Laporan Resmi AWS

Hosting Aplikasi Web di AWS Cloud

Hosting Aplikasi Web di AWS Cloud: Laporan Resmi AWS

Table of Contents

Hosting Aplikasi Web di AWS Cloud

Tanggal publikasi: 20 Agustus 2021 (Revisi dokumen)

Abstrak

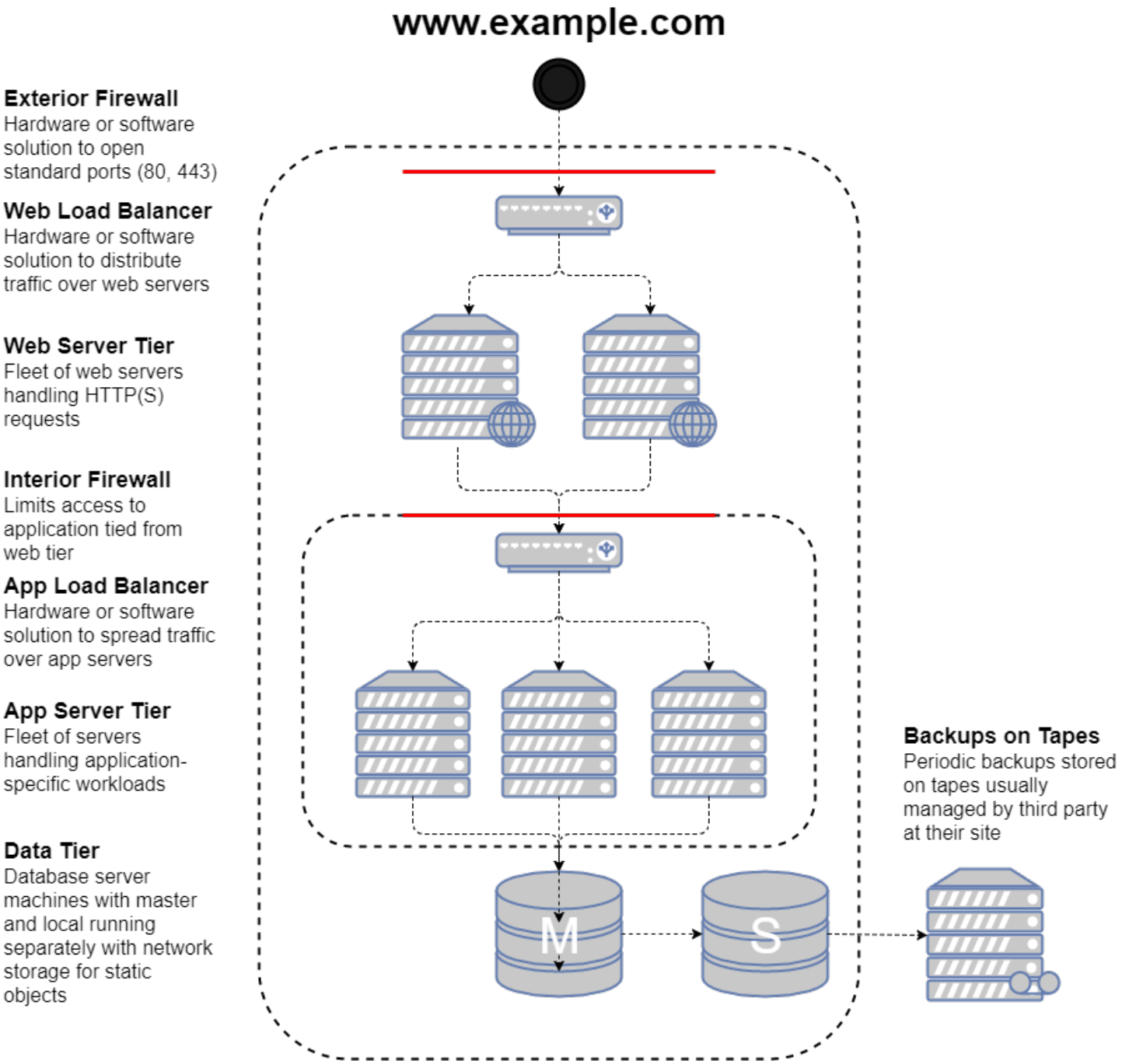
Arsitektur web on-premise tradisional memerlukan solusi kompleks dan perkiraan kapasitas tersimpan yang akurat untuk memastikan keandalan. Periode lalu lintas puncak yang padat dan ayunan liar dalam pola lalu lintas menghasilkan tingkat pemanfaatan perangkat keras yang mahal. Ini menghasilkan biaya operasi yang tinggi untuk mempertahankan perangkat keras yang tidak aktif, dan penggunaan modal yang tidak efisien untuk perangkat keras yang kurang digunakan.

Amazon Web Services (AWS) memberikan infrastruktur yang andal, dapat diskalakan, aman, dan berperforma tinggi yang diperlukan untuk aplikasi web yang paling menuntut. Infrastruktur ini cocok dengan biaya IT dengan pola lalu lintas pelanggan dalam hampir waktu nyata.

Laporan resmi ini dimaksudkan untuk Manajer IT dan Arsitek Sistem yang ingin memahami cara menjalankan arsitektur web tradisional di cloud untuk mencapai elastisitas, skalabilitas, dan keandalan.

Gambaran umum hosting web tradisional

Hosting web yang dapat diskalakan adalah ruang masalah yang terkenal. Gambar berikut menggambarkan arsitektur hosting web tradisional yang mengimplementasikan model aplikasi web tiga tingkat umum. Dalam model ini, arsitektur dipisahkan menjadi lapisan presentasi, aplikasi, dan ketekunan. Skalabilitas disediakan dengan menambahkan host pada lapisan ini. Arsitektur ini juga memiliki fitur performa, failover, dan ketersediaan bawaan. Arsitektur hosting web tradisional mudah di-porting ke AWS Cloud dengan hanya beberapa modifikasi.



Arsitektur hosting web tradisional

Bagian berikut melihat mengapa dan bagaimana arsitektur semacam itu seharusnya dan dapat di-deploy di AWS Cloud.

Hosting aplikasi web di cloud menggunakan AWS

Pertanyaan pertama yang harus Anda tanyakan menyangkut nilai memindahkan solusi hosting aplikasi web klasik ke AWS Cloud. Jika Anda memutuskan bahwa cloud tepat untuk Anda, Anda akan memerlukan arsitektur yang sesuai. Bagian ini membantu Anda mengevaluasi solusi AWS Cloud. Ini membandingkan men-deploy aplikasi web Anda di cloud dengan penerapan on-premise, menghadirkan arsitektur AWS Cloud untuk meng-host aplikasi Anda, dan membahas komponen utama solusi AWS Cloud Architecture.

