aws modern apps

Modernisasikan sekarang dengan kontainer di AWS

Banyak perusahaan di dunia sedang mengalami transformasi digital. Dengan memodernisasi aplikasi yang dimiliki, mereka dapat memberikan pelayanan lebih baik kepada pelanggan, dan bersaing dalam lanskap yang kompetitif. Salah satu cara AWS dalam membantu perusahaan memodernisasi adalah dengan menerapkan kontainer dan memulai pergeseran budaya untuk merampingkan pengembangan. Dalam eBook ini, kami membahas praktik terbaik dalam pembuatan kontainer dan cara Anda memulai sekarang dengan kontainer di AWS.

Menuju digital atau tertinggal

Selama beberapa tahun belakangan, transformasi digital mulai bergelora di seluruh dunia. Perusahaan dari semua skala menemukan berbagai cara baru untuk memanfaatkan teknologi guna meningkatkan ketangkasan mereka dan merespons permintaan dari pelanggan dengan lebih baik. Hal yang memicu perubahan ini adalah kebutuhan untuk bertahan di lingkungan yang berubah. Sekarang pilihannya adalah digital atau tertinggal—Anda tidak perlu mencari terlalu jauh untuk menemukan contoh dari perusahaan ***cloud-native*** yang mendisrupsi industri sekaligus membuat bisnis warisan tertinggal jauh di belakang. Bagi banyak perusahaan, langkah awal menuju transformasi digital adalah memodernisasi aplikasi mereka dan memanfaatkan lingkungan terotomasi di ***cloud***. Modernisasi memberdayakan perusahaan dengan:

Elastisitas: Kemampuan untuk merespons pada lonjakan permintaan pelanggan

Ketersediaan: Kemampuan untuk melayani permintaan pelanggan di mana pun dan kapan pun

Ketangkasan: Kemampuan untuk mengatasi sebuah masalah dengan cepat atau men-***deploy*** fungsionalitas baru yang diinginkan pelanggan

aws modern apps

Kontainer dapat membantu Anda mencapainya

Transformasi digital memakan waktu, tetapi pada akhirnya, peningkatan produktivitas tersebut masuk akal secara bisnis. Meskipun tersedia banyak alat untuk membantu modernisasi perusahaan, kontainer menjadi makin populer sebagai solusi yang dicari bagi developer untuk menggabungkan dan men-***deploy*** aplikasi secara efisien. Menurut Cloud Native Computing Foundation (CNCF) Survey 2020, penggunaan kontainer dalam produksi meningkat 300 persen sejak 2016. Ini terbukti dengan adopsi dua teknologi kontainer berbeda yang saat ini digunakan secara luas—Docker dan Kubernetes. Dari semua beban kerja Kubernetes yang di-***host*** di ***cloud***, 82 persen di antaranya berjalan di AWS.[1]

Kontainer merampingkan pengembangan

Kontainer menyediakan lingkungan perangkat lunak yang portabel, konsisten, dan ringan bagi aplikasi agar mudah menjalankan dan menskalakan di mana pun. Di sepanjang siklus hidupnya, sebuah aplikasi akan beroperasi di banyak lingkungan yang berbeda, entah itu tahapan awal berupa pengujian dan produksi, maupun dari mesin virtual ***on-premise*** ke ***cloud*** selama migrasi. Sebelum kontainer, tim IT harus mempertimbangkan batasan kecocokan dari setiap lingkungan baru dan menulis kode tambahan untuk memastikan aplikasi bisa berfungsi. Kontainer dikembangkan untuk menggabungkan aplikasi dengan dependensi, ***file*** konfigurasi, dan antarmukanya, yang memungkinkan developer untuk menggunakan satu citra yang berpindah dengan lancar ke berbagai ***host***, sehingga tim pengembangan tidak perlu direpotkan dengan manajemen infrastruktur yang monoton dan membebani.

Bagi developer, kontainer memungkinkan mereka untuk fokus membangun aplikasi—seperti menambahkan fitur baru atau keamanan terbaru—bukan menghabiskan waktu mengelola persyaratan kompatibilitas di lingkungan yang berbeda.

Kontainer juga berperan penting dalam membedah arsitektur monolitik tradisional, dan juga memungkinkan transisi ke layanan mikro untuk penskalaan yang lebih mudah. Dengan layanan mikro, setiap komponen aplikasi berjalan dengan layanannya sendiri, memungkinkan developer untuk bekerja secara independen di aspek yang berbeda.

