



Laporan Resmi AWS

Praktik Terbaik untuk WordPress di AWS



Praktik Terbaik untuk WordPress di AWS: Laporan Resmi AWS

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan produk Amazon tidak dapat digunakan sehubungan dengan produk atau layanan yang bukan milik Amazon, dengan segala cara yang mungkin menyebabkan kebingungan di antara pelanggan, atau dengan segala cara yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon adalah properti dari pemiliknya masing-masing, yang mungkin atau mungkin tidak berafiliasi dengan, berhubungan dengan, atau disponsori oleh Amazon.

Table of Contents

Abstrak	1
Abstrak	1
Pengantar	2
Deployment sederhana	3
Pertimbangan	3
Pendekatan yang tersedia	3
Amazon Lightsail	4
Memilih paket harga Amazon Lightsail	4
Menginstal WordPress	5
Memulihkan dari kegagalan	5
Meningkatkan performa dan efisiensi biaya	7
Mempercepat pengiriman konten	7
Pelepasan konten statis	8
Konten dinamis	8
Caching basis data	9
Caching bytecode	10
Deployment elastis	11
Arsitektur referensi	11
Menskalakan tingkat web	13
Tingkat web stateless	14
Penyimpanan bersama (Amazon S3 dan Amazon EFS)	15
Tingkat data (Amazon Aurora dan Amazon ElastiCache)	16
WordPress High Availability by Bitnami di AWS Quick Start	17
Kesimpulan	20
Kontributor	21
Revisi dokumen	22
Lampiran A: Konfigurasi CloudFront	23
Asal dan perilaku	23
Pembuatan distribusi CloudFront	23
Lampiran B: Penginstalan dan konfigurasi plugin	27
Plugin AWS for WordPress	27
Penginstalan dan konfigurasi plugin	27
Amazon CloudFront dan AWS Certificate Manager	29
Terjemahkan dan vokalisasikan konten Anda	29

Penyediaan podcast dengan Amazon Pollycast	32
Membacakan konten Anda melalui Perangkat Amazon Alexa	32
Konfigurasi konten statis	33
Pembuatan pengguna IAM	34
Pembuatan bucket Amazon S3	34
Pembuatan asal statis	35
Lampiran C: Pencadangan dan pemulihan	37
Lampiran D: Men-deploy plugin dan tema baru	39
Pemberitahuan	40

Praktik Terbaik untuk WordPress di AWS

Tanggal publikasi: 19 Oktober 2021 ([Revisi dokumen](#))

Abstrak

Laporan resmi ini memberikan panduan khusus kepada administrator sistem tentang cara memulai WordPress di Amazon Web Services (AWS) dan cara meningkatkan efisiensi biaya deployment serta pengalaman pengguna akhir. Laporan resmi ini juga menguraikan arsitektur referensi yang membahas skalabilitas umum dan persyaratan yang berketersediaan tinggi.

Pengantar

WordPress adalah alat pembuatan blog dan Content Management System (CMS) sumber terbuka berbasis PHP dan MySQL yang digunakan untuk mendukung segala hal mulai dari blog pribadi hingga situs web dengan lalu lintas tinggi.

Saat versi pertama WordPress diluncurkan pada tahun 2003, layanan ini tidak dibangun dengan mempertimbangkan infrastruktur berbasis cloud modern yang elastis dan dapat diskalakan. Melalui kontribusi dari komunitas WordPress dan peluncuran berbagai modul WordPress, kemampuan solusi CMS ini terus berkembang. Saat ini, layanan ini memungkinkan pembangunan arsitektur WordPress yang mengambil banyak keuntungan dari AWS Cloud.

Deployment sederhana

Untuk blog atau situs web dengan lalu lintas rendah tanpa persyaratan ketersediaan tinggi yang ketat, deployment sederhana satu server mungkin cocok. Deployment ini bukan arsitektur yang paling tangguh atau dapat diskalakan, tetapi ini adalah cara tercepat dan paling ekonomis untuk membuat situs web Anda beroperasi.

Topik

- [Pertimbangan](#)
- [Pendekatan yang tersedia](#)
- [Amazon Lightsail](#)

Pertimbangan

Diskusi ini dimulai dengan deployment server web tunggal. Mungkin pada akhirnya Anda akan tumbuh melampaui kapasitasnya, misalnya:

- Mesin virtual yang digunakan situs web WordPress Anda adalah satu titik kegagalan. Masalah yang terjadi pada mesin virtual ini akan mengakibatkan gangguan layanan untuk situs web Anda.
- Sumber daya penskalaan untuk meningkatkan performa hanya dapat dicapai dengan “penskalaan vertikal”, yaitu dengan meningkatkan ukuran mesin virtual yang menjalankan situs web WordPress Anda.

Pendekatan yang tersedia

AWS memiliki sejumlah opsi berbeda untuk penyediaan mesin virtual. Ada tiga cara utama untuk meng-host situs web WordPress Anda sendiri di AWS:

- Amazon Lightsail
- Amazon Elastic Compute Cloud (Amazon EC2)
- AWS Marketplace

[Amazon Lightsail](#) adalah layanan yang memungkinkan Anda meluncurkan server privat virtual dengan cepat (instans Lightsail) untuk meng-host situs web WordPress. Lightsail adalah cara

termudah untuk memulai jika Anda tidak memerlukan jenis instans yang sangat dapat dikonfigurasi atau akses ke fitur jaringan lanjutan.

[Amazon EC2](#) adalah layanan web yang menyediakan kapasitas komputasi yang dapat diubah ukurannya sehingga Anda dapat meluncurkan server virtual dalam hitungan menit. Amazon EC2 menyediakan lebih banyak opsi konfigurasi dan manajemen daripada Lightsail, yang biasanya diperlukan dalam arsitektur yang lebih canggih. Anda memiliki akses administratif ke instans EC2 Anda dan dapat menginstal paket perangkat lunak apa pun yang Anda pilih, termasuk WordPress.

[AWS Marketplace](#) adalah penyimpanan online tempat Anda dapat dengan cepat menemukan, membeli, dan men-deploy perangkat lunak yang berjalan di AWS. Anda dapat menggunakan deployment Sekali-Klik untuk meluncurkan image WordPress yang telah dikonfigurasi secara langsung ke Amazon EC2 di akun AWS Anda sendiri hanya dalam beberapa menit. Ada sejumlah vendor Marketplace yang menawarkan instans WordPress yang siap dijalankan.

Laporan resmi ini membahas opsi Lightsail sebagai implementasi yang direkomendasikan untuk situs web WordPress server tunggal.

Amazon Lightsail

Lightsail adalah cara termudah untuk mulai menggunakan AWS bagi developer, bisnis kecil, mahasiswa, dan pengguna lain yang membutuhkan solusi server privat virtual (VPS) sederhana.

Layanan ini mengambil alih banyak elemen yang lebih kompleks dalam manajemen infrastruktur dari pengguna. Oleh karena itu, layanan ini adalah titik awal yang ideal jika Anda memiliki pengalaman infrastruktur yang lebih sedikit, atau ketika Anda perlu fokus untuk menjalankan situs web Anda dan sebuah produk yang disederhanakan akan cukup untuk kebutuhan Anda.

Dengan Amazon Lightsail, Anda dapat memilih sistem operasi Windows atau Linux/Unix dan aplikasi web populer, termasuk WordPress, dan men-deploy-nya dengan sekali klik dari templat yang telah dikonfigurasi sebelumnya.

Saat kebutuhan Anda berkembang, Anda memiliki kemampuan untuk melampaui batas awal dengan lancar dan terhubung ke layanan basis data, penyimpanan objek, caching, dan distribusi konten AWS tambahan.

Memilih paket harga Amazon Lightsail

[Paket Lightsail](#) mendefinisikan biaya bulanan sumber daya Lightsail yang Anda gunakan untuk meng-host situs web WordPress Anda. Ada sejumlah paket yang tersedia untuk mencakup berbagai kasus

penggunaan, dengan berbagai tingkat sumber daya CPU, memori, penyimpanan Solid-state Drive (SSD), dan transfer data. Jika situs web Anda kompleks, Anda mungkin memerlukan instans yang lebih besar dengan lebih banyak sumber daya. Anda dapat mewujudkannya dengan memigrasikan server Anda ke paket yang lebih besar [menggunakan konsol web](#) atau seperti yang dijelaskan dalam [dokumentasi Amazon Lightsail CLI](#).

Menginstal WordPress

Lightsail menyediakan templat untuk aplikasi yang umum digunakan seperti WordPress. Templat ini adalah titik awal yang bagus untuk menjalankan situs web WordPress Anda sendiri karena telah diinstal sebelumnya dengan sebagian besar perangkat lunak yang Anda butuhkan. Anda dapat menginstal perangkat lunak tambahan atau menyesuaikan konfigurasi perangkat lunak dengan menggunakan terminal dalam browser atau klien SSH Anda sendiri, atau melalui antarmuka web administrasi WordPress.

Amazon Lightsail memiliki kemitraan dengan produk GoDaddy Pro Sites untuk membantu pelanggan WordPress mengelola instans mereka secara mudah dan gratis. Server virtual Lightsail WordPress telah dikonfigurasi sebelumnya dan dioptimalkan untuk performa dan keamanan yang cepat, sehingga mudah untuk membuat situs WordPress Anda beroperasi dalam waktu singkat. Pelanggan yang menjalankan sejumlah instans WordPress menganggap pekerjaan dalam memperbarui, memelihara, dan mengelola semua situs mereka itu sulit dan menyita waktu. Dengan integrasi ini, Anda dapat dengan mudah mengelola sejumlah instans WordPress Anda dalam hitungan menit hanya dengan beberapa klik.

Untuk informasi lebih lanjut tentang mengelola WordPress di Lightsail, lihat [Mulai menggunakan WordPress dari instans Amazon Lightsail Anda](#). Setelah selesai menyesuaikan situs WordPress Anda, kami sarankan untuk mengambil snapshot dari instans Anda.

[Snapshot](#) adalah cara untuk membuat image cadangan instans Lightsail Anda. Ini adalah salinan disk sistem dan juga menyimpan konfigurasi mesin asli (yaitu memori, CPU, ukuran disk, dan tingkat transfer data). Snapshot dapat digunakan untuk kembali ke konfigurasi yang diketahui baik setelah deployment atau upgrade yang salah.

Snapshot ini memungkinkan Anda memulihkan server Anda jika diperlukan, tetapi meluncurkan instans baru dengan penyesuaian yang sama.

Memulihkan dari kegagalan

Server web tunggal adalah satu titik kegagalan, jadi Anda harus memastikan bahwa data situs web Anda dicadangkan. Mekanisme snapshot yang dijelaskan sebelumnya juga dapat digunakan untuk

tujuan ini. Untuk memulihkan dari kegagalan, Anda dapat memulihkan instans baru dari snapshot terbaru Anda. Untuk mengurangi jumlah data yang dapat hilang selama pemulihan, snapshot Anda harus sebaru mungkin.