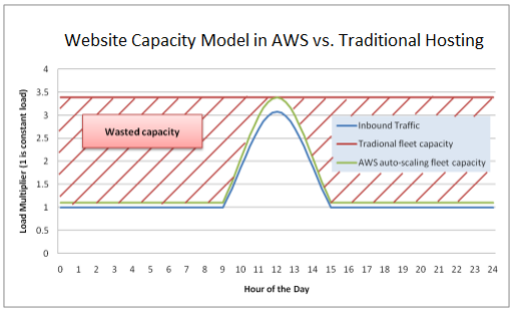
Bagaimana AWS dapat mengatasi masalah hosting aplikasi web yang umum

Jika Anda bertanggung jawab untuk menjalankan aplikasi web, Anda dapat menghadapi berbagai masalah infrastruktur dan arsitektur yang AWS dapat menyediakan solusi yang lancar dan hemat biaya. Berikut adalah beberapa manfaat menggunakan AWS dibandingkan model hosting tradisional.

Alternatif hemat biaya untuk armada besar yang dibutuhkan untuk menangani puncak

Dalam model hosting tradisional, Anda harus menyediakan server untuk menangani kapasitas puncak. Siklus yang tidak terpakai terbuang di luar periode puncak. Aplikasi web yang di-host oleh AWS dapat memanfaatkan penyediaan server tambahan sesuai permintaan, sehingga Anda dapat terus menyesuaikan kapasitas dan biaya dengan pola lalu lintas yang sebenarnya.

Sebagai contoh, grafik berikut menunjukkan aplikasi web dengan puncak penggunaan dari 9.00 sampai 15.00 dan kurang penggunaan untuk sesudahnya. Pendekatan penskalaan otomatis berdasarkan tren lalu lintas aktual, yang menyediakan sumber daya hanya bila diperlukan, akan menghasilkan kapasitas yang kurang terbuang dan pengurangan biaya lebih dari 50 persen.



Contoh kapasitas terbuang dalam model hosting klasik

Solusi yang dapat diskalakan untuk menangani puncak lalu lintas yang tidak terduga

Konsekuensi yang lebih mengerikan dari penyediaan lambat yang terkait dengan model hosting tradisional adalah ketidakmampuan untuk merespons pada waktunya terhadap lonjakan lalu lintas yang tak terduga. Ada sejumlah cerita tentang aplikasi web menjadi tidak tersedia karena lonjakan lalu lintas yang tak terduga setelah situs tersebut disebutkan di media populer. Di AWS Cloud, kemampuan sesuai permintaan yang sama yang membantu skala aplikasi web agar sesuai dengan lonjakan lalu lintas reguler juga dapat menangani beban yang tidak terduga. Host baru dapat diluncurkan dan tersedia dalam hitungan menit, dan mereka dapat diambil secara offline dengan cepat ketika lalu lintas kembali normal.

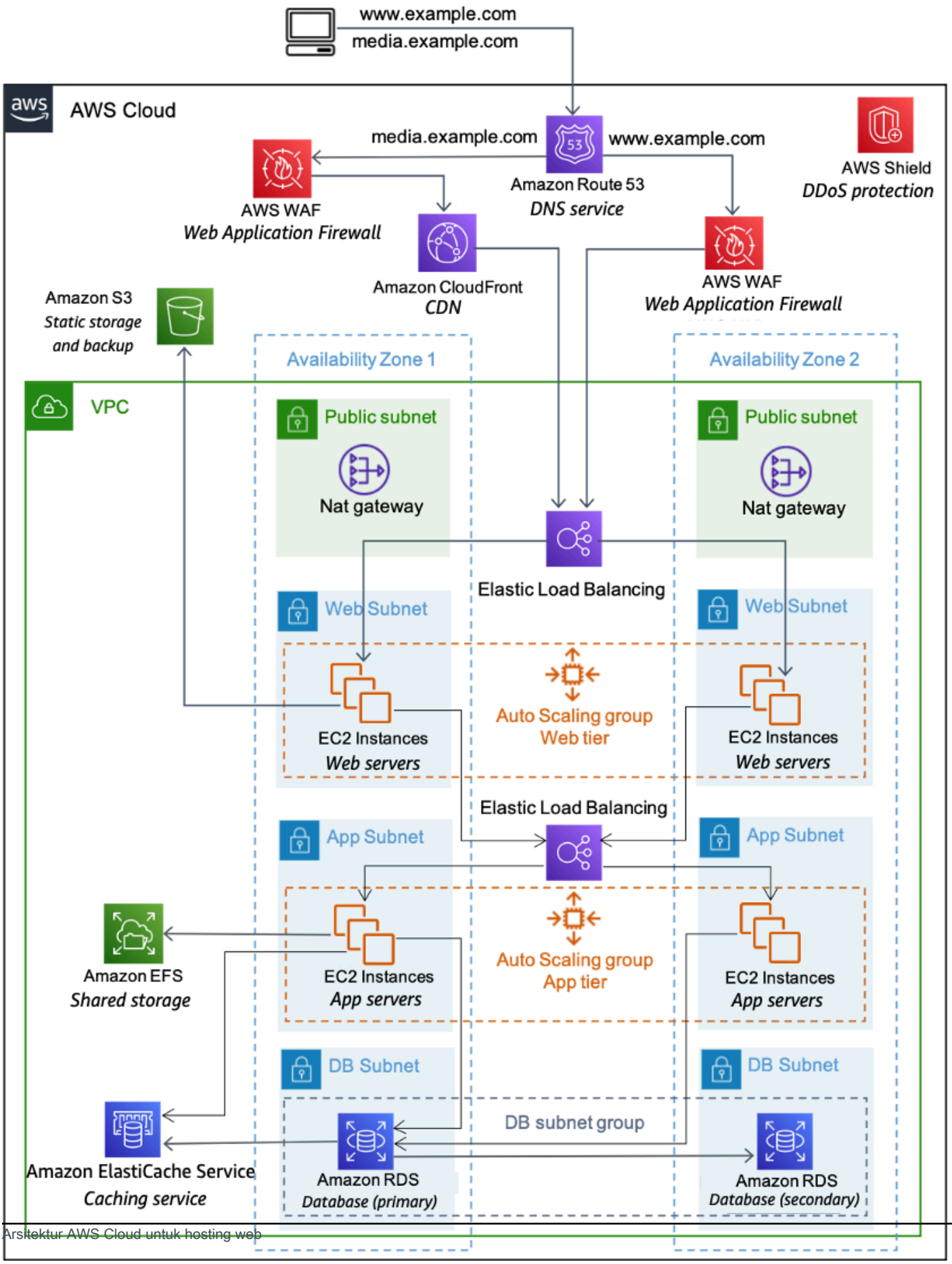
Solusi sesuai permintaan untuk lingkungan pengujian, beban, beta, dan reproduksi

Biaya perangkat keras untuk membangun dan memelihara lingkungan hosting tradisional untuk aplikasi web produksi tidak berhenti dengan armada produksi. Seringkali, Anda perlu membuat armada praproduksi, beta, dan pengujian untuk memastikan kualitas aplikasi web pada setiap tahap siklus hidup pengembangan. Meskipun Anda dapat membuat berbagai optimalisasi untuk memastikan penggunaan tertinggi dari perangkat keras pengujian ini, armada paralel ini tidak selalu digunakan secara optimal, dan banyak perangkat keras mahal duduk tidak terpakai untuk jangka waktu yang lama.