[1] *Laporan Nuclear Research 2019*

Kontainer mendorong pergeseran budaya yang berkelanjutan dalam praktik pengembangan

Kontainer bukan sekadar alat untuk memodernisasi aplikasi Anda, tetapi juga merupakan katalisator untuk meningkatkan praktik pengembangan Anda. Kontainer merombak siklus pengembangan tradisional dengan mendorong developer untuk mengambil alih kontrol kualitas dari aplikasi dan kode yang dikembangkan. Developer umumnya hanya fokus membangun aplikasi, dan mereka tidak berkecimpung dalam keberhasilan penggabungan dan ***deployment***. Dengan kontainer, developer mengalami fenomena yang dikenal dengan sebutan "***shift***", yang memastikan mereka menangani kontrol kualitas di awal, bukan setelah jauh memasuki proses pengembangan.

Setelah developer didorong untuk bertanggung jawab atas implikasi dari kerja mereka, langkah selanjutnya adalah menciptakan sebuah budaya yang mendukung gagal dengan cepat dan belajar dari kesalahan tersebut. Pembuatan kontainer memungkinkan hal ini dengan mengotomatiskan integrasi/***deployment*** kode, sehingga meningkatkan ketangkasan perusahaan secara keseluruhan. sehingga meningkatkan ketangkasan perusahaan secara keseluruhan.

Perusahaan warisan bisa memodernisasi

Sebuah perusahaan besar yang sudah berdiri lebih dari 100 tahun mengambil kesempatan untuk memodernisasikan aplikasinya di salah satu pasar internasionalnya. Dengan menerapkan pendekatan berbasis ***cloud*** ini, tim pengembangan di perusahaan ini menjadi jauh lebih tangkas dan dapat melakukan iterasi jauh lebih cepat. Dengan memanfaatkan kontainer dan lingkungan terotomasi AWS, mereka mampu mendorong platform baru mereka dari tahap konsep ke prototipe ke ***deployment*** dalam waktu enam pekan saja.

aws modern apps

Mulai dengan kontainer di AWS sekarang juga

AWS menyediakan semua solusi yang dibutuhkan untuk membuat kontainer dengan mudah. Alat yang terbukti untuk penyediaan infrastruktur, orkestrasi, keamanan, jaringan, otomatisasi dan pemantauan tersedia untuk membantu Anda memulai dengan kontainer.

Penyediaan

Sediakan infrastruktur dasar dan sumber daya dengan mulus

Untuk menjalankan kontainer, infrastruktur dasar perlu disediakan. AWS menawarkan dua solusi yang berbeda sesuai taraf manajemen atau otomatisasi yang diinginkan:

- Gunakan AWS Fargate untuk mengotomatiskan penyediaan infrastruktur dasar
- Gunakan instans Amazon Elastic Compute Cloud (Amazon EC2) untuk secara manual menentukan komputasi, penyimpanan, dan kemampuan jaringan dari infrastruktur

Orkestrasi

Skalakan dan kelola kontainer Docker atau Kubernetes

- Manfaatkan Amazon Elastic Container Registry (Amazon ECR) untuk menyimpan dan mengelola citra Docker dengan Amazon Elastic Container Service (Amazon ECS) sebagai pengatur orkestrasi untuk Anda
- Adopsi Amazon Elastic Kubernetes Service (Amazon EKS) untuk mengatur kontainer Kubernetes Anda

Keamanan

Amankan, pindai, dan deteksi kerentanan dalam kontainer

- AWS Identity and Access Management (IAM) dan penandaan, grup keamanan untuk instans Amazon EC2, serta Amazon Virtual Private Cloud (VPC) memudahkan Anda mengamankan kontainer
- Solusi pemindaian citra mendeteksi kerentanan pada citra kontainer Docker

Jaringan dan konektivitas

Distribusikan lalu lintas aplikasi di berbagai kontainer

- Distribusikan lalu lintas aplikasi di berbagai kontainer dan lingkungan nirserver dengan AWS Elastic Load Balancing
- Rutekan lalu lintas untuk aplikasi terdistribusi global yang berjalan di kontainer dengan AWS Global Accelerator dan AWS Elastic Load Balancing
- Tingkatkan performa aplikasi dengan AWS Global Accelerator dan AWS Elastic Load Balancing
- Kelola komunikasi dan keamanan antar layanan dengan AWS App Mesh