Untuk meminimalkan potensi kehilangan data, pastikan bahwa snapshot diambil secara rutin. Anda dapat menjadwalkan snapshot otomatis instans Lightsail Linux/Unix Anda. Untuk langkahnya, lihat [Mengaktifkan atau menonaktifkan snapshot otomatis untuk instans atau disk di Amazon Lightsail](#).

AWS merekomendasikan agar Anda menggunakan IP statis: alamat IP publik tetap yang dikhususkan untuk akun Lightsail Anda. Jika Anda perlu mengganti instans Anda dengan yang lain, Anda dapat menetapkan kembali IP statis ke instans baru. Dengan cara ini, Anda tidak perlu mengonfigurasi ulang sistem eksternal apa pun (seperti catatan DNS) untuk menunjuk ke alamat IP baru setiap kali Anda ingin mengganti instans Anda.

Meningkatkan performa dan efisiensi biaya

Pada akhirnya Anda mungkin akan tumbuh melampaui kapasitas deployment server tunggal Anda. Dalam hal ini, Anda mungkin perlu mempertimbangkan opsi untuk meningkatkan performa situs web Anda. Sebelum bermigrasi ke deployment multi-server yang dapat diskalakan (dibahas nanti dalam laporan ini), ada sejumlah performa dan efisiensi biaya yang dapat Anda terapkan. Ini adalah praktik yang baik yang harus Anda ikuti, bahkan jika Anda beralih ke arsitektur multi-server.

Bagian berikut memperkenalkan sejumlah opsi yang dapat meningkatkan aspek performa dan skalabilitas situs web WordPress Anda. Beberapa di antaranya dapat diterapkan untuk deployment server tunggal, sedangkan yang lain mengambil keuntungan dari skalabilitas sejumlah server. Banyak dari modifikasi tersebut memerlukan penggunaan satu atau beberapa plugin WordPress. Meskipun berbagai opsi tersedia, [W3 Total Cache](#) adalah pilihan populer yang menggabungkan banyak modifikasi tersebut dalam satu plugin.

Topik

- [Mempercepat pengiriman konten](#)
- [Caching basis data](#)
- [Caching bytecode](#)

Mempercepat pengiriman konten

Setiap situs WordPress perlu memberikan campuran konten statis dan dinamis. Konten statis mencakup gambar, file JavaScript, atau lembar gaya. Konten dinamis mencakup apa pun yang dihasilkan di sisi server menggunakan kode PHP WordPress, misalnya, elemen situs Anda yang dihasilkan dari basis data atau dipersonalisasi untuk setiap penampil (viewer).

Aspek penting dari pengalaman pengguna akhir adalah latensi jaringan yang terjadi saat mengirimkan konten sebelumnya kepada pengguna di seluruh dunia. Mempercepat pengiriman konten sebelumnya akan meningkatkan pengalaman pengguna akhir, terutama pengguna yang tersebar secara geografis di seluruh dunia. Hal ini dapat dicapai dengan Content Delivery Network (CDN) seperti Amazon CloudFront.

[Amazon CloudFront](#) adalah layanan web yang menyediakan cara mudah dan hemat biaya untuk mendistribusikan konten dengan latensi rendah dan kecepatan transfer data yang tinggi melalui sejumlah lokasi edge di seluruh dunia. Permintaan penampil (viewer) secara otomatis dialihkan

ke [lokasi edge](#) CloudFront yang sesuai untuk menurunkan latensi. Jika konten dapat di-cache (selama beberapa detik, menit, atau bahkan sehari-hari) dan sudah disimpan di lokasi edge tertentu, CloudFront segera mengirimkannya. Jika konten tidak boleh di-cache, telah kedaluwarsa, atau saat ini tidak berada di lokasi edge tersebut, CloudFront mengambil konten dari satu atau beberapa sumber kebenaran, yang disebut sebagai asal (dalam hal ini, instans Lightsail) dalam konfigurasi CloudFront. Pengambilan ini terjadi melalui koneksi jaringan yang dioptimalkan, yang berfungsi untuk mempercepat pengiriman konten di situs web Anda. Selain meningkatkan pengalaman pengguna akhir, model yang dibahas ini juga mengurangi beban pada server asal Anda dan memiliki potensi untuk menghasilkan penghematan biaya yang signifikan.

Pelepasan konten statis

Ini termasuk file CSS, JavaScript, dan image – baik file yang merupakan bagian dari tema WordPress Anda atau file media yang diunggah oleh administrator konten. Semua file ini dapat disimpan di Amazon Simple Storage Service (Amazon S3) menggunakan plugin seperti W3 Total Cache dan disajikan kepada pengguna dengan cara yang dapat diskalakan dan sangat tersedia. [Amazon S3](#) menawarkan infrastruktur penyimpanan data yang sangat dapat diskalakan, andal, dan berlatensi rendah dengan biaya rendah, yang dapat diakses melalui API REST. Amazon S3 menyimpan objek Anda secara redundan, tidak hanya di sejumlah perangkat, tetapi juga di berbagai fasilitas di Wilayah AWS, sehingga memberikan tingkat ketahanan yang sangat tinggi.

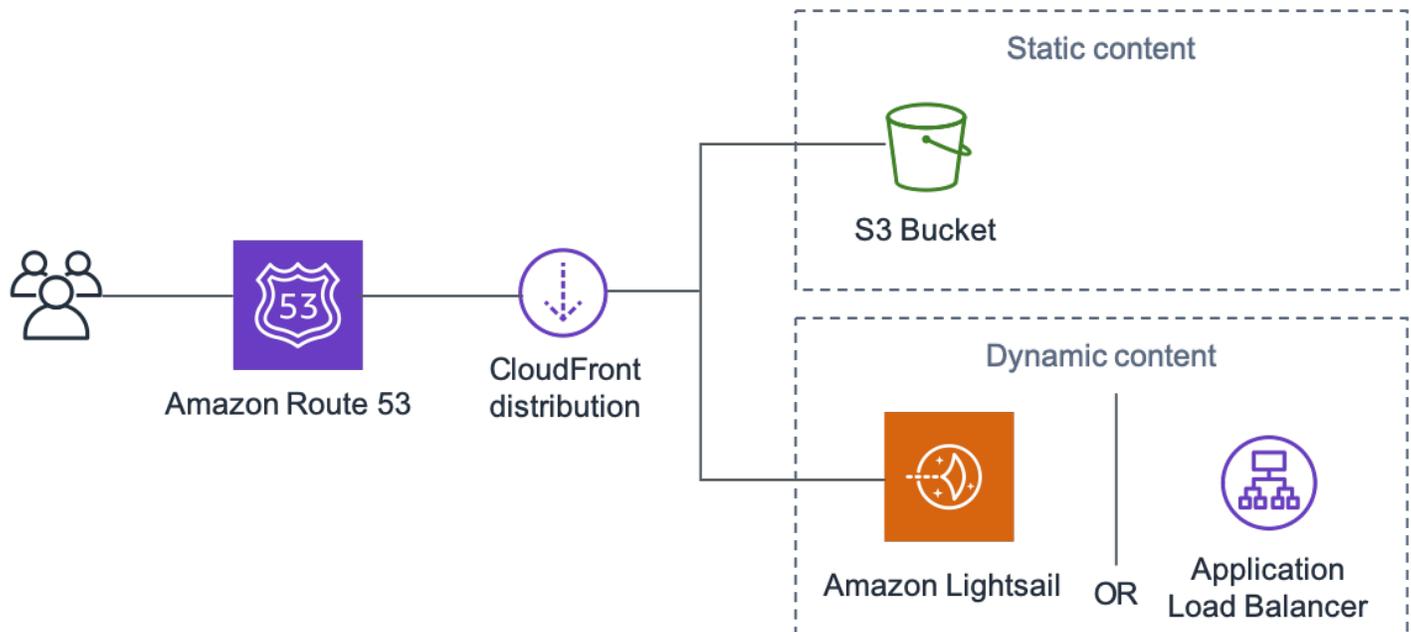
Hal tersebut memberikan efek samping positif dengan melepas beban kerja ini dari instans Lightsail Anda dan memungkinkannya berfokus pada pembuatan konten dinamis. Hal ini mengurangi beban pada server dan merupakan langkah penting untuk membuat arsitektur stateless (prasyarat sebelum menerapkan penskalaan otomatis).

Anda selanjutnya dapat mengonfigurasi Amazon S3 sebagai asal CloudFront untuk meningkatkan pengiriman aset statis tersebut ke pengguna di seluruh dunia. Meskipun WordPress tidak langsung terintegrasi dengan Amazon S3 dan CloudFront, berbagai plugin dapat menambahkan dukungan untuk layanan ini (misalnya, W3 Total Cache).

Konten dinamis

Konten dinamis mencakup output skrip PHP WordPress sisi server. Konten dinamis juga dapat disajikan melalui CloudFront dengan mengonfigurasi situs web WordPress sebagai asal. Karena konten dinamis menyertakan konten yang dipersonalisasi, Anda perlu mengonfigurasi CloudFront untuk meneruskan cookie HTTP dan header HTTP tertentu sebagai bagian dari permintaan ke server asal kustom Anda. CloudFront menggunakan nilai cookie yang diteruskan sebagai bagian dari kunci

yang mengidentifikasi objek unik dalam cache. Untuk memastikan bahwa Anda memaksimalkan efisiensi caching, Anda harus mengonfigurasi CloudFront untuk hanya meneruskan cookie HTTP dan header HTTP yang benar-benar memvariasikan konten (bukan cookie yang hanya digunakan di sisi klien atau oleh aplikasi pihak ketiga, misalnya, untuk analisis web).



Pengiriman seluruh situs web melalui Amazon CloudFront

Gambar sebelumnya berisi dua asal: satu untuk konten statis dan satu lagi untuk konten dinamis. Untuk detail implementasi, lihat [Lampiran A: Konfigurasi CloudFront](#) dan [Lampiran B: Penginstalan dan konfigurasi plugin](#).

CloudFront menggunakan header kontrol cache standar untuk mengidentifikasi apakah layanan ini harus meng-cache respons HTTP tertentu atau tidak dan durasinya. Header kontrol cache yang sama juga digunakan oleh browser web untuk memutuskan kapan dan berapa lama konten harus di-cache secara lokal untuk pengalaman pengguna akhir yang lebih optimal (misalnya, file `.css` yang sudah diunduh tidak akan diunduh ulang setiap kali pengunjung kembali melihat sebuah halaman). Anda dapat mengonfigurasi header kontrol cache pada tingkat server web (misalnya, melalui file `.htaccess` atau modifikasi `httpd.conf` file) atau menginstal plugin WordPress (misalnya, W3 Total Cache) untuk menentukan bagaimana header tersebut diatur untuk konten statis dan dinamis.