Di AWS Cloud, Anda dapat menyediakan armada pengujian saat dan kapan Anda membutuhkannya. Hal ini tidak hanya menghilangkan kebutuhan untuk pra-penyediaan sumber daya hari atau bulan sebelum penggunaan aktual, tetapi memberi Anda fleksibilitas untuk meruntuhkan komponen infrastruktur ketika Anda tidak membutuhkannya. Selain itu, Anda dapat mensimulasikan lalu lintas pengguna di AWS Cloud selama pengujian beban. Anda juga dapat menggunakan armada paralel ini sebagai lingkungan pementasan untuk rilis produksi baru. Hal ini memungkinkan peralihan cepat dari produksi saat ini ke versi aplikasi baru dengan sedikit atau tanpa pemadaman layanan.

Arsitektur AWS Cloud untuk hosting web

Gambar berikut memberikan tampilan lain pada arsitektur aplikasi web klasik itu dan bagaimana hal itu dapat memanfaatkan infrastruktur komputasi AWS Cloud.


www.example.com
media.example.com
AWS Cloud
media.example.com
www.example.com
AWS Shield
DDoS protection
AWS WAF
Web Application Firewall
Amazon Route 53
DNS service
Amazon CloudFront
CDN
AWS WAF
Web Application Firewall
Amazon S3
Static storage and backup
Availability Zone 1
Availability Zone 2
VPC
Public subnet
Nat gateway
Public subnet
Nat gateway
Elastic Load Balancing
Web Subnet
Web Subnet
EC2 Instances
Web servers
Auto Scaling group
Web tier
EC2 Instances
Web servers
Elastic Load Balancing
App Subnet
App Subnet
Amazon EFS
Shared storage
EC2 Instances
App servers
Auto Scaling group
App tier
EC2 Instances
App servers
DB Subnet
DB Subnet
DB subnet group
Amazon ElastiCache Service
Caching service
Amazon RDS
Database (primary)
Amazon RDS
Database (secondary)

Contoh arsitektur hosting web di AWS

1. Layanan DNS dengan Amazon Route 53 — Menyediakan layanan DNS untuk menyederhanakan pengelolaan domain.
2. Caching edge dengan Amazon CloudFront — Edge menyimpan konten bervolume tinggi untuk mengurangi latensi kepada pelanggan.
3. Keamanan edge untuk Amazon CloudFront dengan AWS WAF — Memfilter lalu lintas berbahaya, termasuk cross site scripting (XSS) dan injeksi SQL melalui aturan yang ditentukan pelanggan.
4. Penyeimbangan beban dengan Elastic Load Balancing (ELB) — Memungkinkan Anda menyebarkan beban di beberapa Zona Ketersediaan dan grup AWS Auto Scaling untuk redundansi dan decoupling layanan.
5. Perlindungan DDoS dengan AWS Shield — Lindungi infrastruktur Anda terhadap serangan DDoS jaringan dan lapisan transportasi yang paling umum secara otomatis.
6. Firewall dengan grup keamanan - Memindahkan keamanan ke instans untuk menyediakan firewall tingkat host yang stateful untuk server web dan aplikasi.
7. Caching dengan Amazon ElastiCache - Menyediakan layanan caching dengan Redis atau Memcached untuk menghapus beban dari aplikasi dan basis data, dan latensi yang lebih rendah untuk permintaan yang sering terjadi.
8. Basis data terkelola dengan Amazon Relational Database Service (Amazon RDS) — Menciptakan arsitektur basis data Multi-AZ yang sangat tersedia dengan enam mesin DB yang mungkin.
9. Penyimpanan dan pencadangan statis dengan Amazon Simple Storage Service (Amazon S3) — Memungkinkan penyimpanan objek berbasis HTTP sederhana untuk pencadangan dan aset statis seperti gambar dan video.

Komponen utama arsitektur hosting web AWS

Bagian berikut menguraikan beberapa komponen utama arsitektur hosting web yang di-deploy di AWS Cloud, dan menjelaskan perbedaannya dari arsitektur hosting web tradisional.

Manajemen jaringan

Di AWS Cloud, kemampuan untuk mengelompokkan jaringan Anda dari pelanggan lain memungkinkan arsitektur yang lebih aman dan dapat diskalakan. Meskipun grup keamanan menyediakan keamanan tingkat host (lihat bagian Keamanan Host), Amazon Virtual Private Cloud

(Amazon VPC) memungkinkan Anda meluncurkan sumber daya dalam jaringan virtual dan terisolasi secara logis yang Anda tetapkan.

Amazon VPC adalah layanan yang memberi Anda kontrol penuh atas detail pengaturan jaringan Anda di AWS. Contoh kontrol ini termasuk membuat subnet yang menghadap publik untuk server web, dan subnet privat tanpa akses internet untuk basis data Anda. Selain itu, Amazon VPC memungkinkan Anda membuat arsitektur hybrid dengan menggunakan jaringan pribadi virtual (VPN) perangkat keras, dan menggunakan AWS Cloud sebagai perpanjangan pusat data Anda sendiri.

Amazon VPC juga menyertakan dukungan IPv6 selain dukungan IPv4 tradisional untuk jaringan Anda.

Pengiriman konten

Ketika lalu lintas web Anda tersebar secara geografis, itu tidak selalu layak dan tentu saja tidak biaya efektif untuk mereplikasi seluruh infrastruktur Anda di seluruh dunia. Jaringan Pengiriman Konten (CDN) memberi Anda kemampuan untuk memanfaatkan jaringan global lokasi edge untuk mengirimkan salinan konten web yang di-cache seperti video, halaman web, gambar, dan sebagainya kepada pelanggan Anda. Untuk mengurangi waktu respons, CDN menggunakan lokasi edge terdekat dengan pelanggan atau lokasi permintaan yang berasal untuk mengurangi waktu respons. Throughput meningkat secara dramatis mengingat bahwa aset web dikirim dari cache. Untuk data dinamis, banyak CDN dapat dikonfigurasi untuk mengambil data dari server asal.