Otomatisasi

Deploy kode secara otomatis menggunakan CI/CD

- Ciptakan repositori kode sumber menggunakan AWS CodeCommit
- Konfigurasikan alur CI/CD menggunakan AWS CodePipeline
- ***Deploy*** AWS CodeBuild untuk membangun citra kontainer Anda
- Bangun, ***deploy***, dan jalankan aplikasi web yang terkemas dalam kontainer menggunakan AWS App Runner

Pengamatan dan pemantauan

Pastikan layanan yang berjalan di kontainer sehat dan saling berkomunikasi satu sama lain seperti yang diharapkan

- ***Deploy*** AWS App Mesh untuk memberikan visibilitas ke dalam pencatatan log, metrik, dan pelacakan, serta untuk memungkinkan penyeimbangan beban serta pembentukan lalu lintas
- Jalankan pemeriksaan kondisi citra kontainer Docker Anda untuk memastikan bahwa kontainer Anda berjalan dan aplikasi Anda bekerja
- Gunakan Wawasan Aplikasi Amazon CloudWatch untuk memantau kondisi dan keadaan aplikasi yang berjalan dalam kontainer yang di-***deploy*** di Amazon ECS, Amazon EKS, atau Kubernetes di Amazon EC2

Penyediaan

AWS memberikan Anda opsi penyediaan infrastruktur Anda dengan Amazon EC2 dan AWS Fargate. Perbedaan utama antara keduanya adalah jumlah dari manajemen dan kontrol yang Anda inginkan atas infrastruktur dasar yang menjalankan aplikasi kontainer Anda.

- **AWS Fargate:** Dengan Fargate, Anda dapat menjalankan kontainer Anda tanpa perlu mengelola server atau klaster. Yang perlu Anda lakukan hanya menyatukan aplikasi Anda dalam kontainer, tentukan persyaratan CPU dan memori, tetapkan kebijakan jaringan dan IAM, lalu jalankan aplikasinya.

- **Amazon EC2:** Tipe peluncuran EC2 yang lebih tradisional memungkinkan Anda untuk membawa instans guna menjalankan kontainer dan menyediakan kontrol yang lebih terperinci di tingkat server atas infrastruktur yang menjalankan aplikasi kontainer Anda.

Orkestrasi

Setelah aplikasi Anda terkemas dalam kontainer, langkah selanjutnya adalah menjalankan kontainer dalam produksi. Untuk menskalakan arsitektur Anda, Anda memerlukan alat orkestrasi. AWS menawarkan platform orkestrasi yang disesuaikan dengan kebutuhan Anda, baik itu ***on-premise*** maupun di ***cloud***.

Amazon ECS dan Amazon ECS Anywhere:

Amazon Elastic Container Service (Amazon ECS) memungkinkan Anda untuk men-***deploy*** beban kerja yang terkemas dalam kontainer di AWS dengan mudah. Kesederhanaan Amazon ECS yang begitu efisien memungkinkan Anda berkembang dari satu kontainer Docker hingga mengelola portofolio aplikasi korporasi Anda secara menyeluruh. Jalankan dan skalakan beban kerja kontainer Anda di berbagai Availability Zone, di ***cloud***, dan ***on-premise***, tanpa risau dengan rumitnya mengelola bidang kendali atau simpul. Amazon ECS menyediakan bidang kendali yang dapat diskalakan secara tak terbatas yang dikelola untuk Anda. Alat orkestrasi semacam ini sangat cocok untuk perusahaan yang menggunakan sistem operasi sumber tertutup atau menginginkan kontrol atas infrastruktur mereka sendiri. AWS Copilot adalah antarmuka baris perintah (CLI) yang memungkinkan pelanggan untuk dengan cepat meluncurkan aplikasi yang terkemas dalam kontainer, serta mengelolanya dengan mudah, di AWS. AWS Copilot menyediakan sekumpulan perintah yang ringkas dan jelas, termasuk contoh dan pengalaman terpandu bawaan untuk membantu pelanggan men-***deploy*** dengan cepat. Untuk mempercepat layanan siap-produksi, Anda hanya memerlukan AWS Copilot, akun AWS, dan kode Anda.