Caching basis data

Caching basis data dapat secara signifikan mengurangi latensi dan meningkatkan throughput untuk beban kerja aplikasi yang banyak melakukan pembacaan seperti WordPress. Performa aplikasi

ditingkatkan dengan menyimpan data yang sering diakses dalam memori untuk akses latensi rendah (misalnya, hasil kueri basis data dengan I/O tinggi). Jika sebagian besar kueri dilayani dari cache, jumlah kueri yang harus menjangkau basis data akan berkurang, sehingga mengurangi biaya yang terkait dengan menjalankan basis data.

Meskipun WordPress memiliki kemampuan caching terbatas yang siap pakai, berbagai plugin mendukung integrasi dengan [Memcached](#), sistem caching objek memori yang diadopsi secara luas. Plugin W3 Total Cache adalah contoh yang bagus.

Dalam skenario yang paling sederhana, Anda menginstal Memcached di server web Anda dan menangkap hasilnya sebagai snapshot baru. Dalam hal ini, Anda bertanggung jawab atas tugas administratif yang terkait dengan menjalankan cache.

Pilihan lainnya adalah memanfaatkan layanan terkelola seperti [Amazon ElastiCache](#) dan menghindari beban operasional tersebut. ElastiCache memudahkan untuk men-deploy, mengoperasikan, dan menskalakan cache dalam memori terdistribusi di cloud. Anda dapat menemukan informasi tentang cara terhubung ke node kluster ElastiCache Anda dalam [dokumentasi Amazon ElastiCache](#).

Jika Anda menggunakan Lightsail dan ingin mengakses kluster ElastiCache di akun AWS Anda secara privat, Anda dapat melakukannya dengan menggunakan peering VPC. Untuk petunjuk mengaktifkan peering VPC, lihat [Menyiapkan peering Amazon VPC untuk menggunakan sumber daya AWS di luar Amazon Lightsail](#).

Caching bytecode

Setiap kali skrip PHP dijalankan, skrip ini akan diurai dan dikompilasi. Dengan menggunakan cache bytecode PHP, output dari kompilasi PHP disimpan dalam RAM sehingga skrip yang sama tidak harus dikompilasi berulang kali. Hal ini mengurangi overhead yang terkait dengan mengeksekusi skrip PHP, sehingga menghasilkan performa yang lebih baik dan persyaratan CPU yang lebih rendah.

Cache bytecode dapat diinstal pada setiap instans Lightsail yang meng-host WordPress dan dapat sangat mengurangi bebannya. Untuk PHP 5.5 dan yang lebih baru, AWS merekomendasikan penggunaan [OPcache](#), ekstensi yang dibundel dengan versi PHP tersebut.

Perhatikan bahwa OPcache diaktifkan secara default di templat Bitnami WordPress Lightsail, jadi tidak ada tindakan lebih lanjut yang diperlukan.

Deployment elastis

Ada banyak skenario saat deployment server tunggal mungkin tidak cukup untuk situs web Anda. Dalam situasi ini, Anda memerlukan arsitektur multi-server yang dapat diskalakan.

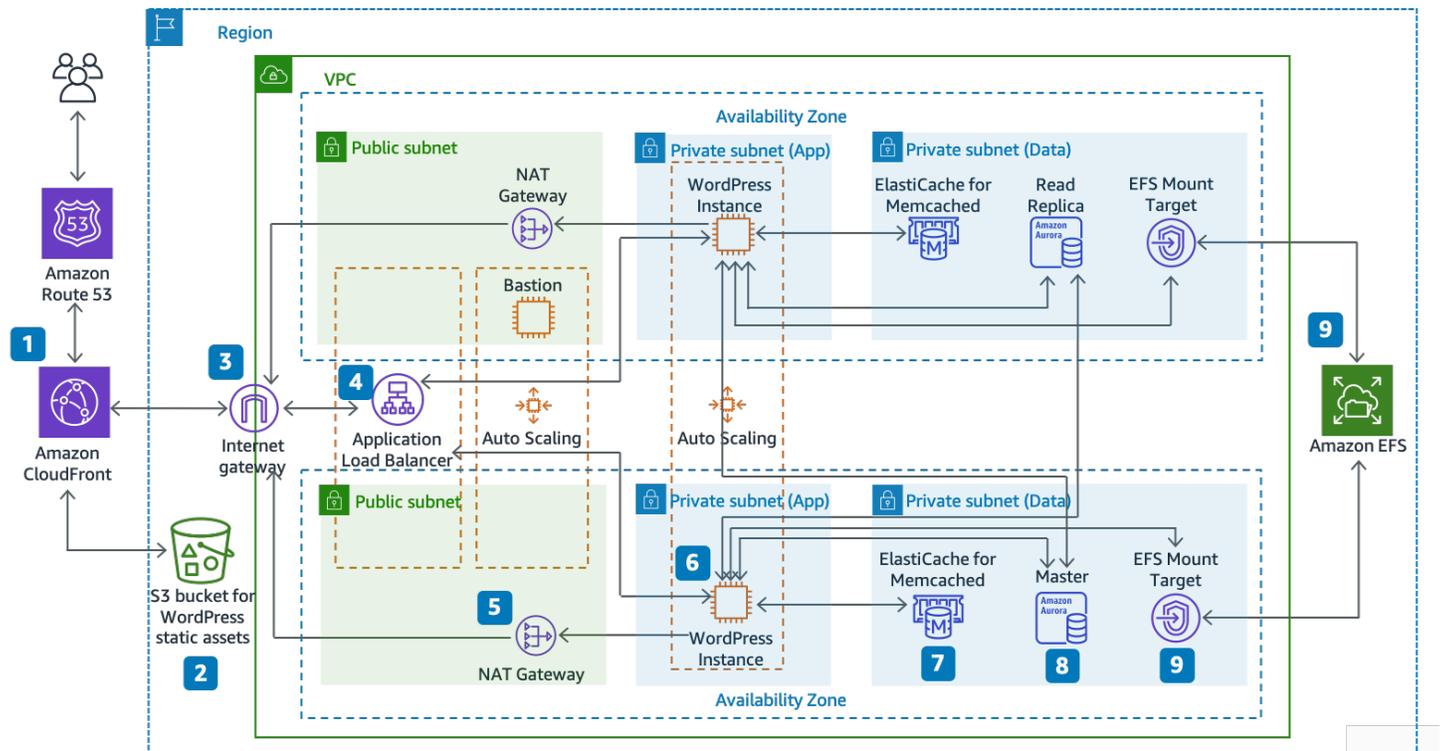
Topik

- [Arsitektur referensi](#)
- [Menskalakan tingkat web](#)
- [Tingkat web stateless](#)
- [WordPress High Availability by Bitnami di AWS Quick Start](#)

Arsitektur referensi

[Meng-host WordPress di arsitektur referensi AWS](#) yang tersedia di GitHub menguraikan praktik terbaik untuk men-deploy WordPress di AWS dan mencakup serangkaian templat AWS CloudFormation untuk membantu Anda beroperasi dengan cepat. Arsitektur berikut didasarkan pada arsitektur referensi tersebut. Selanjutnya, bagian ini akan meninjau alasan di balik pilihan arsitektur.

AMI dasar di GitHub diubah dari Amazon Linux1 menjadi Amazon Linux2 pada bulan Juli 2021. Namun, templat deployment di S3 belum berubah. Sebaiknya gunakan templat di GitHub jika ada masalah untuk men-deploy arsitektur referensi dengan templat di S3.



Arsitektur referensi untuk meng-host WordPress di AWS

Komponen arsitektur

Arsitektur referensi menggambarkan deployment praktik terbaik yang lengkap untuk situs web WordPress di AWS.

- Dimulai dengan caching edge di Amazon CloudFront (1) untuk menyimpan konten yang dekat dengan pengguna akhir untuk pengiriman yang lebih cepat.
- CloudFront menarik konten statis dari bucket S3 (2) dan konten dinamis dari Application Load Balancer (4) di depan instans web.
- Instans web ini berjalan dalam grup Auto Scaling pada instans Amazon EC2 (6).
- Kluster ElastiCache (7) meng-cache data yang sering dikueri untuk mempercepat respons.

Instans MySQL Amazon Aurora (8) meng-host basis data WordPress.

- Instans WordPress EC2 mengakses data WordPress bersama di sistem file Amazon EFS melalui EFS Mount Target (9) di setiap Zona Ketersediaan.
- Gateway Internet (3) memungkinkan komunikasi antara sumber daya di VPC Anda dan internet.
- Gateway NAT (5) di setiap Zona Ketersediaan memungkinkan instans EC2 di subnet privat (Aplikasi dan Data) mengakses internet.

Dalam Amazon VPC, terdapat dua jenis subnet: publik (Public Subnet) dan privat (App Subnet dan Data Subnet). Sumber daya yang di-deploy ke subnet publik akan menerima alamat IP publik dan akan terlihat secara publik di internet. Application Load Balancer (4) dan host Bastion untuk administrasi di-deploy di sini. Sumber daya yang di-deploy ke subnet privat hanya menerima alamat IP privat dan oleh karena itu, tidak terlihat secara publik di internet, sehingga meningkatkan keamanan sumber daya tersebut. Instans server web WordPress (6), instans klaster ElastiCache (7), instans basis data Aurora MySQL (8), dan EFS Mount Target (9) semuanya di-deploy di subnet privat.

Selanjutnya, bagian ini membahas masing-masing pertimbangan ini secara lebih mendetail.

Menskalakan tingkat web

Untuk mengembangkan arsitektur server tunggal Anda menjadi arsitektur multi-server yang dapat diskalakan, Anda harus menggunakan lima komponen utama:

- Instans Amazon EC2
- Amazon Machine Image (AMI)
- Penyeimbang beban
- Penskalaan otomatis
- Pemeriksaan kondisi

AWS menyediakan berbagai jenis instans EC2 sehingga Anda dapat memilih konfigurasi server terbaik untuk performa dan biaya. Secara umum, jenis instans yang dioptimalkan komputasi (misalnya, C4) mungkin merupakan pilihan yang baik untuk server web WordPress. Anda dapat men-deploy instans di sejumlah Zona Ketersediaan dalam Wilayah AWS untuk meningkatkan keandalan arsitektur keseluruhan.

Karena Anda memiliki kontrol penuh atas instans EC2 Anda, Anda dapat masuk dengan akses root untuk menginstal dan mengonfigurasi semua komponen perangkat lunak yang diperlukan untuk menjalankan situs web WordPress. Setelah selesai, Anda dapat menyimpan konfigurasi tersebut sebagai AMI, yang dapat Anda gunakan untuk meluncurkan instans baru dengan semua penyesuaian yang telah Anda buat.

Untuk mendistribusikan permintaan pengguna akhir ke sejumlah node server web, Anda memerlukan solusi penyeimbangan beban. AWS menyediakan kemampuan ini melalui [Elastic Load Balancing](#), layanan yang sangat tersedia yang mendistribusikan lalu lintas ke sejumlah instans EC2. Karena

situs web Anda menyajikan konten kepada pengguna melalui HTTP atau HTTPS, sebaiknya Anda menggunakan Application Load Balancer, penyeimbang beban lapisan aplikasi dengan perutean konten dan kemampuan untuk menjalankan sejumlah situs web WordPress pada domain yang berbeda-beda, jika diperlukan.