Anda dapat menggunakan CloudFront untuk mengirimkan situs web Anda, termasuk konten dinamis, statis, dan streaming, menggunakan jaringan global lokasi edge. CloudFront secara otomatis merutekan permintaan konten Anda ke lokasi edge terdekat, sehingga konten dikirimkan dengan performa terbaik. CloudFront dioptimalkan untuk bekerja dengan layanan AWS lainnya, seperti Amazon S3 dan Amazon Elastic Compute Cloud (Amazon EC2). CloudFront juga berfungsi lancar dengan server asal mana pun yang bukan server asal AWS, yang menyimpan versi asli file Anda.

Seperti layanan AWS lainnya, tidak ada kontrak atau komitmen bulanan untuk menggunakan CloudFront — Anda hanya membayar sebanyak atau sesedikit konten yang Anda berikan melalui layanan.

Selain itu, setiap solusi yang ada untuk caching edge dalam infrastruktur aplikasi web Anda harus bekerja dengan baik di AWS Cloud.

Mengelola DNS publik

Memindahkan aplikasi web ke AWS Cloud memerlukan beberapa perubahan Sistem Nama Domain (DNS). Untuk membantu Anda mengelola perutean DNS, AWS menyediakan Amazon Route 53, layanan web DNS cloud yang sangat tersedia dan dapat diskalakan. Route 53 dirancang untuk memberikan pengembang dan bisnis cara yang sangat tepercaya dengan biaya hemat untuk merutekan pengguna akhir ke aplikasi Internet dengan menerjemahkan nama seperti “www.example.com” ke alamat IP numerik, seperti 192.0.2.1, yang digunakan komputer agar saling terhubung. Amazon Route 53 juga sangat sesuai dengan IPv6.

Keamanan host

Selain pemfilteran lalu lintas jaringan masuk di edge, AWS juga merekomendasikan aplikasi web menerapkan pemfilteran lalu lintas jaringan di tingkat host. Amazon EC2 menyediakan fitur bernama grup keamanan. Grup keamanan analog dengan firewall jaringan masuk, yang mana Anda dapat menentukan protokol, port, dan rentang IP sumber yang diizinkan untuk mencapai instans EC2 Anda.

Anda dapat menetapkan satu atau lebih grup keamanan untuk setiap instans EC2. Setiap grup keamanan memungkinkan lalu lintas yang sesuai untuk setiap instans. Grup keamanan dapat dikonfigurasi sehingga hanya subnet, alamat IP, dan sumber daya tertentu yang memiliki akses ke instans EC2. Sebagai alternatif, mereka dapat mereferensikan grup keamanan lain untuk membatasi akses ke instans EC2 yang ada di grup tertentu.

Dalam arsitektur hosting web AWS pada Gambar 3, grup keamanan untuk klaster server web mungkin mengizinkan akses hanya dari Load Balancer lapisan web dan hanya melalui TCP pada port 80 dan 443 (HTTP dan HTTPS). Grup keamanan server aplikasi, di sisi lain, mungkin mengizinkan akses hanya dari Load Balancer lapisan aplikasi. Dalam model ini, teknisi dukungan Anda juga perlu mengakses instans EC2, apa yang dapat dicapai dengan Manajer Sesi AWS Systems Manager. Untuk diskusi lebih mendalam tentang keamanan, lihat AWS Cloud Security, yang berisi buletin keamanan, informasi sertifikasi, dan laporan resmi keamanan yang menjelaskan kemampuan keamanan AWS.

Penyeimbangan beban di seluruh klaster

Penyeimbang beban perangkat keras adalah alat jaringan umum yang digunakan dalam arsitektur aplikasi web tradisional. AWS menyediakan kemampuan ini melalui layanan Elastic Load Balancing (ELB). Mendistribusikan secara otomatis lalu lintas aplikasi yang akan datang pada seluruh target sekaligus, misalnya fungsi Instans Amazon EC2, kontainer, alamat IP, fungsi AWS Lambda, dan

peralatan virtual. Langkah ini dapat menangani berbagai beban lalu lintas aplikasi Anda di Zona Ketersediaan tunggal atau di beberapa Zona Ketersediaan. Elastic Load Balancing menawarkan empat jenis penyeimbang beban yang semuanya menghadirkan ketersediaan yang sangat baik, penskalaan otomatis, dan keamanan tangguh yang diperlukan untuk membuat aplikasi Anda toleran terhadap kesalahan.

Menemukan host dan layanan lain

Dalam arsitektur hosting web tradisional, sebagian besar host Anda memiliki alamat IP statis. Di AWS Cloud, sebagian besar host Anda memiliki alamat IP dinamis. Meskipun setiap instans EC2 dapat memiliki entri DNS publik dan pribadi dan akan dialamatkan melalui internet, entri DNS dan alamat IP ditetapkan secara dinamis ketika Anda meluncurkan instans. Mereka tidak dapat ditetapkan secara manual. Alamat IP statis (Alamat IP elastis dalam terminologi AWS) dapat ditetapkan ke instans yang berjalan setelah diluncurkan. Anda harus menggunakan alamat IP Elastic untuk instans dan layanan yang memerlukan titik akhir yang konsisten, seperti basis data utama, server file pusat, dan penyeimbang beban yang di-host EC2.

Caching dalam aplikasi web

Cache aplikasi dalam memori dapat mengurangi beban pada layanan dan meningkatkan performa dan skalabilitas pada tingkat basis data dengan caching informasi yang sering digunakan. Amazon ElastiCache adalah layanan web yang memudahkan deployment, pengoperasian, dan penskalaan cache dalam memori di cloud. Anda dapat mengonfigurasi cache dalam memori yang Anda buat untuk secara otomatis skala dengan beban dan untuk secara otomatis mengganti simpul yang gagal. ElastiCache mematuhi protokol dengan Memcached dan Redis, yang menyederhanakan migrasi dari solusi on-premise Anda saat ini.

Konfigurasi basis data, cadangan, dan failover

Banyak aplikasi web berisi beberapa bentuk persistensi, biasanya dalam bentuk basis datarelasional atau non-relasional. AWS menawarkan layanan basis data relasional dan non-relasional. Atau, Anda dapat menggunakan perangkat lunak basis data Anda sendiri pada instans EC2. Tabel berikut merangkum pilihan ini, yang dibahas secara lebih rinci di bagian ini.