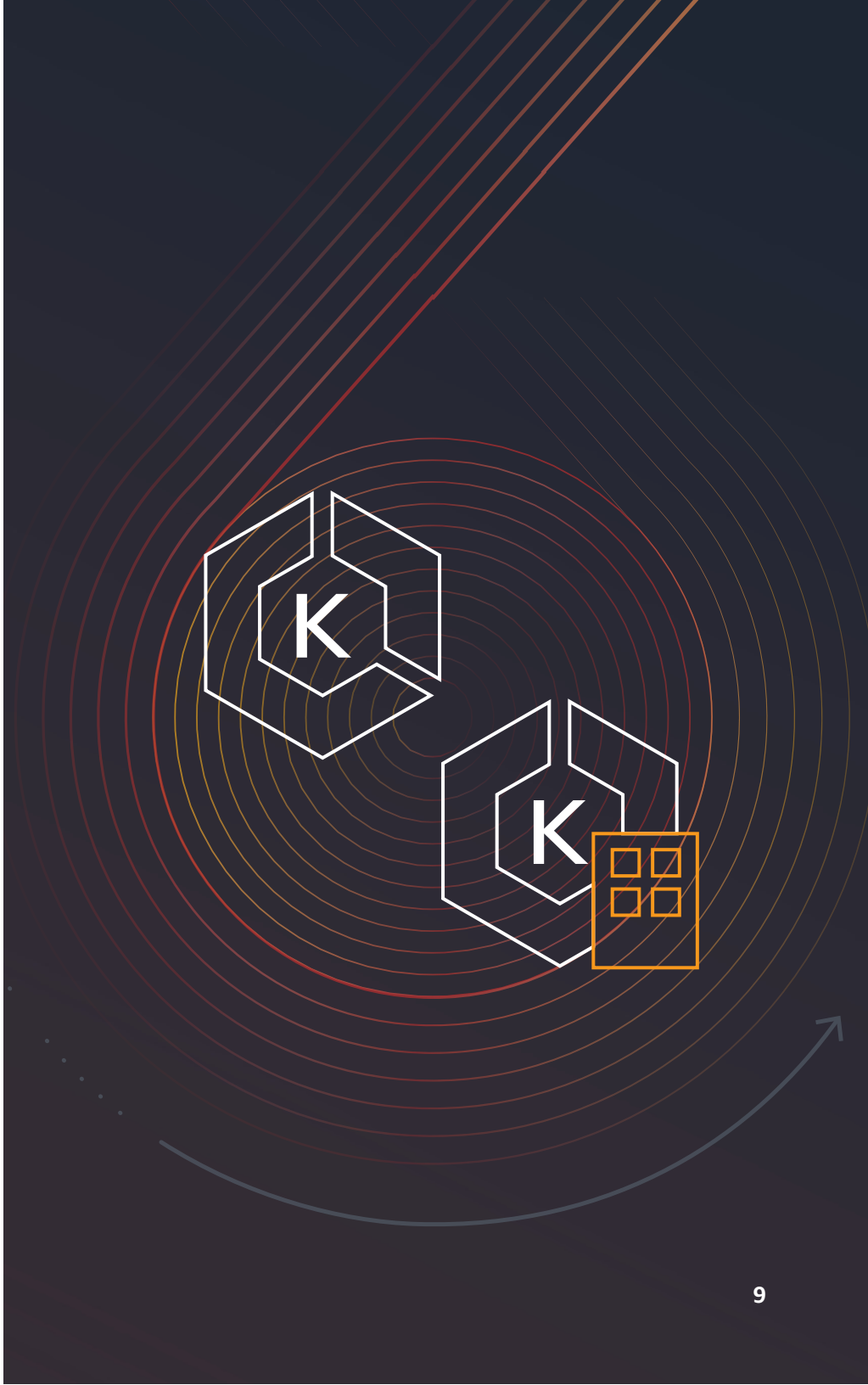
Dengan Amazon ECS Anywhere, Anda bisa menggunakan alat operator dan konsol Amazon ECS yang serupa untuk mengelola beban kerja kontainer *on-premise* Anda untuk menghasilkan pengalaman yang konsisten di semua aplikasi Anda yang berbasis kontainer. Integrasi AWS Systems Manager (dulu dikenal sebagai SSM) secara otomatis dan aman membentuk kepercayaan antara perangkat keras *on-premise* Anda dan bidang kendali AWS.

Amazon EKS dan Amazon EKS Anywhere:

Kubernetes adalah platform manajemen kontainer sumber terbuka yang berkembang cepat, yang dibangun untuk membantu Anda menjalankan kontainer dalam skala besar. Dengan Amazon EKS, infrastruktur manajemen Kubernetes dijalankan untuk Anda di berbagai Availability Zone AWS.

