)(AWS CDK) adalah kerangka kerja pengembangan perangkat lunak open source untuk mendefinisikan infrastruktur cloud sebagai kode dengan bahasa pemrograman modern dan menerapkannya melalui AWS. CloudFormation AWS Cloud Development Kit (AWS CDK) mempercepat pengembangan cloud menggunakan bahasa pemrograman umum untuk memodelkan aplikasi Anda. AWS CDK memungkinkan Anda membangun aplikasi yang andal, terukur, dan hemat biaya di cloud dengan kekuatan ekspresif yang cukup besar dari bahasa pemrograman.

Pikirkan AWS CDK sebagai toolkit yang berpusat pada pengembang yang memanfaatkan kekuatan penuh bahasa pemrograman modern untuk mendefinisikan infrastruktur AWS Anda sebagai kode. Ketika aplikasi AWS CDK dijalankan, aplikasi tersebut dikompilasi ke templat CloudFormation JSON/YAMM yang terbentuk sepenuhnya yang kemudian dikirimkan ke layanan untuk disediakan. CloudFormation Karena AWS CDK memanfaatkan CloudFormation, Anda tetap menikmati semua manfaat yang CloudFormation diberikan seperti penerapan yang aman, rollback otomatis, dan deteksi drift.

Pendekatan ini menghasilkan banyak manfaat, termasuk:

- Bangun dengan konstruksi tingkat tinggi yang secara otomatis menyediakan default yang masuk akal dan aman untuk sumber daya AWS Anda, mendefinisikan lebih banyak infrastruktur dengan kode yang lebih sedikit.
- Gunakan idiom pemrograman seperti parameter, kondisional, loop, komposisi, dan pewarisan untuk memodelkan desain sistem Anda dari blok bangunan yang disediakan oleh AWS dan lainnya.
- Letakkan infrastruktur, kode aplikasi, dan konfigurasi Anda di satu tempat, pastikan bahwa pada setiap tonggak sejarah Anda memiliki sistem yang lengkap dan dapat diterapkan di cloud.

- Gunakan praktik rekayasa perangkat lunak seperti tinjauan kode, pengujian unit, dan kontrol sumber untuk membuat infrastruktur Anda lebih kuat.
- AWS Solutions Constructs adalah ekstensi perpustakaan sumber terbuka AWS CDK. AWS Solutions Constructs memberi Anda kumpulan pola arsitektur multi-layanan yang telah diperiksa yang dibuat menggunakan praktik terbaik yang ditetapkan oleh AWS Well-Architected Framework.

AWS Serverless Application Model dan AWS CDK keduanya merupakan infrastruktur AWS abstrak sebagai kode sehingga memudahkan Anda menentukan infrastruktur cloud Anda. AWS SAM secara khusus berfokus pada kasus penggunaan dan arsitektur tanpa server dan memungkinkan Anda menentukan infrastruktur Anda dalam templat JSON/YAMM deklaratif yang ringkas. AWS CDK menawarkan cakupan luas di semua layanan AWS dan memungkinkan Anda menentukan infrastruktur cloud dalam bahasa pemrograman modern

Amazon EC2 Image Builder

EC2 Image Builder menyederhanakan pembuatan, pengujian, dan penerapan VM dan image kontainer untuk digunakan di AWS atau lokal. Menyimpan gambar VM dan kontainer up-to-date dapat memakan waktu, intensif sumber daya, dan rawan kesalahan. Saat ini, pelanggan memperbarui dan memotret secara manual VMs atau memiliki tim yang membuat skrip otomatisasi untuk memelihara gambar. Image Builder secara signifikan mengurangi upaya menjaga gambar up-to-date dan keamanan dengan menyediakan antarmuka grafis sederhana, otomatisasi bawaan, dan pengaturan keamanan yang disediakan AWS. Dengan Image Builder, tidak ada langkah manual untuk memperbarui gambar dan Anda juga tidak perlu membuat pipeline otomatisasi Anda sendiri. Image Builder ditawarkan tanpa biaya, selain biaya sumber daya AWS dasar yang digunakan untuk membuat, menyimpan, dan berbagi gambar.