Tabel 1 - Solusi basis data relasional dan non-relasional

	Solusi Basis Data Relasional	Solusi NoSQL
Layanan basis data terkelola	Amazon RDS for MySQL, Oracle, SQL Server, MariaDB, PostgreSQL, Amazon Aurora	Amazon DynamoDB , Amazon Keyspaces, Amazon Neptune, Amazon QLDB, Amazon Timestream
Dikelola sendiri	Hosting sistem manajemen basis data relasional (DBMS) pada instans Amazon EC2	Hosting solusi basis data non-relasional pada instans EC2

Amazon RDS

Amazon Relational Database Service (Amazon RDS) memberi Anda akses ke kemampuan mesin basis data MySQL, PostgreSQL, Oracle, dan Microsoft SQL Server yang sudah dikenal. Kode, aplikasi, dan alat yang sudah Anda gunakan dapat digunakan dengan Amazon RDS. Amazon RDS secara otomatis menambal perangkat lunak basis data dan mencadangkan basis data Anda, dan menyimpan cadangan untuk periode retensi yang ditentukan pengguna. Ini juga mendukung pemulihan titik waktu. Anda mendapatkan keuntungan dari fleksibilitas untuk dengan mudah menskalakan sumber daya komputasi atau kapasitas penyimpanan yang terkait dengan instans basis data Anda.

Penerapan Multi-AZ Amazon RDS meningkatkan ketersediaan basis data Anda dan melindungi basis data Anda dari pemadaman yang tidak direncanakan. Replika Baca Amazon RDS menyediakan replika baca-saja basis data Anda, sehingga Anda dapat meningkatkan skala di luar kapasitas penyebaran basis data tunggal untuk beban kerja basis data yang berat baca. Seperti semua layanan AWS, tidak diperlukan investasi di muka, dan Anda hanya membayar untuk sumber daya yang Anda gunakan.

Hosting sistem manajemen basis data relasional (RDBMS) pada instans Amazon EC2

Selain penawaran Amazon RDS terkelola, Anda dapat menginstal pilihan RDBMS (seperti MySQL, Oracle, SQL Server, atau DB2) pada instans EC2 dan mengelolanya sendiri. Pelanggan AWS yang meng-host basis data di Amazon EC2 berhasil menggunakan berbagai model primer/siaga dan replikasi, termasuk mirroring untuk salinan hanya-baca dan pengiriman log untuk budak pasif yang selalu siap.

Saat mengelola perangkat lunak basis data Anda sendiri secara langsung di Amazon EC2, Anda juga harus mempertimbangkan ketersediaan penyimpanan yang toleran terhadap kesalahan dan persisten. Untuk tujuan ini, kami menyarankan agar basis data yang berjalan di Amazon EC2 menggunakan volume Amazon Elastic Block Store (Amazon EBS), yang mirip dengan penyimpanan yang terpasang di jaringan.

Untuk instans EC2 yang menjalankan basis data, Anda harus menempatkan semua data basis data dan log pada volume EBS. Ini akan tetap tersedia bahkan jika host basis data gagal. Konfigurasi ini memungkinkan skenario failover sederhana, di mana instans EC2 baru dapat diluncurkan jika host gagal, dan volume EBS yang ada dapat dilampirkan ke instans baru. Basis data kemudian dapat diteruskan kembali.

Volume EBS secara otomatis memberikan redundansi dalam Zona Ketersediaan. Jika performa volume EBS tunggal tidak cukup untuk kebutuhan basis data Anda, volume dapat bergaris untuk meningkatkan performa operasi input/output per detik (IOPS) untuk basis data Anda.

Untuk beban kerja yang menuntut, Anda juga dapat menggunakan IOPS Provisioned EBS, di mana Anda menentukan IOPS yang diperlukan. Jika Anda menggunakan Amazon RDS, layanan mengelola penyimpanannya sendiri sehingga Anda dapat fokus mengelola data Anda.

Basis data non-relasional

Selain mendukung basis data relasional, AWS juga menawarkan sejumlah basis data non-relasional terkelola:

- Amazon DynamoDB adalah layanan basis data NoSQL yang dikelola penuh yang memberikan performa cepat dan dapat diprediksi dengan skalabilitas mulus. Menggunakan AWS Management Console atau DynamoDB API, Anda dapat menskalakan kapasitas naik atau turun tanpa waktu henti atau degradasi performa. Karena DynamoDB menangani beban administratif operasi dan penskalaan basis data terdistribusi ke AWS, Anda tidak perlu khawatir tentang penyediaan perangkat keras, penyiapan dan konfigurasi, replikasi, patch perangkat lunak, atau penskalaan klaster.
- Amazon DocumentDB (dengan kompatibilitas MongoDB) adalah layanan basis data yang dibangun khusus untuk manajemen data JSON sesuai skala, dikelola sepenuhnya, dan terintegrasi dengan AWS, serta siap digunakan untuk perusahaan dengan daya tahan tinggi.
- Amazon Keyspaces (untuk Apache Cassandra) adalah layanan basis data yang kompatibel dengan Apache Cassandra yang dapat diskalakan, sangat tersedia, dan terkelola. Dengan Amazon Keyspaces, Anda dapat menjalankan beban kerja Cassandra di AWS dengan kode aplikasi Cassandra dan alat pengembang yang sama dengan yang Anda gunakan saat ini.

- Amazon Neptune merupakan layanan basis data grafik yang cepat, andal, dan dikelola sepenuhnya yang memudahkan untuk membuat dan menjalankan aplikasi yang bekerja dengan data set yang selalu terhubung. Inti dari Amazon Neptune adalah mesin basis data grafik performa tinggi dan dibuat dengan tujuan khusus yang dioptimalkan untuk menyimpan miliaran hubungan dan melakukan kueri grafik dengan latensi milidetik.
- Amazon Quantum Ledger Database (Amazon QLDB) (QLDB) adalah basis data buku besar terkelola sepenuhnya yang menyediakan log transaksi transparan, tetap, dan dapat diverifikasi secara kriptografis milik otoritas pusat tepercaya. Amazon QLDB dapat digunakan untuk melacak setiap perubahan data aplikasi dan mempertahankan riwayat perubahan yang lengkap dan dapat diverifikasi dari waktu ke waktu.
- Amazon Timestream adalah layanan basis data deret waktu yang cepat, dapat diskalakan, dan nirserver untuk IoT dan aplikasi operasional yang memudahkan penyimpanan dan analisis triliunan peristiwa per hari hingga 1.000 kali lebih cepat dan dengan biaya sebesar 1/10 dari biaya basis data relasional.

Selain itu, Anda dapat menggunakan Amazon EC2 untuk meng-host teknologi basis data non-relasional lain yang mungkin Anda gunakan.

Penyimpanan dan cadangan data dan aset

Ada banyak opsi dalam AWS Cloud untuk menyimpan, mengakses, dan mencadangkan data dan aset aplikasi web Anda. Amazon S3 menyediakan toko objek yang sangat tersedia dan berlebihan. Amazon S3 adalah solusi penyimpanan yang bagus untuk objek yang agak statis atau lambat, seperti gambar, video, dan media statis lainnya. Amazon S3 juga mendukung caching tepi dan streaming aset ini dengan berinteraksi dengan CloudFront.