Elastic Load Balancing mendukung distribusi permintaan di sejumlah Zona Ketersediaan dalam Wilayah AWS. Anda juga dapat mengonfigurasi pemeriksaan kondisi sehingga Application Load Balancer secara otomatis berhenti mengirim lalu lintas ke instans individual yang gagal (misalnya, karena masalah perangkat keras atau crash perangkat lunak). AWS merekomendasikan penggunaan halaman login admin WordPress (`/wp-login.php`) untuk pemeriksaan kondisi karena halaman ini mengonfirmasikan bahwa server web sedang berjalan dan bahwa server web dikonfigurasi untuk menyajikan file PHP dengan benar.

Anda dapat memilih untuk membuat halaman pemeriksaan kondisi kustom yang memeriksa sumber daya dependen lainnya, seperti sumber daya basis data dan cache. Untuk informasi lebih lanjut, lihat [Pemeriksaan kondisi untuk grup target Anda](#) dalam Panduan Application Load Balancer.

Elastisitas adalah karakteristik utama dari AWS Cloud. Anda dapat meluncurkan lebih banyak kapasitas komputasi (misalnya, server web) saat Anda membutuhkannya dan menjalankan lebih sedikit saat tidak. [AWS Auto Scaling](#) adalah layanan AWS yang membantu Anda mengotomatisasi penyediaan ini untuk menaikkan atau menurunkan skala kapasitas Amazon EC2 Anda sesuai dengan kondisi yang Anda tentukan tanpa perlu intervensi manual. Anda dapat mengonfigurasi AWS Auto Scaling sehingga jumlah instans EC2 yang Anda gunakan meningkat secara lancar selama lonjakan permintaan untuk mempertahankan performa dan menurun secara otomatis saat lalu lintas berkurang, sehingga dapat meminimalkan biaya.

Elastic Load Balancing juga mendukung penambahan dan penghapusan dinamis host Amazon EC2 dari rotasi penyeimbangan beban. Elastic Load Balancing sendiri juga secara dinamis meningkatkan dan mengurangi kapasitas penyeimbangan beban untuk menyesuaikan dengan permintaan lalu lintas tanpa intervensi manual.

Tingkat web stateless

Untuk memanfaatkan sejumlah server web dalam konfigurasi penskalaan otomatis, tingkat web Anda harus bersifat stateless. Sebuah aplikasi stateless adalah aplikasi yang tidak perlu mengetahui interaksi sebelumnya dan tidak menyimpan informasi sesi. Dalam kasus WordPress, ini berarti bahwa semua pengguna akhir menerima respons yang sama, terlepas dari server web mana yang memproses permintaan mereka. Sebuah aplikasi stateless dapat diskalakan secara horizontal

karena setiap permintaan dapat dilayani oleh salah satu sumber daya komputasi yang tersedia (yaitu, instans server web). Ketika kapasitas itu tidak lagi diperlukan, setiap sumber daya individual dapat dihentikan dengan aman (setelah tugas yang berjalan di-draining). Sumber daya tersebut tidak perlu mengetahui keberadaan sumber daya lain – yang diperlukan hanyalah cara untuk mendistribusikan beban kerja ke sumber daya tersebut.

Dalam hal penyimpanan data sesi pengguna, inti WordPress benar-benar stateless karena bergantung pada cookie yang disimpan di browser web klien. Penyimpanan sesi tidak menjadi perhatian kecuali jika Anda telah menginstal kode kustom (misalnya, plugin WordPress) yang bergantung pada sesi PHP native.

Namun, WordPress awalnya dirancang untuk berjalan di satu server. Akibatnya, layanan ini menyimpan beberapa data pada sistem file lokal server. Saat menjalankan WordPress dalam konfigurasi multi-server, hal ini menciptakan masalah karena ada inkonsistensi di server web. Misalnya, jika pengguna mengunggah gambar baru, gambar itu hanya disimpan di salah satu server.

Ini menunjukkan mengapa kita perlu meningkatkan konfigurasi default WordPress yang berjalan untuk memindahkan data penting ke penyimpanan bersama. Arsitektur praktik terbaik memiliki basis data sebagai lapisan terpisah di luar server web dan memanfaatkan penyimpanan bersama untuk menyimpan unggahan, tema, dan plugin pengguna.

Penyimpanan bersama (Amazon S3 dan Amazon EFS)

Secara default, WordPress menyimpan unggahan pengguna pada sistem file lokal sehingga tidak stateless. Oleh karena itu, kita perlu memindahkan penginstalan WordPress dan semua penyesuaian pengguna (seperti konfigurasi, plugin, tema, dan unggahan yang dibuat pengguna) ke platform data bersama untuk membantu mengurangi beban pada server web dan menjadikan tingkat web ini bersifat stateless.

[Amazon Elastic File System](#) (Amazon EFS) menyediakan sistem file jaringan yang dapat diskalakan untuk digunakan dengan instans EC2. Sistem file Amazon EFS didistribusikan di seluruh server penyimpanan yang jumlahnya tak terbatas, sehingga memungkinkan sistem file berkembang dengan elastis hingga skala petabita dan memungkinkan akses paralel secara masif dari instans Amazon EC2. Desain terdistribusi Amazon EFS mencegah hambatan dan batasan yang biasanya terdapat di server file tradisional.

Dengan memindahkan seluruh direktori penginstalan WordPress ke sistem file EFS dan memasangnya ke setiap instans EC2 Anda saat boot, situs WordPress Anda dan semua datanya secara otomatis disimpan pada sistem file terdistribusi yang tidak tergantung pada salah satu

instans EC2, sehingga membuat tingkat web Anda sepenuhnya stateless. Manfaat dari arsitektur ini adalah Anda tidak perlu menginstal plugin dan tema pada setiap peluncuran instans baru, dan Anda dapat secara signifikan mempercepat penginstalan dan pemulihan instans WordPress. Hal ini juga mempermudah untuk men-deploy perubahan pada plugin dan tema di WordPress, seperti yang diuraikan dalam [Pertimbangan Deployment](#) bagian di dokumen ini.

Untuk memastikan performa optimal situs web Anda saat berjalan dari sistem file EFS, periksa pengaturan konfigurasi yang disarankan untuk Amazon EFS dan OPcache di [Arsitektur Referensi AWS untuk WordPress](#).

Anda juga memiliki opsi untuk melepas semua aset statis, seperti file gambar, CSS, dan JavaScript, ke bucket S3 dengan caching CloudFront di depan. Mekanisme untuk melakukan ini dalam arsitektur multi-server sama persis dengan arsitektur server tunggal, seperti yang dibahas di bagian [Konten Statis](#) dalam laporan resmi ini. Manfaatnya sama seperti dalam arsitektur server tunggal — Anda dapat melepas pekerjaan yang terkait dengan menyajikan aset statis Anda ke Amazon S3 dan CloudFront, sehingga memungkinkan server web Anda fokus menghasilkan konten dinamis saja dan melayani lebih banyak permintaan pengguna per server web.

Tingkat data (Amazon Aurora dan Amazon ElastiCache)

Dengan penginstalan WordPress yang disimpan pada sistem file jaringan bersama yang terdistribusi dan dapat diskalakan, serta aset statis yang disajikan dari Amazon S3, Anda dapat memusatkan perhatian Anda pada komponen stateful lainnya: basis data. Seperti dengan tingkat penyimpanan, basis data tidak boleh bergantung pada server tunggal, sehingga basis data tidak dapat di-host di salah satu server web. Sebagai gantinya, basis data WordPress harus di-host di Amazon Aurora.

[Amazon Aurora](#) adalah basis data relasional yang kompatibel dengan MySQL dan PostgreSQL yang dibangun untuk cloud, yang menggabungkan performa dan ketersediaan basis data komersial canggih dengan kesederhanaan dan penghematan biaya basis data sumber terbuka. Aurora MySQL meningkatkan performa dan ketersediaan MySQL dengan mengintegrasikan mesin basis data dengan sistem penyimpanan terdistribusi yang dibuat khusus dan didukung oleh SSD. Layanan ini toleran terhadap kesalahan dan dapat melakukan pemulihan secara otomatis, mereplikasi enam salinan data Anda di tiga Zona Ketersediaan, dirancang untuk ketersediaan lebih dari 99,99%, dan terus mencadangkan data Anda di Amazon S3. Amazon Aurora dirancang agar secara otomatis mendeteksi crash basis data dan memulai ulang tanpa perlu melakukan pemulihan crash atau membangun ulang cache basis data.

Amazon Aurora menyediakan sejumlah [jenis instans](#) yang sesuai dengan profil aplikasi yang berbeda-beda, termasuk instans yang dioptimalkan memori dan dapat menangani burst. Untuk

meningkatkan performa basis data Anda, Anda dapat memilih jenis instans besar untuk menyediakan lebih banyak sumber daya CPU dan memori.

Amazon Aurora secara otomatis menangani failover antara instans utama dan [Aurora Replica](#) sehingga aplikasi Anda dapat melanjutkan operasi basis data secepat mungkin tanpa intervensi administratif manual. Failover biasanya memakan waktu kurang dari 30 detik.

Setelah membuat setidaknya satu Aurora Replica, hubungkan ke instans utama Anda menggunakan titik akhir klaster untuk memungkinkan aplikasi Anda melakukan failover gagal secara otomatis jika instans utama gagal. Anda dapat membuat hingga 15 replika baca berlatensi rendah di tiga Zona Ketersediaan.

Seiring basis data diskalakan Anda, cache basis data Anda juga perlu diskalakan. Seperti yang dibahas sebelumnya di bagian [Caching Basis Data](#), ElastiCache memiliki fitur untuk menskalakan cache di sejumlah node dalam klaster ElastiCache, dan di sejumlah Zona Ketersediaan di sebuah Wilayah untuk meningkatkan ketersediaan. Saat menskalakan klaster ElastiCache Anda, pastikan Anda mengonfigurasi plugin caching Anda untuk terhubung menggunakan titik akhir konfigurasi sehingga WordPress dapat menggunakan node klaster baru saat node klaster ini ditambahkan dan berhenti menggunakan node klaster lama saat node klaster ini dihapus. Anda juga harus menyiapkan server web Anda untuk menggunakan [ElastiCache Cluster Client for PHP](#) dan memperbarui AMI Anda untuk menyimpan perubahan ini.