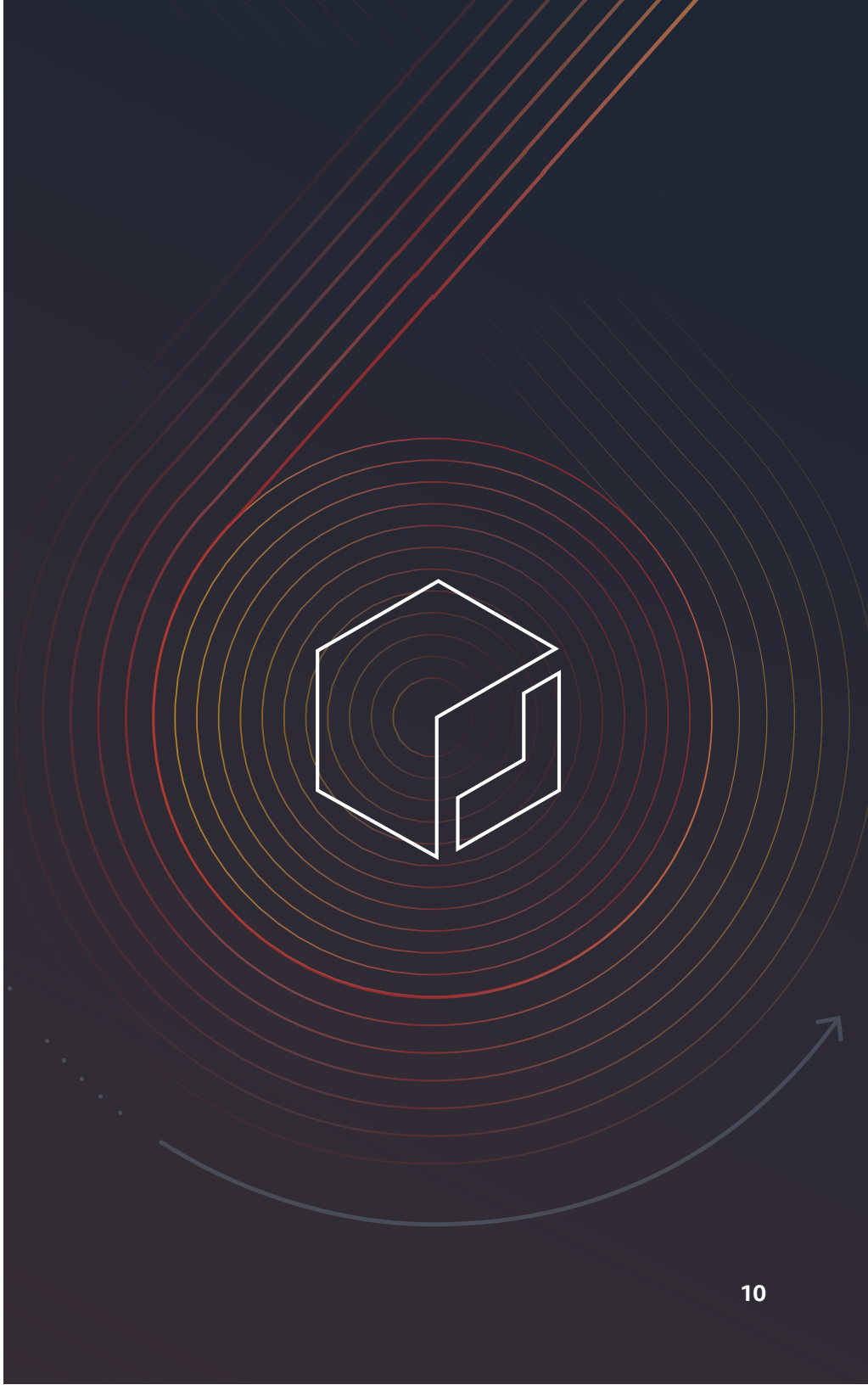
Pelanggan kami percaya bahwa terdapat keuntungan besar menjalankan Kubernetes di AWS. Amazon EKS secara otomatis mengelola ketersediaan dan skalabilitas simpul bidang kontrol Kubernetes yang bertanggung jawab untuk menjadwalkan kontainer, mengelola ketersediaan aplikasi, menyimpan data klaster, dan berbagai tugas penting lainnya. Anda dapat menjalankan aplikasi Kubernetes Anda di Amazon EC2 maupun AWS Fargate, yang menyediakan komputasi nirserver untuk kontainer.