EC2 Image Builder dapat membantu mempermudah penerapan di AWS dengan menyederhanakan proses pembuatan dan pengelolaan gambar khusus untuk digunakan dengan Amazon EC2, container, dan server lokal. Layanan ini menyediakan cara yang disederhanakan dan fleksibel untuk membuat dan mengelola gambar khusus, dengan pipeline build otomatis yang memungkinkan Anda merampingkan proses pembuatan dan manajemen gambar.

EC2 Image Builder menyediakan antarmuka yang ramah pengguna yang mengabstraksi sebagian besar manajemen infrastruktur yang mendasarinya, sehingga memudahkan pengembang untuk membuat dan mengelola gambar khusus. Dengan EC2 Image Builder, pengembang dapat menentukan sistem operasi, aplikasi, dan paket yang ingin mereka sertakan dalam gambar, dan layanan mengotomatiskan proses pembuatan dan pengujian gambar, termasuk pembaruan,

tambalan, dan perbaikan keamanan. Pipeline build otomatis memungkinkan pengembang untuk merampingkan proses pembuatan dan manajemen gambar, mengurangi waktu dan upaya yang diperlukan untuk pembuatan dan pengujian gambar manual. Ini dapat membantu meningkatkan konsistensi, mengurangi kesalahan, dan memastikan bahwa gambar aman up-to-date, dan sesuai.

Berikut ini adalah beberapa manfaat EC2 Image Builder:

- Pembuatan gambar yang disederhanakan: EC2 Image Builder menyediakan cara yang disederhanakan dan fleksibel untuk membuat gambar khusus untuk digunakan dengan Amazon EC2, container, dan server lokal. Ini dapat membantu mengurangi waktu dan upaya yang diperlukan untuk membuat dan memelihara gambar khusus, dan memungkinkan Anda untuk fokus pada aspek penerapan lainnya, seperti pengembangan dan pengujian aplikasi.
- Pipeline pembuatan gambar otomatis: EC2 Image Builder menyediakan saluran pipa otomatis untuk membangun, menguji, dan menerapkan gambar khusus, yang dapat membantu merampingkan proses pembuatan dan manajemen gambar. Ini dapat membantu memastikan bahwa gambar Anda aman up-to-date, dan sesuai, serta mengurangi waktu dan upaya yang diperlukan untuk pembuatan dan pengujian gambar manual.
- Integrasi dengan layanan AWS: EC2 Image Builder terintegrasi dengan layanan AWS lainnya, seperti Amazon Elastic Container Registry (ECR) dan Amazon Elastic Kubernetes Service (EKS), untuk memungkinkan Anda membuat gambar khusus untuk digunakan dengan container. Ini dapat membantu merampingkan proses pembuatan dan penerapan kontainer, memungkinkan Anda membuat gambar khusus yang menyertakan aplikasi, pustaka, dan konfigurasi Anda.
- Pembuatan gambar fleksibel: EC2 Image Builder menyediakan cara yang fleksibel untuk membuat gambar khusus, memungkinkan Anda menentukan sistem operasi, aplikasi, dan paket yang ingin Anda sertakan dalam gambar. Ini dapat membantu memastikan bahwa gambar Anda disesuaikan dengan kasus penggunaan dan persyaratan khusus Anda, dan mengurangi risiko kesalahan atau ketidakcocokan selama penerapan.
- Peningkatan keamanan dan kepatuhan EC2 gambar: Image Builder memungkinkan Anda mengotomatiskan pengujian gambar, termasuk pemindaian kerentanan dan kepatuhan, untuk memastikan bahwa gambar Anda aman dan sesuai. Ini dapat membantu mengurangi risiko pelanggaran keamanan dan meningkatkan kepatuhan, dan memungkinkan Anda untuk menyebarkan aplikasi Anda dengan percaya diri.

Strategi penyebaran

Selain memilih alat yang tepat untuk memperbarui kode aplikasi dan infrastruktur pendukung, menerapkan proses penyebaran yang tepat adalah bagian penting dari solusi penyebaran yang lengkap dan berfungsi dengan baik. Proses penerapan yang Anda pilih untuk memperbarui aplikasi dapat bergantung pada keseimbangan kontrol, kecepatan, biaya, toleransi risiko, dan faktor lainnya yang Anda inginkan.