WordPress High Availability by Bitnami di AWS Quick Start

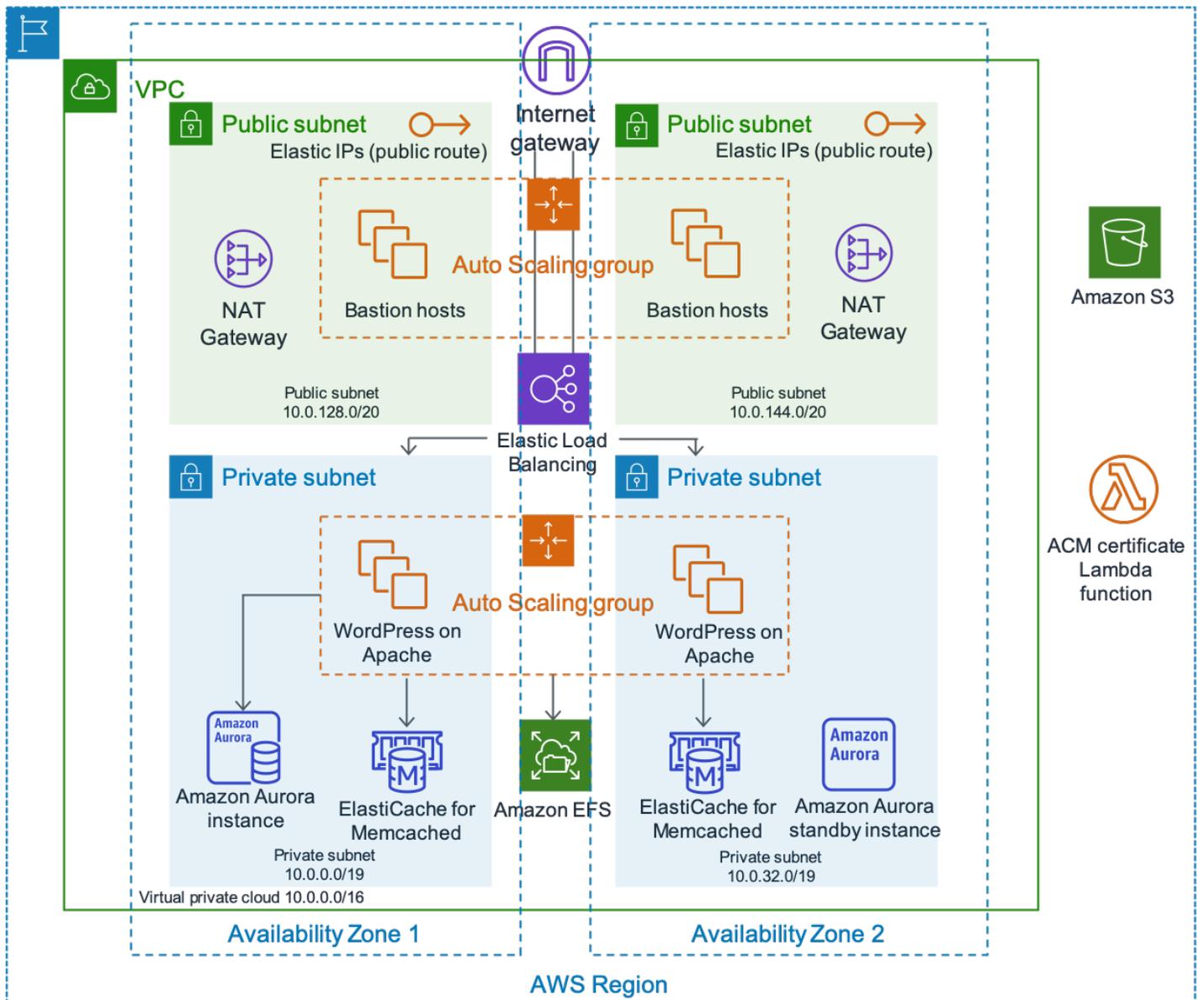
Quick Start dibuat oleh arsitek dan partner solusi AWS untuk membantu Anda men-deploy teknologi yang populer di AWS berdasarkan praktik terbaik AWS untuk keamanan dan ketersediaan yang sangat baik. Akselerator ini mengurangi ratusan prosedur manual menjadi hanya beberapa langkah, sehingga Anda dapat membangun lingkungan dan produksi Anda dengan cepat dan mulai menggunakannya dengan segera. Tiap Quick Start mencakup templat AWS CloudFormation yang mengotomatiskan deployment dan panduan yang membahas arsitektur dan memuat petunjuk deployment langkah demi langkah.

[WordPress High Availability by Bitnami di AWS Quick Start](#) menyiapkan lingkungan yang dapat dikonfigurasi berikut di AWS:

- Arsitektur yang sangat tersedia yang mencakup dua Zona Ketersediaan.*
- Virtual Private Cloud (VPC) yang dikonfigurasi dengan subnet publik dan privat sesuai dengan praktik terbaik AWS. Hal ini menyediakan infrastruktur jaringan untuk deployment Anda.*

- Gateway internet untuk menyediakan akses ke internet. Gateway ini digunakan oleh host bastion untuk mengirim dan menerima lalu lintas.*
- Dalam subnet publik, gateway NAT terkelola bertujuan untuk memungkinkan akses internet keluar untuk sumber daya di subnet privat.*
- Di subnet publik, host bastion Linux dalam grup Auto Scaling untuk mengizinkan akses Secure Shell (SSH) masuk ke instans EC2 di subnet publik dan privat.*
- Elastic Load Balancing untuk mendistribusikan permintaan HTTP dan HTTPS di sejumlah contoh WordPress.
- Di subnet privat, instans EC2 yang meng-host aplikasi WordPress di Apache. Instans ini disediakan dalam grup Auto Scaling untuk memastikan ketersediaan tinggi.
- Dalam subnet privat, instans Amazon Aurora DB yang dikelola oleh Amazon Relational Database Service (Amazon RDS).
- Di subnet privat, Amazon Elastic File System (Amazon EFS) untuk berbagi aset (seperti plugin, tema, dan image) di instans WordPress.
- Di subnet privat, node Amazon ElastiCache for Memcached untuk caching kueri basis data.

* Templat yang men-deploy Quick Start ke VPC yang ada akan melewati tugas yang ditandai dengan tanda bintang dan meminta konfigurasi VPC Anda yang ada.



Arsitektur WordPress High Availability by Bitnami

Penjelasan mendetail tentang deployment WordPress High Availability by Bitnami di AWS berada di luar cakupan dokumen ini. Untuk konfigurasi dan opsi, lihat [WordPress High Availability by Bitnami di AWS](#).

Kesimpulan

AWS menyajikan banyak opsi arsitektur untuk menjalankan WordPress. Pilihan paling sederhana adalah penginstalan server tunggal untuk situs web berlalu lintas rendah. Untuk situs web yang lebih canggih, administrator situs dapat menambahkan beberapa opsi lain, yang masing-masing merepresentasikan peningkatan tambahan dalam hal ketersediaan dan skalabilitas. Administrator dapat memilih fitur yang paling sesuai dengan kebutuhan dan anggaran mereka.

Kontributor

Kontributor dokumen ini meliputi:

- Paul Lewis, Arsitek Solusi (Solutions Architect) , Amazon Web Services
- Ronan Guilfoyle, Arsitek Solusi (Solutions Architect) , Amazon Web Services
- Andreas Chatzakis, Manajer Arsitek Solusi (Solutions Architect Manager), Amazon Web Services
- Jibril Touzi, Manajer Akun Teknis (Technical Account Manager), Amazon Web Services
- Hakmin Kim, Arsitek Solusi Partner Migrasi (Migration Partner Solutions Architect) , Amazon Web Services

Revisi dokumen

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

update-history-change	update-history-description	update-history-date
Laporan resmi diperbarui	Diperbarui untuk memodifikasi Arsitektur Referensi dan plugin AWS for WordPress.	19 Oktober 2021
Laporan resmi diperbarui	Diperbarui untuk menyertakan pendekatan deployment baru dan plugin AWS for WordPress.	30 Oktober 2019
Laporan resmi diperbarui	Diperbarui untuk memperjelas pesan produk Amazon Aurora.	1 Februari 2018
Laporan resmi diperbarui	Diperbarui untuk menyertakan layanan AWS yang diluncurkan sejak publikasi pertama.	1 Desember 2017
Publikasi awal	Pertama kali dipublikasikan.	1 Desember 2014

Lampiran A: Konfigurasi CloudFront

Untuk mendapatkan performa dan efisiensi yang optimal saat menggunakan Amazon CloudFront dengan situs web WordPress Anda, penting untuk mengonfigurasi situs web dengan benar untuk berbagai jenis konten yang disajikan.

Topik

- [Asal dan perilaku](#)
- [Pembuatan distribusi CloudFront](#)

Asal dan perilaku

[Asal](#) adalah lokasi yang memungkinkan CloudFront mengirimkan permintaan untuk konten yang didistribusikan melalui lokasi edge. Tergantung pada implementasi Anda, Anda dapat memiliki satu atau dua asal. Satu untuk konten dinamis (instans Lightsail dalam [opsi deployment server tunggal](#), atau Application Load Balancer dalam [opsi deployment elastis](#)) menggunakan asal kustom. Anda boleh memiliki asal kedua untuk mengarahkan CloudFront ke konten statis Anda. Dalam [arsitektur referensi](#) sebelumnya, ini adalah bucket S3. Saat menggunakan Amazon S3 sebagai asal untuk distribusi, Anda harus menggunakan [kebijakan bucket](#) agar konten dapat diakses publik.

[Perilaku](#) memungkinkan Anda menetapkan aturan yang mengatur cara CloudFront meng-cache konten Anda, dan, pada gilirannya, menentukan seberapa efektif cache tersebut. Perilaku memungkinkan Anda mengontrol protokol dan siapa yang dapat mengakses metode HTTP situs web Anda. Perilaku juga memungkinkan Anda mengontrol apakah akan meneruskan header HTTP, cookie, atau string kueri ke backend Anda (dan, jika demikian, yang mana). Perilaku berlaku untuk pola jalur URL tertentu.

Pembuatan distribusi CloudFront

Buat distribusi web CloudFront dengan mengikuti distribusi, dan asal dan perilaku default yang dibuat secara otomatis akan digunakan untuk konten dinamis. Buat empat perilaku tambahan untuk lebih menyesuaikan cara permintaan statis dan dinamis diperlakukan. Tabel berikut merangkum properti konfigurasi untuk lima perilaku. Anda juga dapat melewati konfigurasi manual ini dan menggunakan plugin AWS for WordPress yang dibahas dalam [Lampiran B: Penginstalan dan Konfigurasi Plugin](#), ini adalah cara termudah untuk mengonfigurasi CloudFront guna mempercepat situs WordPress Anda.

Tabel 1: Ringkasan properti konfigurasi untuk perilaku CloudFront

Properti	Statis	Dinamis (admin)	Dinamis (front end)
Jalur (Perilaku)	wp-content/* wp-includes/*	wp-admin/* wp-login.php	default (*)
Protokol	HTTP dan HTTPS	Pengalihan ke HTTPS	HTTP dan HTTPS
Metode HTTP	GET, HEAD	SEMUA	SEMUA
Header HTTP	TIDAK ADA	SEMUA	Host CloudFront-Forwarded-Proto CloudFront-Is-Mobile-Viewer CloudFront-Is-Tablet-Viewer CloudFront-Is-Desktop-Viewer
Cookie	TIDAK ADA	SEMUA	comment_* wordpress_* wp-settings-*
String Kueri	YA (invalidasi)	YA	YA

Untuk perilaku default, AWS merekomendasikan konfigurasi berikut:

- Atur Origin Protocol Policy (Kebijakan Protokol Asal) ke Match Viewer (Cocokkan dengan Penampil), sehingga jika penampil (viewer) terhubung ke CloudFront menggunakan HTTPS, CloudFront juga akan terhubung ke asal Anda menggunakan HTTPS, sehingga mewujudkan

enkripsi ujung ke ujung. Perhatikan bahwa hal ini mengharuskan Anda menginstal sertifikat SSL tepercaya pada penyeimbang beban. Untuk detailnya, lihat [Mewajibkan HTTPS untuk Komunikasi Antara CloudFront dan Asal Kustom Anda](#).