Amazon EKS Anywhere memungkinkan Anda untuk membuat dan mengoperasikan klaster Kubernetes dengan mudah (membangun dengan perangkat lunak di Amazon EKS Distro) ***on-premise***, termasuk mesin virtual (VM) Anda sendiri dan server ***bare metal***. Dengan EKS Anywhere, Anda tidak perlu repot membangun dan mendukung peralatan Anda sendiri untuk mengelola klaster Kubernetes. EKS Anywhere menyediakan peralatan otomatisasi yang menyederhanakan pembuatan klaster, administrasi, dan pengoperasian di infrastruktur seperti ***bare metal***, vSphere, dan mesin virtual ***cloud*** dengan konfigurasi ***default*** untuk pencatatan log, pemantauan, jaringan, dan penyimpanan, tetapi menghadirkan peralatan dogmatis dan komponen tambahan yang Anda perlukan untuk menjalankan Kubernetes dalam produksi, seperti penginstalan klaster dan manajemen siklus hidup, observabilitas, pencadangan klaster, dan manajemen kebijakan. Amazon EKS Distro menggabungkan distribusi perangkat lunak Kubernetes sumber terbuka yang sama dengan yang digunakan di EKS di AWS untuk digunakan di infrastruktur Anda secara ***on-premise***. Klaster EKS Distro dapat dikelola dengan peralatan Anda sendiri atau dengan Amazon EKS Anywhere.



Amazon ECR:

Untuk menggunakan kontainer Docker, hal pertama yang Anda butuhkan adalah citra Docker. Gambar ini berfungsi sebagai cetak biru untuk membuat instans kontainer. Amazon ECR adalah registri kontainer Docker terkelola penuh yang memudahkan Anda menyimpan, mengelola, dan men-***deploy*** citra kontainer Docker. Amazon ECR terintegrasi dengan Amazon ECS, menyederhanakan alur kerja pengembangan-ke-produksi Anda. Dengan Amazon ECR Anda tidak perlu lagi mengoperasikan repositori kontainer Anda atau khawatir tentang penskalaan infrastruktur yang mendasarinya.