Setiap layanan penerapan AWS mendukung sejumlah strategi penerapan. Bagian ini akan memberikan gambaran umum tentang strategi penyebaran tujuan umum yang dapat digunakan dengan solusi penerapan Anda.

Memanggang vs. bootstrap AMIs

Jika aplikasi Anda sangat bergantung pada penyesuaian atau penerapan aplikasi ke EC2 instans Amazon, Anda dapat mengoptimalkan penerapan melalui praktik bootstrap dan prebaking.

Menginstal aplikasi, dependensi, atau kustomisasi Anda setiap kali instance Amazon diluncurkan disebut bootstrapping sebuah EC2 instance. Jika Anda memiliki aplikasi yang kompleks atau unduhan besar yang diperlukan, ini dapat memperlambat penyebaran dan peristiwa penskalaan.

Amazon Machine Image (AMI) menyediakan informasi yang diperlukan untuk meluncurkan instance (sistem operasi, volume penyimpanan, izin, paket perangkat lunak, dll.). Anda dapat meluncurkan beberapa instance identik dari satu AMI. Setiap kali EC2 instance diluncurkan, Anda memilih AMI yang akan digunakan sebagai template. Prebaking adalah proses menyematkan sebagian besar artefak aplikasi Anda dalam AMI.

Memanggang komponen aplikasi ke dalam AMI dapat mempercepat waktu peluncuran dan mengoperasionalkan instans Amazon. EC2 Praktik prebaking dan bootstrap dapat digabungkan selama proses penerapan untuk dengan cepat membuat instance baru yang disesuaikan dengan lingkungan saat ini.

Deployment blue/green

Strategi blue/green deployment is a deployment strategy in which you create two separate, but identical environments. One environment (blue) is running the current application version and one environment (green) is running the new application version. Using a blue/green penerapan meningkatkan ketersediaan aplikasi dan mengurangi risiko penerapan dengan menyederhanakan

proses rollback jika penerapan gagal. Setelah pengujian selesai pada lingkungan hijau, lalu lintas aplikasi langsung diarahkan ke lingkungan hijau dan lingkungan biru tidak digunakan lagi.

Sejumlah layanan penerapan AWS mendukung strategi penerapan biru/hijau termasuk Elastic Beanstalk,,,,, dan Amazon ECS. OpsWorks CloudFormation CodeDeploy Lihat Penerapan Biru/Hijau di AWS untuk detail dan strategi selengkapnya untuk menerapkan proses penerapan biru/hijau untuk aplikasi Anda.

Deployment bergulir

Penyebaran bergulir adalah strategi penyebaran yang secara perlahan menggantikan versi aplikasi sebelumnya dengan versi aplikasi baru dengan sepenuhnya mengganti infrastruktur tempat aplikasi berjalan. Misalnya, dalam penerapan bergulir di Amazon ECS, kontainer yang menjalankan versi aplikasi sebelumnya akan diganti one-by-one dengan kontainer yang menjalankan versi baru aplikasi.

Penyebaran bergulir umumnya lebih cepat daripada blue/green deployment; however, unlike a blue/green penerapan, dalam penerapan bergulir tidak ada isolasi lingkungan antara versi aplikasi lama dan baru. Hal ini memungkinkan penerapan bergulir untuk menyelesaikan lebih cepat, tetapi juga meningkatkan risiko dan mempersulit proses rollback jika penerapan gagal.

Strategi penyebaran bergulir dapat digunakan dengan sebagian besar solusi penerapan. Lihat Kebijakan CloudFormation Pembaruan untuk informasi lebih lanjut tentang penerapan bergulir dengan CloudFormation; Pembaruan Bergulir dengan Amazon ECS untuk detail selengkapnya tentang penerapan bergulir dengan Amazon ECS; Pembaruan Konfigurasi Lingkungan Bergulir Elastis Beanstalk untuk detail selengkapnya tentang penerapan bergulir dengan Elastic Beanstalk; dan Menggunakan Penyebaran Bergulir untuk detail lebih lanjut tentang penerapan bergulir dengan. AWS OpsWorks OpsWorks

Penyebaran kenari

Penerapan Canary adalah jenis strategi penyebaran biru/hijau yang lebih menghindari risiko. Strategi ini melibatkan pendekatan bertahap di mana lalu lintas dialihkan ke versi baru aplikasi dalam dua peningkatan. Kenaikan pertama adalah persentase kecil dari lalu lintas, yang disebut sebagai kelompok kenari. Grup ini digunakan untuk menguji versi baru, dan jika berhasil, lalu lintas digeser ke versi baru dengan kenaikan kedua.