- Izinkan semua metode HTTP karena bagian dinamis situs web memerlukan permintaan GET dan POST (misalnya, untuk mendukung POST untuk formulir pengiriman komentar).
- Teruskan hanya cookie yang memvariasikan output WordPress; misalnya, `>wordpress_*`, `wp-settings-*`, dan `comment_*`. Anda harus memperpanjang daftar tersebut jika Anda telah menginstal plugin yang bergantung pada cookie lain yang tidak ada dalam daftar.
- Teruskan hanya header HTTP yang memengaruhi output WordPress, misalnya, `Host`, `CloudFront-Forwarded-Proto`, `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer`, dan `CloudFront-is-Tablet-Viewer`:
 - `Host` memungkinkan sejumlah situs web WordPress di-host pada asal yang sama.
 - `CloudFront-Forwarded-Proto` memungkinkan versi halaman yang berbeda-beda di-cache tergantung pada apakah halaman ini diakses melalui HTTP atau HTTPS.
 - `CloudFront-is-Desktop-Viewer`, `CloudFront-is-Mobile-Viewer`, `CloudFront-is-Tablet-Viewer` memungkinkan Anda menyesuaikan output tema Anda berdasarkan jenis perangkat pengguna akhir.
- Teruskan semua string kueri ke cache berdasarkan nilainya karena WordPress bergantung pada string ini. String ini juga dapat digunakan untuk menginvalidasi objek cache.

Jika Anda ingin menyajikan situs web Anda dengan nama domain kustom (yaitu, bukan `*.cloudfront.net`), maka masukkan URI yang sesuai di bagian Alternate Domain Names (Nama Domain Alternatif) di Pengaturan Distribusi. Dalam hal ini, Anda juga memerlukan sertifikat SSL untuk nama domain kustom Anda. Anda dapat [meminta](#) sertifikat SSL melalui AWS Certificate Manager dan mengonfigurasinya berdasarkan distribusi CloudFront.

Sekarang, buat dua perilaku cache lagi untuk konten dinamis: satu untuk halaman login (pola jalur: `wp-login.php`) dan satu untuk dasbor admin (pola jalur: `wp-admin/*`). Kedua perilaku ini memiliki pengaturan yang sama persis sebagai berikut:

- Berlakukan Kebijakan Protokol Penampil HTTPS Saja.
- Izinkan semua metode HTTP.
- Cache berdasarkan semua header HTTP.
- Teruskan semua cookie.
- Teruskan dan cache berdasarkan semua string kueri.

Alasan di balik konfigurasi ini adalah bahwa bagian situs web ini sangat dipersonalisasi dan biasanya hanya memiliki hanya beberapa pengguna, sehingga efisiensi caching bukanlah perhatian utama. Fokusnya adalah menjaga konfigurasi tetap sederhana untuk memastikan kompatibilitas maksimum dengan plugin yang terinstal dengan meneruskan semua cookie dan header ke asal.

[Plugin AWS for WordPress](#) yang dibahas dalam Lampiran B akan secara otomatis membuat distribusi CloudFront yang memenuhi konfigurasi sebelumnya.

Secara default, WordPress menyimpan semuanya secara lokal di server web, yaitu penyimpanan blok (Amazon EBS) untuk [deployment server tunggal](#), dan penyimpanan file (Amazon EFS) untuk [deployment elastis](#). Selain mengurangi biaya penyimpanan dan transfer data, memindahkan aset statis ke Amazon S3 menawarkan skalabilitas, ketersediaan data, keamanan, dan performa. Ada beberapa plugin yang memudahkan untuk memindahkan konten statis ke Amazon S3; salah satunya adalah [W3 Total Cache](#), yang juga dibahas dalam [Lampiran B: Penginstalan dan konfigurasi plugin](#).

Lampiran B: Penginstalan dan konfigurasi plugin

Topik

- [Plugin AWS for WordPress](#)
- [Konfigurasi konten statis](#)

Plugin AWS for WordPress

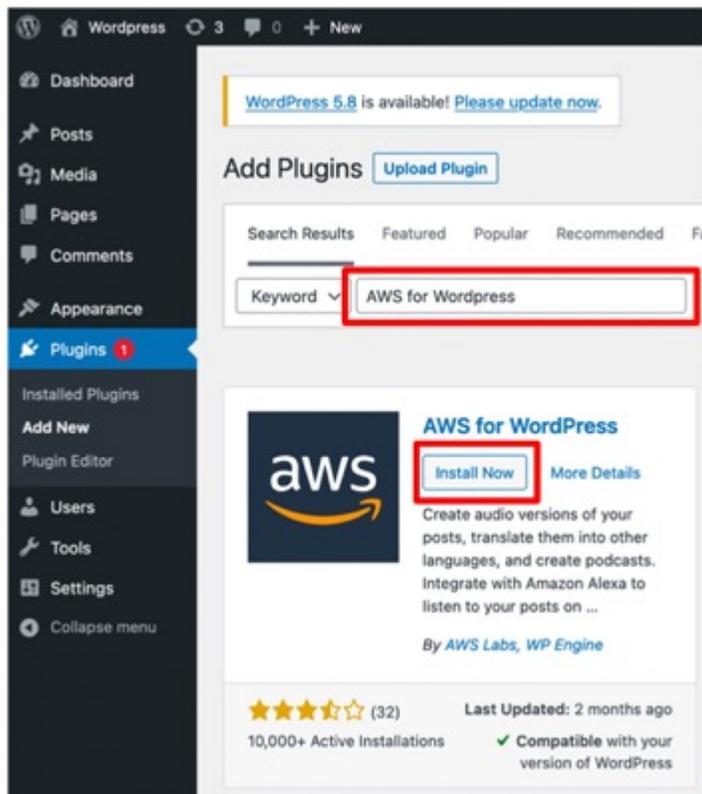
Plugin AWS for WordPress adalah satu-satunya plugin WordPress yang ditulis dan dikelola secara aktif oleh AWS. Plugin ini memungkinkan pelanggan mengonfigurasi [Amazon CloudFront](#) dan [AWS Certificate Manager \(ACM\)](#) dengan mudah ke situs web WordPress untuk meningkatkan performa dan keamanan. Plugin ini menggunakan layanan [Amazon Machine Learning](#) untuk menerjemahkan konten ke dalam satu atau beberapa bahasa, menghasilkan versi audio dari setiap terjemahan, dan membacakan situs web WordPress melalui perangkat Amazon Alexa.

Plugin ini sudah terinstal di [WordPress High Availability by Bitnami di AWS Quick Start](#).

Penginstalan dan konfigurasi plugin

Untuk menginstal plugin:

1. Untuk menggunakan plugin AWS for WordPress, Anda harus membuat pengguna IAM untuk plugin ini. Pengguna IAM adalah orang atau aplikasi dalam akun AWS yang memiliki izin untuk melakukan panggilan API ke layanan AWS.
2. Anda memerlukan [AWS Identity and Access Management \(IAM\)](#) role atau pengguna IAM untuk mengontrol autentikasi dan otorisasi untuk akun AWS Anda. Untuk mencegah pengguna yang tidak sah mendapatkan izin ini, lindungi kredensial pengguna IAM. Perlakukan secret access key ini seperti kata sandi; simpan di tempat yang aman, dan jangan bagikan dengan siapa pun. Seperti kata sandi, [rotasikan access key secara berkala](#). Jika secret access key tidak sengaja bocor, [hapus segera](#). Kemudian Anda dapat membuat access key baru untuk digunakan dengan plugin AWS for WordPress.
3. Di menu Plugins (Plugin) pada panel admin WordPress, cari AWS for WordPress dan pilih Install Now (Instal Sekarang).



4. Jika penginstalan plugin tidak berfungsi, mungkin ada masalah izin pengguna. Hubungkan ke server web WordPress dan selesaikan petunjuk berikut untuk memecahkan masalah ini.
- Buka file `wp-config.php` di direktori penginstalan WordPress dan tulis kode berikut di bagian akhir file `wp-config.php` ini:

```
define('FS_METHOD','direct');
```

- Luncurkan perintah berikut untuk memberikan izin tulis:

```
chmod 777 <WordPress install directory>/wp-content
```

Peringatan: Tetap menggunakan izin tulis 777 akan berisiko. Jika izin 777 tidak diubah, siapa pun dapat mengedit atau menghapus folder ini. Ubah izin tulis menjadi 755 atau di bawahnya setelah menyelesaikan pekerjaan plugin.

- Jika arsitektur referensi digunakan, direktori penginstalan WordPress adalah ``/var/www/wordpress/<site directory>``.

Penjelasan mendetail tentang semua pengaturan AWS for WordPress berada di luar cakupan dokumen ini. Untuk konfigurasi dan opsi, lihat [Mulai menggunakan plugin AWS for WordPress](#).

Amazon CloudFront dan AWS Certificate Manager

Untuk menyiapkan CloudFront dan AWS Certificate Manager:

1. Pada menu plugin, pilih CloudFront dan masukkan parameter berikut:
 - Origin domain name (Nama domain asal): Domain DNS server asal HTTP tempat CloudFront mendapatkan konten situs web Anda (seperti `example.com`).
 - Alternate domain name (CNAME) (Nama domain alternatif (CNAME)): Nama domain yang digunakan pengunjung Anda untuk pengalaman situs web yang dipercepat. AWS merekomendasikan penggunaan 'www' di depan domain (seperti `www.example.com`).
2. Pilih Initiate Setup (Inisiasikan Penyiapan) untuk memulai konfigurasi.

Plugin ini secara otomatis akan meminta sertifikat SSL untuk CNAME melalui ACM, dan setelah Anda memvalidasi token ACM dengan [memperbarui catatan DNS](#) menggunakan entri CNAME, plugin ini akan membuat distribusi CloudFront yang memenuhi praktik terbaik yang didefinisikan dalam [Lampiran A](#).

Note

Plugin AWS for WordPress memerlukan HTTPS untuk komunikasi antara CloudFront dan asal kustom Anda. Pastikan asal Anda memiliki sertifikat SSL yang valid untuk nama domain Asal. Untuk informasi selengkapnya, lihat [Mewajibkan HTTPS untuk komunikasi antara CloudFront dan asal kustom Anda](#).