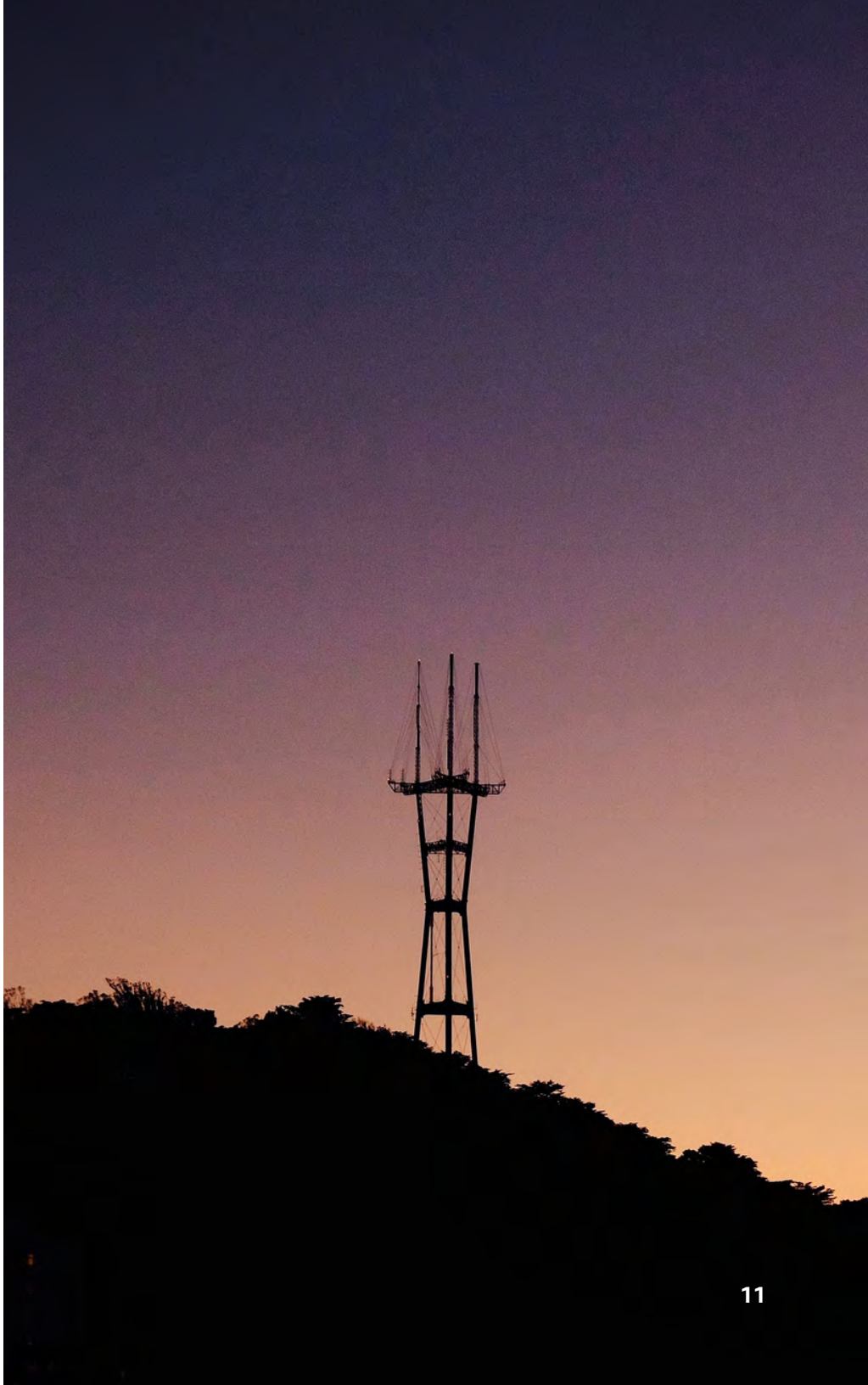


Keamanan

Seperti kontrol kualitas, pertimbangan keamanan juga telah bergeser ke tahap awal dari siklus pengembangan. Dengan otonomi yang lebih besar, developer bisa lebih siap dalam mengadaptasi kodenya untuk menangani ancaman keamanan terbaru. Meski begitu, keamanan harus menjadi prioritas di seluruh organisasi. Satu cara untuk mendukung perubahan budaya ini adalah dengan menciptakan transparansi semaksimal mungkin dalam usaha keamanan dan menetapkan terlebih dahulu arsitektur yang mempertimbangkan berbagai alat serta praktik terbaik dalam hal keamanan.

AWS menyediakan berbagai alat untuk mengontrol siapa yang dapat mengakses kontainer Anda. Anda bisa menggunakan AWS IAM untuk menentukan siapa yang diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya. Amazon VPC memungkinkan Anda mengisolasi tugas kontainer (Docker) atau ***pod*** (Kubernetes) secara logis di jaringan virtual yang Anda tetapkan. Anda dapat menetapkan grup keamanan untuk membuat ***firewall*** virtual antar instans EC2.

Dengan menggunakan solusi pemindaian citra, Anda dapat mendeteksi kerentanan citra kontainer atau dependensi citra. Tim keamanan dapat melakukan pemindaian pada kontainer atau dependensi citra tersebut, kemudian menerbitkan sumber daya yang telah disetujui dan dapat digunakan developer tanpa ragu.



Jaringan dan konektivitas

Setelah aplikasi Anda berjalan, Anda sebaiknya memastikan bahwa lalu lintas selalu terdistribusi di seluruh kontainer Anda, karena hal ini memungkinkan pengguna akhir Anda untuk menggunakan aplikasi Anda tanpa gangguan.

- **Elastic Load Balancing** membantu pengguna mengakses aplikasi Anda dengan mudah, menggunakan algoritme penyeimbangan beban yang secara ***native*** terintegrasi dengan layanan kontainer AWS.

- **Amazon Global Accelerator** juga membantu memastikan bahwa aplikasi Anda memiliki performa tinggi dan dapat diakses oleh pengguna di seluruh dunia, dan dilayani dari Wilayah AWS yang paling dekat dengan lokasi mereka.

- Untuk mengelola komunikasi antara layanan kontainer, **AWS App Mesh** memungkinkan keamanan terperinci dan visibilitas mendalam.

Otomatisasi

Dari sekarang hingga 2024, permintaan untuk kemampuan integrasi hibrida akan semakin meningkat seiring dengan meningkatnya konsolidasi lanskap yang kompetitif dari sebagian besar penawaran PaaS.[2] Dengan Amazon EKS Anywhere dan Amazon ECS Anywhere, penuhi kepatuhan, gravitasi data, dan persyaratan bisnis lainnya dengan menjalankan beban kerja Anda di infrastruktur Anda sendiri, dengan peralatan yang sederhana dan sudah Anda pahami dengan baik. Daripada memasang dan mengoperasikan bidang kendali lokal, Anda bisa menggunakan bidang kendali skala raksasa yang sama, tepercaya, dan terkelola penuh untuk beban kerja kontainer ***on-premise*** Anda. Jalankan beban kerja pemrosesan data yang terkemas dalam kontainer di lokasi ***edge*** pada perangkat keras Anda sendiri, sehingga Anda bisa tetap berada dekat dengan pelanggan akhir dan mempertahankan latensi yang rendah.