Untuk penyimpanan seperti sistem file terlampir, instans EC2 dapat memiliki volume EBS terpasang. Ini bertindak seperti disk terpasang untuk menjalankan instans EC2. Amazon EBS sangat bagus untuk data yang perlu diakses sebagai penyimpanan blok dan yang memerlukan ketekunan di luar masa pakai instans yang berjalan, seperti partisi basis data dan log aplikasi.

Selain memiliki masa pakai yang independen dari instans EC2, Anda dapat mengambil snapshot volume EBS dan menyimpannya di Amazon S3. Karena snapshot EBS hanya mencadangkan perubahan sejak snapshot sebelumnya, snapshot yang lebih sering dapat mengurangi waktu snapshot. Anda juga dapat menggunakan snapshot EBS sebagai garis dasar untuk mereplikasi data di beberapa volume EBS dan melampirkan volume tersebut ke instans berjalan lainnya.

Volume EBS bisa sebesar 16TB, dan beberapa volume EBS dapat bergaris untuk volume yang lebih besar atau untuk peningkatan performa input/output (I/O). Untuk memaksimalkan performa aplikasi I/O-intensif Anda, Anda dapat menggunakan volume IOPS Provisioned. Volume IOPS yang disediakan dirancang untuk memenuhi kebutuhan beban kerja I/O-intensif, terutama beban kerja basis data yang sensitif terhadap performa penyimpanan dan konsistensi dalam throughput I/O akses acak.

Anda menetapkan tarif IOPS saat membuat volume dan ketentuan Amazon EBS yang menilai seumur hidup volume. Amazon EBS saat ini mendukung IOPS per volume mulai dari maksimum 16000 (untuk semua tipe instans) hingga 64.000 (untuk instans yang dibuat pada Sistem Nitro). Anda dapat stripe beberapa volume bersama-sama untuk mengirimkan ribuan IOPS per instans ke aplikasi Anda. Selain itu, untuk beban kerja kritis throughput dan misi yang lebih tinggi yang membutuhkan latensi sub-milidetik, Anda dapat menggunakan tipe volume ekspres blok io2 yang dapat mendukung hingga 256.000 IOPS dengan kapasitas penyimpanan maksimum 64TB.

Secara otomatis menskalakan armada

Salah satu perbedaan utama antara arsitektur AWS Cloud dan model hosting tradisional adalah bahwa AWS dapat secara otomatis menskalakan armada aplikasi web sesuai permintaan untuk menangani perubahan lalu lintas. Dalam model hosting tradisional, model perkiraan lalu lintas umumnya digunakan untuk menyediakan host di depan lalu lintas yang diproyeksikan. Di AWS, instans dapat disediakan dengan cepat sesuai dengan satu set pemicu untuk menskalakan armada keluar dan kembali masuk.

Layanan Auto Scaling dapat membuat grup kapasitas server yang dapat tumbuh atau menyusut sesuai permintaan. Auto Scaling juga bekerja secara langsung dengan CloudWatch untuk data metrik dan dengan Elastic Load Balancing untuk menambah dan menghapus host untuk distribusi beban. Misalnya, jika server web melaporkan lebih besar dari 80 persen pemanfaatan CPU selama periode waktu tertentu, server web tambahan dapat dengan cepat di-deploy dan kemudian secara otomatis ditambahkan ke penyeimbang beban untuk segera dimasukkan dalam rotasi penyeimbangan beban.

Seperti yang ditunjukkan dalam model arsitektur hosting web AWS, Anda dapat membuat beberapa grup Auto Scaling untuk berbagai lapisan arsitektur, sehingga setiap lapisan dapat menskalakan secara independen. Misalnya, server web grup Auto Scaling mungkin memicu penskalaan masuk dan keluar dalam menanggapi perubahan dalam jaringan I/O, sedangkan server aplikasi Auto Scaling kelompok mungkin skala keluar dan sesuai dengan penggunaan CPU. Anda dapat mengatur minimum dan maksimal untuk membantu memastikan ketersediaan 24/7 dan untuk membatasi penggunaan dalam grup.

Pemicu Auto Scaling dapat diatur baik untuk tumbuh dan mengecilkan total armada pada lapisan tertentu untuk mencocokkan pemanfaatan sumber daya dengan permintaan aktual. Selain layanan Auto Scaling, Anda dapat menskalakan armada Amazon EC2 secara langsung melalui API Amazon EC2, yang memungkinkan peluncuran, penghentian, dan memeriksa instans.

Fitur keamanan tambahan

Jumlah dan kecanggihan serangan Distributed Denial of Service (DDoS) meningkat. Secara tradisional, serangan ini sulit ditangkis. Mereka sering berakhir mahal baik dalam waktu mitigasi dan daya yang dihabiskan, serta biaya kesempatan dari kunjungan yang hilang ke situs web Anda selama serangan. Ada sejumlah faktor dan layanan AWS yang dapat membantu Anda mempertahankan diri dari serangan tersebut. Salah satunya adalah skala jaringan AWS. Infrastruktur AWS cukup besar, dan memungkinkan Anda memanfaatkan skala kami untuk mengoptimalkan pertahanan Anda. Beberapa layanan, termasuk Elastic Load Balancing, Amazon CloudFront, dan Amazon Route 53, efektif untuk menskalakan aplikasi web Anda sebagai respons terhadap peningkatan lalu lintas yang besar.

Layanan perlindungan infrastruktur khususnya membantu dengan strategi pertahanan Anda:

- AWS Shield adalah layanan perlindungan DDoS yang dikelola yang membantu melindungi terhadap berbagai bentuk vektor serangan DDoS. Penawaran standar AWS Shield gratis dan aktif secara otomatis di seluruh akun Anda. Penawaran standar ini membantu mempertahankan diri dari serangan jaringan dan lapisan transportasi yang paling umum. Selain tingkat ini, penawaran lanjutan memberikan tingkat perlindungan yang lebih tinggi terhadap aplikasi web Anda dengan menyediakan visibilitas waktu nyata ke dalam serangan yang sedang berlangsung, serta mengintegrasikan pada tingkat yang lebih tinggi dengan layanan yang disebutkan sebelumnya. Selain itu, Anda mendapatkan akses ke AWS DDoS Response Team (DRT) untuk membantu mengurangi serangan skala besar dan canggih terhadap sumber daya Anda.
- AWS WAF (Web Application Firewall) dirancang untuk melindungi aplikasi web Anda dari serangan yang dapat membahayakan ketersediaan atau keamanan, atau mengkonsumsi sumber daya yang berlebihan. AWS WAF bekerja sejalan dengan CloudFront atau Application Load Balancer, bersama dengan aturan kustom Anda, untuk mempertahankan diri dari serangan seperti cross-site scripting, SQL injection, dan DDoS. Seperti kebanyakan layanan AWS, AWS WAF hadir dengan API berfitur lengkap yang dapat membantu mengotomatisasi pembuatan dan pengeditan aturan untuk instans AWS WAF Anda karena kebutuhan keamanan Anda berubah.
- AWS Firewall Manager adalah layanan manajemen keamanan yang memungkinkan Anda mengonfigurasi dan mengelola aturan firewall di seluruh akun dan aplikasi Anda di AWS

Organizations. Ketika aplikasi baru dibuat, AWS Firewall Manager memudahkan untuk membawa aplikasi dan sumber daya baru ke dalam kepatuhan dengan menegakkan seperangkat aturan keamanan yang umum.