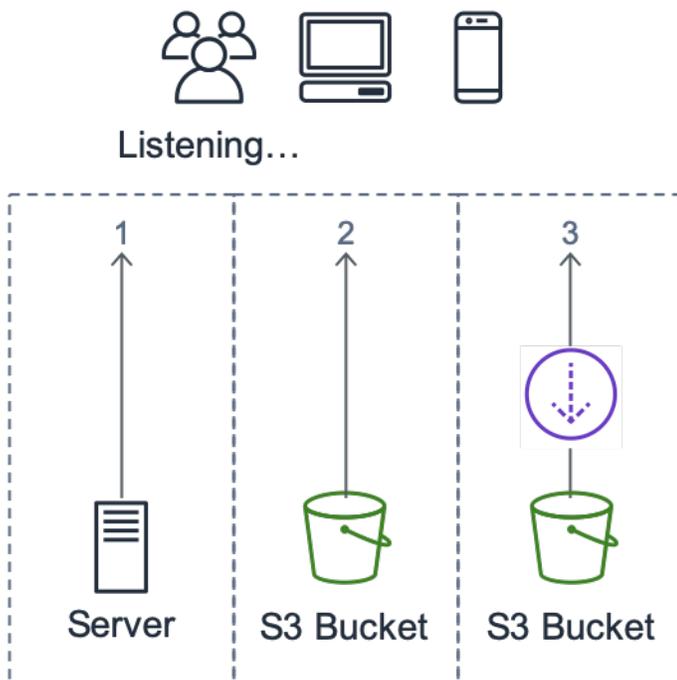
Terjemahkan dan vokalisasikan konten Anda

Plugin AWS for WordPress memungkinkan Anda menerjemahkan teks secara otomatis dalam berbagai bahasa, dan mengubah konten tertulis menjadi format audio multibahasa. Fitur-fitur ini didukung oleh layanan Amazon Machine Learning.

[Amazon Polly](#) adalah layanan yang mengubah teks menjadi ucapan seperti asli. Dengan banyak suara yang dalam berbagai bahasa, Anda dapat memilih suara yang ideal dan membangun aplikasi dengan dukungan ucapan yang digunakan di banyak negara yang berbeda. Gunakan plugin ini untuk membuat file audio dalam salah satu suara dan bahasa yang didukung oleh Amazon Polly. Pengunjung Anda dapat melakukan pengaliran audio sesuai keinginan mereka menggunakan pemutar audio bawaan dan aplikasi seluler.

Secara default, plugin ini menyimpan file audio baru di server web Anda. Anda dapat memilih untuk menyimpan file ini di Amazon S3 atau di Amazon CloudFront. Pengguna memiliki pengalaman mendengarkan yang sama terlepas di mana Anda menyimpan file audio Anda. Hanya lokasi siaran yang berubah:

- Untuk file audio yang tersimpan di server WordPress, file disiarkan langsung dari server.
- Untuk file yang disimpan dalam bucket S3, file disiarkan dari bucket ini.
- Jika Anda menggunakan CloudFront, file disimpan di Amazon S3 dan disiarkan dengan CloudFront.

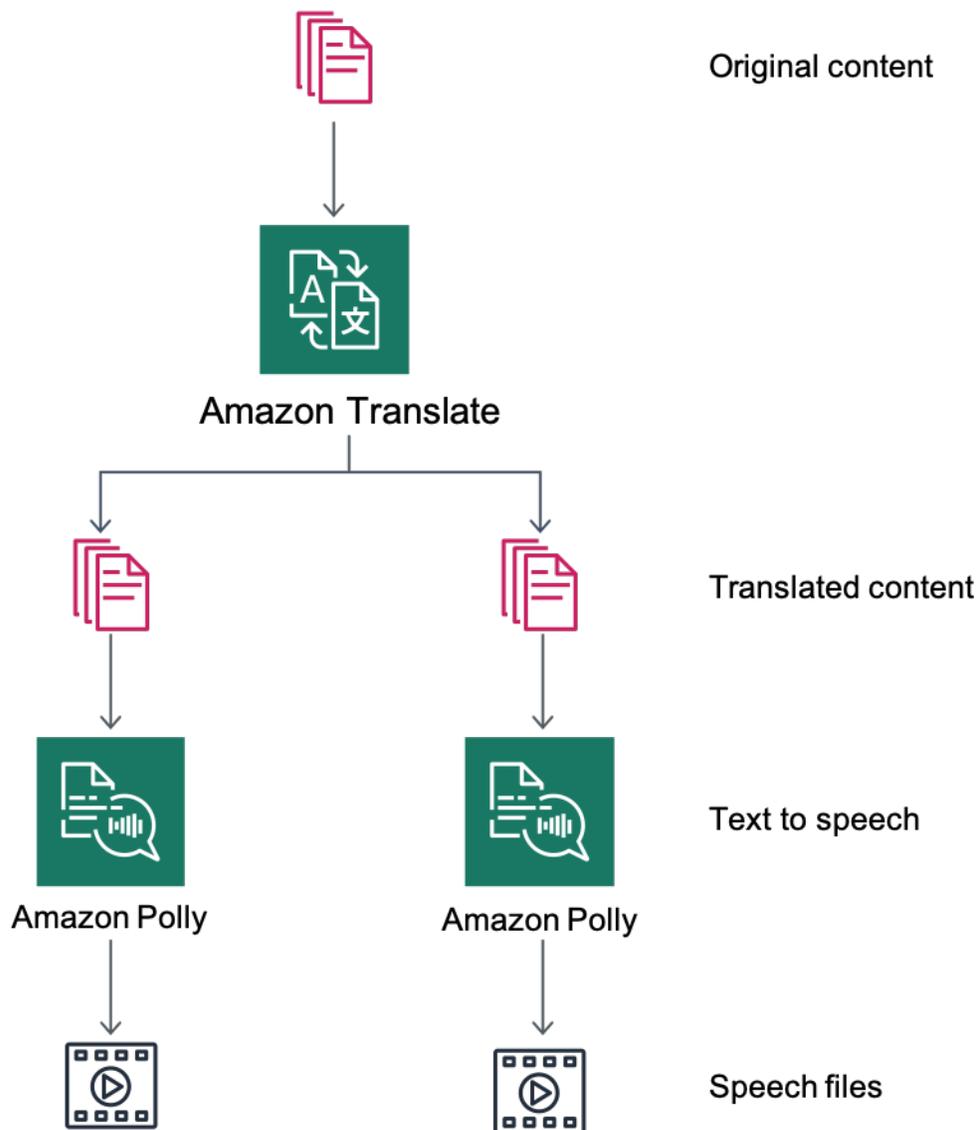


Lokasi siaran

[Amazon Translate](#) adalah layanan terjemahan mesin neural yang memberikan terjemahan bahasa yang cepat, berkualitas tinggi, dan terjangkau. Menyediakan konten multibahasa merupakan peluang besar bagi pemilik situs. Meskipun bahasa Inggris adalah bahasa dominan di web, penutur asli bahasa Inggris hanya berjumlah 26% dari total audiens online.

Dengan menawarkan versi tertulis dan audio dari konten WordPress Anda dalam berbagai bahasa, Anda dapat memenuhi kebutuhan audiens internasional yang lebih besar. Anda dapat mengonfigurasi plugin untuk melakukan hal berikut:

- Secara otomatis menerjemahkan ke dalam bahasa yang berbeda-beda dan membuat rekaman audio dari setiap terjemahan untuk konten baru pada saat publikasi, atau memilih untuk menerjemahkan dan membuat rekaman untuk postingan individual
- Menerjemahkan ke dalam bahasa yang berbeda-beda dan membuat rekaman audio untuk setiap terjemahan konten yang diarsipkan
- Menggunakan umpan RSS Amazon Pollycast untuk konten audio podcast



Gambaran umum terjemahan konten dan fungsi teks-ke-ucapan

Penyediaan podcast dengan Amazon Pollycast

Dengan umpan Amazon Pollycast, pengunjung Anda dapat mendengarkan konten audio Anda menggunakan aplikasi podcast standar. Umpan Pollycast yang memenuhi standar RSS 2.0 menyediakan data XML yang diperlukan untuk mengagregasi podcast oleh aplikasi podcast seluler populer, seperti iTunes, dan direktori podcast.

Saat Anda menginstal plugin AWS for WordPress, Anda akan menemukan opsi untuk mengaktifkan pembuatan umpan XML di tab konfigurasi Podcast. Di sana, Anda juga akan menemukan opsi untuk mengonfigurasi sejumlah properti opsional. Setelah mengaktifkan fungsionalitas ini, Anda akan menerima tautan feed.

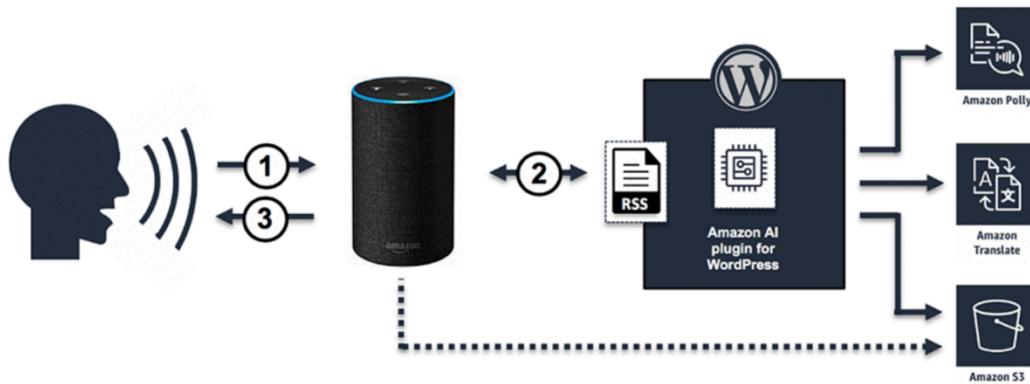
Membacakan konten Anda melalui Perangkat Amazon Alexa

Anda dapat memperluas situs web dan blog WordPress melalui perangkat Alexa. Hal ini membuka kemungkinan baru bagi kreator dan penulis situs web untuk menjangkau audiens yang lebih luas. Hal ini juga mempermudah orang untuk mendengarkan blog favorit mereka cukup dengan meminta Alexa untuk membacakannya.

Untuk mengekspos situs web WordPress ke Alexa, Anda harus mengaktifkan:

- Plugin AWS for WordPress.
- Fungsionalitas teks-ke-ucapan dan Amazon Pollycast. Fungsionalitas ini menghasilkan umpan RSS di situs WordPress Anda yang dikonsumsi oleh Amazon Alexa.
- Amazon S3 adalah penyimpanan default untuk file Anda dalam teks-ke-ucapan, sehingga penting bahwa situs web Anda menggunakan koneksi HTTPS aman untuk mengekspos umpannya ke Alexa.

Diagram berikut menyajikan alur interaksi dan komponen yang diperlukan untuk mengekspos situs web Anda melalui Alexa.