Lingkungan terotomasi tidak memerlukan penerapan kode secara manual. Saat infrastruktur Anda berisi ratusan atau ribuan kontainer, melakukan otomatisasi dengan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) memungkinkan Anda menskalakan dan bereaksi lebih cepat sekaligus meminimalkan risiko kesalahan manusia. AWS CodeCommit, AWS CodePipeline, dan AWS CodeBuild memungkinkan Anda untuk membuat repositori kode sumber, mengonfigurasikan alur CI/CD, dan membangun citra kontainer Anda.

AWS App Runner adalah layanan aplikasi terkelola penuh yang memudahkan developer Anda untuk men-***deploy*** dan menjalankan aplikasi web yang terkemas dalam kontainer serta API dalam skala besar, hanya dengan beberapa klik. Dengan App Runner, Anda tidak perlu mengonfigurasi dan mengelola infrastruktur. Anda cukup menyediakan kode sumber, citra kontainer, atau alur ***deployment***, dan App Runner membangun serta men-***deploy*** aplikasi web, menyeimbangkan beban lalu lintas jaringan, menaikkan atau menurunkan skala kapasitas sesuai permintaan, memantau kondisi aplikasi, dan mengenkripsi lalu lintas secara ***default***. Anda juga tidak memerlukan pengalaman dalam hal kontainer untuk menggunakan App Runner dan memanfaatkan kontainer yang portabel, efisien, dan hemat biaya.

[2] Analisis Gartner Forecast, Feb 2021

Pengamatan dan pemantauan

Setelah layanan mikro di-***deploy*** menggunakan kontainer, Anda perlu memantau kondisi kontainer dan memastikan bahwa layanan-layanan saling berkomunikasi satu sama lain sebagaimana diharapkan.

- **AWS App Mesh** memudahkan Anda menjalankan layanan dengan menyediakan visibilitas dan kontrol lalu lintas jaringan secara konsisten untuk layanan yang dibangun di beragam jenis infrastruktur komputasi. App Mesh menghilangkan keperluan untuk memperbarui kode aplikasi untuk mengubah bagaimana cara data pemantauan dikumpulkan atau lalu lintas dirutekan antarlayanan. App Mesh mengonfigurasi setiap layanan untuk mengekspor data pemantauan dan menerapkan logika kontrol komunikasi yang konsisten di seluruh aplikasi Anda. Hal ini memudahkan Anda menentukan lokasi kesalahan dengan cepat dan secara otomatis mengubah rute lalu lintas jaringan saat terjadi kegagalan atau saat perubahan kode perlu diterapkan. App Mesh menggunakan Envoy Proxy sumber terbuka, membuatnya cocok dengan berbagai partner AWS dan alat sumber terbuka.

- Citra kontainer Docker memungkinkan Anda untuk memverifikasi kondisi kontainer dan aplikasi. Dengan perintah pemeriksaan kondisi yang sederhana, sebuah ***file*** Docker akan memeriksa sebuah kontainer untuk memastikan bahwa kontainer tersebut masih berfungsi. Proses ini dapat mendeteksi saat sebuah server web macet dalam sebuah putaran tak berujung dan tidak dapat menangani koneksi baru, meski proses server masih berjalan.

- **Wawasan Aplikasi Amazon CloudWatch** mendukung pemantauan kontainer: Anda dapat dengan mudah menyiapkan pemantauan, alarm, dan dasbor untuk aplikasi Anda yang di-***deploy*** di kontainer **Amazon ECS**, **Amazon EKS**, dan **Kubernetes di EC2** yang berjalan di AWS. Opsi tingkatan pemantauan kini tersedia untuk mencatat metrik, telemetri, dan log guna memantau kondisi dan keadaan aplikasi yang berjalan dalam kontainer di AWS.

Siap untuk memanfaatkan kontainer semaksimal mungkin?

AWS dan ekosistem partner kami yang luas akan menyediakan alat-alat yang Anda perlukan untuk memulai—di tahap apa pun perjalanan modernisasi Anda berada.

Anda selalu dapat menghubungi karyawan penjualan AWS atau berbicara dengan Partner AWS »

Untuk mempelajari lebih lanjut, kunjungi: aws.amazon.com/containers

aws modern apps