Failover dengan AWS

Keuntungan utama lain dari AWS melalui hosting web tradisional adalah Zona Ketersediaan yang memberi Anda akses mudah ke lokasi deployment redundan. Zona Ketersediaan adalah lokasi berbeda-beda yang direkayasa untuk terisolasi dari kegagalan di Zona Ketersediaan lainnya. Mereka menyediakan konektivitas jaringan latensi rendah yang murah ke Zona Ketersediaan lain di Wilayah AWSyang sama. Seperti yang ditunjukkan diagram arsitektur hosting web AWS, AWS merekomendasikan agar Anda menerapkan host EC2 di beberapa Zona Ketersediaan untuk membuat aplikasi web Anda lebih toleran terhadap kesalahan.

Penting untuk memastikan bahwa ada ketentuan untuk memigrasikan titik akses tunggal di Zona Ketersediaan jika terjadi kegagalan. Misalnya, Anda harus menyiapkan basis data siaga di Zona Ketersediaan kedua sehingga persistensi data tetap konsisten dan sangat tersedia, bahkan selama skenario kegagalan yang tidak mungkin terjadi. Anda dapat melakukan ini di Amazon EC2 atau Amazon RDS dengan mengeklik tombol.

Meskipun beberapa perubahan arsitektur sering diperlukan saat memindahkan aplikasi web yang ada ke AWS Cloud, ada peningkatan signifikan terhadap skalabilitas, keandalan, dan efektivitas biaya yang membuat penggunaan AWS Cloud layak dilakukan. Bagian selanjutnya membahas perbaikan tersebut.

Pertimbangan utama saat menggunakan AWS untuk hosting web

Ada beberapa perbedaan utama antara AWS Cloud dan model hosting aplikasi web tradisional. Bagian sebelumnya menyoroti banyak area utama yang harus Anda pertimbangkan saat men-deploy aplikasi web ke cloud. Bagian ini menunjukkan beberapa pergeseran arsitektur utama yang perlu Anda pertimbangkan saat Anda membawa aplikasi apa pun ke cloud.

Tidak ada lagi peralatan jaringan fisik

Anda tidak dapat menerapkan peralatan jaringan fisik di AWS. Misalnya, firewall, router, dan penyeimbang beban untuk aplikasi AWS Anda tidak dapat lagi berada di perangkat fisik, tetapi harus diganti dengan solusi perangkat lunak. Ada berbagai macam solusi perangkat lunak berkualitas perusahaan, baik untuk penyeimbangan beban atau membangun koneksi VPN. Ini bukan batasan dari apa yang dapat dijalankan di AWS Cloud, tetapi ini adalah perubahan arsitektur untuk aplikasi Anda jika Anda menggunakan perangkat ini saat ini.

Firewall di mana-mana

Di mana Anda pernah memiliki zona demiliterisasi sederhana (DMZ) dan kemudian membuka komunikasi di antara host Anda dalam model hosting tradisional, AWS memberlakukan model yang lebih aman, di mana setiap host dikunci. Salah satu langkah dalam merencanakan deployment AWS adalah analisis lalu lintas antar host. Analisis ini akan memandu keputusan tentang apa port yang perlu dibuka. Anda dapat membuat grup keamanan untuk setiap jenis host dalam arsitektur Anda. Anda juga dapat membuat berbagai macam model keamanan sederhana dan berjenjang untuk memungkinkan akses minimum di antara host dalam arsitektur Anda. Penggunaan daftar kontrol akses jaringan dalam Amazon VPC dapat membantu mengunci jaringan Anda di tingkat subnet.

Pertimbangkan ketersediaan beberapa pusat data

Pikirkan Zona Ketersediaan dalam Wilayah AWS sebagai beberapa pusat data. Instans EC2 di Zona Ketersediaan berbeda dipisahkan secara logis dan fisik, dan mereka menyediakan model yang mudah digunakan untuk men-deploy aplikasi Anda di seluruh pusat data untuk ketersediaan dan keandalan yang tinggi. Amazon VPC sebagai layanan Regional memungkinkan Anda memanfaatkan Zona Ketersediaan sambil menyimpan semua sumber daya Anda dalam jaringan logis yang sama.

Perlakukan host sebagai sementara dan dinamis

Mungkin pergeseran yang paling penting dalam bagaimana Anda mungkin arsitek aplikasi AWS Anda adalah bahwa host Amazon EC2 harus dianggap sebagai fana dan dinamis. Setiap aplikasi yang dibuat untuk AWS Cloud tidak boleh berasumsi bahwa host akan selalu tersedia dan harus dirancang dengan pengetahuan bahwa data apa pun di penyimpanan instan EC2 akan hilang jika instans EC2 gagal.

Ketika host baru dibesarkan, Anda tidak boleh membuat asumsi tentang alamat IP atau lokasi dalam Zona Ketersediaan host. Model konfigurasi Anda harus fleksibel, dan pendekatan Anda untuk bootstrapping host harus memperhitungkan sifat dinamis cloud. Teknik-teknik ini sangat penting untuk membangun dan menjalankan aplikasi yang sangat mudah diskalakan dan toleran terhadap kesalahan.

Pertimbangkan kontainer dan nirserver

Laporan resmi ini terutama berfokus pada arsitektur web yang lebih tradisional. Namun, pertimbangkan untuk memodernisasi aplikasi web Anda dengan pindah ke Kontainer dan teknologi Nirserver , memanfaatkan layanan seperti AWS Fargate dan AWS Lambda untuk memungkinkan Anda untuk mengabstraksi penggunaan mesin virtual untuk melakukan tugas-tugas komputasi. Dengan komputasi nirserver, tugas manajemen infrastruktur seperti penyediaan kapasitas dan patching ditangani oleh AWS, sehingga Anda dapat membangun aplikasi yang lebih gesit yang memungkinkan Anda untuk berinovasi dan merespons perubahan lebih cepat.