Penerapan Canary dapat diimplementasikan dalam dua langkah atau linier. Dalam pendekatan dua langkah, kode aplikasi baru digunakan dan diekspos untuk uji coba. Setelah diterima, itu diluncurkan

baik ke seluruh lingkungan atau secara linier. Pendekatan linier melibatkan peningkatan lalu lintas secara bertahap ke versi baru aplikasi sampai semua lalu lintas mengalir ke rilis baru.

Penerapan di tempat

Penyebaran di tempat adalah strategi penyebaran yang memperbarui versi aplikasi tanpa mengganti komponen infrastruktur apa pun. Dalam penerapan di tempat, 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.

Penerapan di tempat memungkinkan Anda untuk menyebarkan 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.

Lihat Ikhtisar penerapan di tempat untuk detail selengkapnya tentang penggunaan strategi penerapan di tempat dengan. CodeDeploy

Menggabungkan Layanan Penerapan

Tidak ada solusi penerapan “satu ukuran cocok untuk semua” di AWS. Dalam konteks merancang solusi penerapan, penting untuk mempertimbangkan jenis aplikasi karena ini dapat menentukan layanan AWS mana yang paling tepat. Untuk memberikan fungsionalitas lengkap untuk menyediakan, mengonfigurasi, menyebarkan, menskalakan, dan memantau aplikasi Anda, seringkali diperlukan untuk menggabungkan beberapa layanan penyebaran

Pola umum untuk aplikasi di AWS adalah menggunakan CloudFormation (dan ekstensinya) untuk mengelola infrastruktur tujuan umum, dan menggunakan solusi penyebaran yang lebih khusus untuk mengelola pembaruan aplikasi. Dalam kasus aplikasi kontainer, CloudFormation dapat digunakan untuk membuat infrastruktur aplikasi, dan Amazon ECS dan Amazon EKS dapat digunakan untuk menyediakan, menyebarkan, dan memantau kontainer.

Layanan penerapan AWS juga dapat digabungkan dengan layanan penyebaran pihak ketiga. Hal ini memungkinkan organisasi untuk dengan mudah mengintegrasikan layanan penerapan AWS ke dalam CI/CD pipelines or infrastructure management solutions. For example, OpsWorks can be used to synchronize configurations between on-premises and AWS nodes, and CodeDeploy can be used with a number of third-party CI/CD layanan mereka yang ada sebagai bagian dari pipeline lengkap.

Kesimpulan

AWS menyediakan sejumlah alat untuk menyederhanakan dan mengotomatiskan penyediaan infrastruktur dan penyebaran aplikasi; setiap layanan penyebaran menawarkan kemampuan yang berbeda untuk mengelola aplikasi. Untuk membangun arsitektur penerapan yang sukses, evaluasi fitur yang tersedia dari setiap layanan terhadap kebutuhan aplikasi dan organisasi Anda.

Kontributor

Para kontributor untuk dokumen ini antara lain:

- Manikandan Chandrasekaran, Ahli Teknologi Utama
- Anil Nadiminti, Arsitek Solusi Senior
- Bryant Bost, Konsultan AWS ProServe

Sumber Bacaan Lebih Lanjut

Untuk mendapatkan informasi tambahan, buka:

- Halaman AWS Whitepaper
- Pengantar DevOps tentang AWS - Strategi Penerapan

Revisi Dokumen

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

Perubahan	Deskripsi	Tanggal
Laporan resmi diperbarui	Diperbarui secara keseluruhan untuk layanan dan strategi penyebaran terbaru	31 Mei 2024
Pembaruan kecil	Bagian Penerapan Biru/Hijau direvisi untuk kejelasan.	8 April 2021
Laporan resmi diperbarui	Diperbarui dengan layanan dan fitur terbaru.	3 Juni 2020
Publikasi awal	Pertama kali laporan resmi dipublikasikan	1 Maret 2015

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.