Pertimbangkan deployment otomatis

- Amazon Lightsail adalah server privat virtual (VPS) yang mudah digunakan yang menawarkan kepada Anda semua yang diperlukan untuk membangun aplikasi atau situs web, ditambah dengan paket bulanan yang hemat biaya. Lightsail cocok untuk beban kerja sederhana, deployment cepat, dan memulai menggunakan AWS. Server ini dirancang untuk membantu Anda memulai dari yang kecil, lalu meningkat seiring pertumbuhan Anda.
- AWS Elastic Beanstalk adalah layanan yang mudah digunakan untuk men-deploy dan menyesuaikan skala aplikasi dan layanan web yang dikembangkan dengan Java, .NET, PHP, Node.js, Python, Ruby, Go, dan Docker pada server yang umum dikenal seperti Apache, NGINX, Passenger, dan IIS. Anda cukup mengunggah kode Anda dan Elastic Beanstalk secara otomatis akan menangani deployment, mulai dari penyediaan kapasitas, penyeimbangan beban, penskalaan otomatis, hingga pemantauan kondisi aplikasi. Pada saat yang sama, Anda tetap

memegang kendali penuh atas sumber daya AWS yang memberdayakan aplikasi Anda dan dapat mengakses sumber daya yang menjadi dasarnya kapan saja.

- AWS App Runner adalah layanan terkelola penuh yang memudahkan developer men-deploy aplikasi web dan API dalam kontainer dengan cepat, sesuai skala, dan tanpa memerlukan pengalaman infrastruktur sebelumnya. Mulailah dengan kode sumber Anda atau gambar kontainer. App Runner secara otomatis membangun dan men-deploy aplikasi web dan menyeimbangkan beban lalu lintas dengan enkripsi. App Runner juga meningkatkan atau turun secara otomatis untuk memenuhi kebutuhan lalu lintas Anda.
- AWS Amplify adalah serangkaian alat dan layanan yang dapat digunakan bersama atau terpisah, untuk membantu developer web dan seluler front-end membangun aplikasi tumpukan penuh yang dapat diskalakan dan didukung oleh AWS. Dengan Amplify, Anda dapat mengonfigurasi backend aplikasi dan menghubungkan aplikasi Anda dalam hitungan menit, menerapkan aplikasi web statis dalam beberapa klik, dan dengan mudah mengelola konten aplikasi di luar AWS Management Console.

Kesimpulan dan kontributor

Kesimpulan

Ada banyak pertimbangan arsitektur dan konseptual ketika Anda mempertimbangkan migrasi aplikasi web Anda ke AWS Cloud. Manfaat memiliki infrastruktur yang hemat biaya, sangat terukur, dan toleran terhadap kesalahan yang tumbuh dengan bisnis Anda jauh melampaui upaya migrasi ke AWS Cloud.

Kontributor

Individu dan organisasi berikut berkontribusi pada dokumen ini:

- Amir Khairalomoum, Arsitek Solusi Senior, AWS
- Dinesh Subramani, Arsitek Solusi Senior, AWS
- Jack Hemion, Arsitek Solusi Senior, AWS
- Jatin Joshi, Insinyur Dukungan Cloud, AWS
- Jorge Fonseca, Arsitek Solusi Senior, AWS
- Shinduri K S, Arsitek Solusi, AWS

Bacaan lebih lanjut

- Men-deploy aplikasi berbasis Django pada Amazon LightSail
- Men-deploy ketersediaan tinggi situs web Drupal ke Elastic Beanstalk
- Men-deploy aplikasi PHP ketersediaan tinggi ke Elastic Beanstalk
- Men-deploy aplikasi Node.js dengan DynamoDB ke Elastic Beanstalk
- Memulai Aplikasi Web Linux di AWS Cloud
- Meng-host Situs Web Statis
- Meng-host situs web statis menggunakan Amazon S3
- Tutorial: Men-deploy aplikasi inti ASP.NET dengan Elastic Beanstalk
- Tutorial: Cara men-deploy aplikasi sampel .NET dengan AWS Elastic Beanstalk

Revisi dokumen

Untuk mendapatkan notifikasi tentang pembaruan laporan resmi ini, sebaiknya berlangganan umpan RSS.

pembaruan-riwayat-perubahan	pembaruan-riwayat-deskripsi	pembaruan-riwayat-tanggal
Laporan resmi diperbarui	Beberapa bagian dan diagram diperbarui dengan layanan, fitur baru, dan batas layanan yang diperbarui.	20 Agustus 2021
Laporan resmi diperbarui	Label ikon yang diperbaru i untuk “Caching dengan ElastiCache” pada Gambar 3.	29 September 2019
Laporan resmi diperbarui	Beberapa bagian ditambahkan dan diperbarui untuk layanan baru. Diagram yang diperbaru i untuk kejelasan dan layanan tambahan. Penambahan VPC sebagai metode jaringan standar di AWS di “Manajeme n Jaringan.” Menambahk an bagian tentang perlindun gan dan mitigasi DDoS di “Fitur Keamanan Tambahan. ” Menambahkan bagian kecil pada arsitektur nirserver untuk hosting web.	1 Juli 2017
Laporan resmi diperbarui	Beberapa bagian diperbarui untuk meningkatkan kejelasan . Diagram yang diperbaru i untuk menggunakan ikon AWS. Penambahan bagian “Mengelola DNS Publik”	1 September 2012

	untuk detail di Amazon Route 53. Bagian “Menemukan Host dan Layanan Lainnya” diperbarui untuk kejelasan. Bagian “Konfigurasi Database, Cadangan, dan Failover” diperbarui untuk kejelasan dan DynamoDB. Bagian “Penyimpanan dan Pencadangan Data dan Aset” diperluas untuk mencakup volume EBS Provisioned IOPS.	
Publikasi awal	Laporan resmi dipublikasikan.	1 Mei 2010

Pemberitahuan

Dokumen ini disediakan hanya untuk tujuan informasi. Dokumen ini menunjukkan penawaran dan praktik produk AWS terbaru saat tanggal penerbitan dokumen ini, yang dapat berubah tanpa pemberitahuan. Pelanggan bertanggung jawab untuk membuat penilaian mandiri mereka sendiri terhadap informasi dalam dokumen ini dan setiap penggunaan produk atau layanan AWS, yang masing-masing diberikan "sebagaimana adanya" tanpa jaminan apa pun, baik tersurat maupun tersirat. Dokumen ini tidak membuat jaminan, pernyataan, komitmen kontraktual, ketentuan, atau kepastian dari AWS, afiliasi, pemasok, atau pemberi lisensinya. 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.