Alur interaksi yang diperlukan untuk mengekspos situs web WordPress melalui Alexa

1. Pengguna memanggil keterampilan Alexa baru, misalnya dengan mengatakan: Alexa, minta info terbaru Demo Blog. Keterampilan itu sendiri dibuat menggunakan salah satu Alexa Skill Blueprints. Hal ini memungkinkan Anda mengekspos keterampilan Anda melalui perangkat Alexa bahkan jika Anda tidak memiliki pengetahuan teknis yang mendalam.
2. Skill Alexa akan menganalisis panggilan dan umpan RSS yang dihasilkan oleh plugin AWS for WordPress, lalu mengembalikan tautan ke versi audio artikel terbaru.
3. Berdasarkan tautan yang disediakan oleh umpan ini, Alexa akan membacakan artikel dengan memutar file audio yang disimpan di Amazon S3.

Lihat [halaman plugin](#) di marketplace WordPress untuk panduan langkah demi langkah mendetail dalam menginstal dan mengonfigurasi plugin dan fungsinya.

Konfigurasi konten statis

Secara default, WordPress menyimpan semuanya secara lokal di server web, yaitu penyimpanan blok (Amazon EBS) untuk [deployment server tunggal](#), dan penyimpanan file (Amazon EFS) untuk [deployment elastis](#). Selain mengurangi biaya penyimpanan dan transfer data, memindahkan aset statis ke Amazon S3 menawarkan skalabilitas, ketersediaan data, keamanan, dan performa.

Dalam contoh ini, plugin W3 Total Cache (W3TC) digunakan untuk menyimpan aset statis di Amazon S3. Namun, ada plugin lain yang tersedia dengan kemampuan serupa. Jika Anda ingin menggunakan alternatif, Anda dapat menyesuaikan langkah-langkah berikut. Langkah-langkah ini hanya mengacu pada fitur atau pengaturan yang relevan dengan contoh ini. Penjelasan mendetail tentang semua pengaturan berada di luar cakupan dokumen ini. Lihat [halaman plugin W3 Total Cache](#) di [wordpress.org](#) untuk informasi lebih lanjut.

Pembuatan pengguna IAM

Anda perlu membuat pengguna AWS Identity and Access Management (IAM) untuk plugin WordPress untuk menyimpan aset statis di Amazon S3. Untuk langkahnya, lihat [Membuat Pengguna IAM di Akun AWS Anda](#).

Catatan: IAM role menyediakan cara yang lebih baik untuk mengelola akses ke sumber daya AWS, tetapi pada saat penulisan dokumen ini, plugin W3 Total Cache tidak mendukung [IAM role](#).

Catat kredensial keamanan pengguna dan simpan dengan cara yang aman – Anda memerlukan kredensial ini nanti.

Pembuatan bucket Amazon S3

1. Pertama, buat bucket Amazon S3 di Wilayah AWS pilihan Anda. Untuk langkahnya, lihat [Membuat bucket](#). Aktifkan hosting situs web statis untuk bucket tersebut dengan mengikuti [Tutorial: Mengonfigurasi situs web statis di Amazon S3](#).
2. Buat kebijakan IAM untuk memberikan akses untuk pengguna IAM yang dibuat sebelumnya ke bucket S3 yang ditentukan, dan lampirkan kebijakan ke pengguna IAM ini. Untuk langkah-langkah membuat kebijakan berikut, lihat [Mengelola Kebijakan IAM](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1389783689000",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::wp-demo",
        "arn:aws:s3:::wp-demo/*"
      ]
    }
  ]
}
```

```
]
}
```

3. Instal dan aktifkan plugin W3TC dari panel admin WordPress.
4. Telusuri ke bagian General Settings (Pengaturan Umum) di konfigurasi plugin ini, dan pastikan bahwa Browser Cache (Cache Browser) dan CDN diaktifkan.
5. Dari daftar drop-down dalam konfigurasi CDN, pilih Origin Push: Amazon CloudFront (Push Asal: Amazon CloudFront) (opsi ini memiliki Amazon S3 sebagai asalnya).
6. Telusuri ke bagian Browser Cache (Cache Browser) di konfigurasi plugin ini dan aktifkan header expires (kedaluwarsa), cache control (kontrol cache), dan entity tag (ETag) (tag entitas (ETag)).
7. Juga aktifkan opsi Mencegah caching objek setelah pengaturan perubahan sehingga string kueri baru dihasilkan dan ditambahkan ke objek setiap kali pengaturan diubah.
8. Telusuri ke bagian CDN pada konfigurasi plugin ini dan masukkan kredensial keamanan pengguna IAM yang Anda buat sebelumnya, serta nama bucket S3.
9. Jika Anda menyajikan situs web Anda melalui URL CloudFront, masukkan nama domain distribusi di kotak yang relevan. Jika tidak, masukkan satu atau beberapa CNAME untuk nama domain kustom Anda.
10. Terakhir, ekspor pustaka media dan unggah wp-includes, file tema, dan file kustom ke Amazon S3 menggunakan plugin W3TC. Fungsi unggah ini tersedia di bagian General (Umum) di halaman konfigurasi CDN.

Pembuatan asal statis

Sekarang file statis disimpan di Amazon S3, kembali ke konfigurasi CloudFront di konsol CloudFront, dan konfigurasikan Amazon S3 sebagai asal konten statis. Untuk melakukannya, tambahkan asal kedua yang menunjuk ke bucket S3 yang Anda buat untuk tujuan tersebut. Kemudian buat dua perilaku cache lagi, satu untuk masing-masing dari dua folder (wp-content dan wp-includes) yang harus menggunakan asal S3 daripada asal default untuk konten dinamis. Konfigurasikan keduanya dengan cara yang sama:

- Sajikan permintaan GET HTTP saja.
- Amazon S3 tidak memvariasikan output-nya berdasarkan cookie atau header HTTP, sehingga Anda dapat meningkatkan efisiensi caching dengan tidak meneruskannya ke asal melalui CloudFront.

- Terlepas dari fakta bahwa perilaku ini hanya menyajikan konten statis (yang tidak menerima parameter), Anda akan meneruskan string kueri ke asal. Hal ini dimaksudkan agar Anda dapat menggunakan string kueri sebagai pengidentifikasi versi untuk langsung menginvalidasi, misalnya, file CSS lama saat men-deploy versi baru. Untuk informasi selengkapnya, lihat [Panduan Developer Amazon CloudFront](#).

 Note

Setelah menambahkan perilaku asal statis ke distribusi CloudFront Anda, periksa urutannya untuk memastikan perilaku untuk `wp-admin/*` dan `wp-login.php` memiliki prioritas yang lebih tinggi daripada perilaku untuk konten statis. Jika tidak, Anda mungkin melihat perilaku aneh saat mengakses panel admin Anda.

Lampiran C: Pencadangan dan pemulihan

Memulihkan dari kegagalan di AWS akan lebih cepat dan lebih mudah dilakukan dibandingkan dengan lingkungan hosting tradisional. Misalnya, Anda dapat meluncurkan instans pengganti dalam hitungan menit sebagai respons terhadap kegagalan perangkat keras, atau Anda dapat menggunakan failover otomatis di banyak layanan terkelola kami untuk meniadakan dampak reboot karena pemeliharaan rutin.

Namun, Anda masih perlu memastikan Anda mencadangkan data yang tepat agar berhasil memulihkannya. Untuk membangun kembali ketersediaan situs web WordPress, Anda harus dapat memulihkan komponen-komponen berikut:

- Peningkatan dan konfigurasi sistem operasi (OS) dan layanan (Apache, MySQL, dan sebagainya)
- Kode dan konfigurasi aplikasi WordPress
- Tema dan plugin WordPress
- Unggahan (misalnya, file media untuk postingan)
- Konten basis data (postingan, komentar, dan sebagainya)

AWS menyediakan berbagai metode untuk mencadangkan dan memulihkan data dan aset aplikasi web Anda.

Laporan resmi ini sebelumnya membahas penggunaan snapshot Lightsail untuk melindungi semua data yang tersimpan di penyimpanan lokal instans. Jika situs web WordPress Anda hanya menjalankan instans Lightsail, snapshot Lightsail biasa seharusnya akan cukup bagi Anda untuk memulihkan situs web WordPress Anda secara keseluruhan. Namun, Anda masih akan kehilangan perubahan apa pun yang diterapkan ke situs web Anda sejak snapshot terakhir diambil jika Anda melakukan pemulihan dari snapshot.

Dalam deployment multi-server, Anda perlu mencadangkan masing-masing komponen yang dibahas sebelumnya menggunakan mekanisme yang berbeda-beda. Setiap komponen mungkin memiliki persyaratan yang berbeda-beda untuk frekuensi pencadangan, misalnya, peningkatan dan konfigurasi OS dan WordPress akan berubah jauh lebih jarang daripada konten yang dibuat pengguna, sehingga dapat dicadangkan lebih jarang tanpa kehilangan data jika terjadi pemulihan.

Untuk mencadangkan peningkatan dan konfigurasi OS dan layanan, serta kode dan konfigurasi aplikasi WordPress, Anda dapat membuat AMI instans EC2 yang dikonfigurasi dengan benar.

AMI dapat memenuhi dua tujuan: untuk berfungsi sebagai cadangan status instans, dan berfungsi sebagai templat saat meluncurkan instans baru.

Untuk mencadangkan kode dan konfigurasi aplikasi WordPress, Anda perlu menggunakan AMI dan juga cadangan Aurora.

Untuk mencadangkan tema dan plugin WordPress yang diinstal di situs web Anda, cadangkan bucket Amazon S3 atau sistem file Amazon EFS yang menyimpannya.

- Untuk tema dan plugin yang disimpan dalam bucket S3, Anda dapat mengaktifkan [replikasi Lintas Wilayah](#) sehingga semua objek yang diunggah ke bucket utama Anda direplikasi secara otomatis ke bucket cadangan Anda di Wilayah AWS lain. Replikasi Lintas Wilayah mengharuskan [versioning](#) diaktifkan pada bucket sumber dan tujuan Anda, yang memberi Anda lapisan perlindungan tambahan dan memungkinkan Anda kembali ke versi sebelumnya dari objek tertentu dalam bucket Anda.
- Untuk tema dan plugin yang disimpan pada sistem file EFS, Anda dapat membuat AWS Data Pipeline untuk menyalin data dari sistem file EFS produksi Anda ke sistem file EFS lain, seperti yang diuraikan dalam halaman dokumentasi [Mencadangkan sistem file Amazon EFS Anda](#). Anda juga dapat mencadangkan sistem file EFS menggunakan aplikasi pencadangan yang sudah Anda kenal.
- Untuk mencadangkan unggahan pengguna, Anda harus mengikuti langkah-langkah yang diuraikan sebelumnya untuk mencadangkan tema dan plugin WordPress.
- Untuk mencadangkan konten basis data, Anda perlu menggunakan [cadangan Aurora](#). Aurora mencadangkan volume klaster Anda secara otomatis dan menyimpan data pemulihan selama periode penyimpanan cadangan. Cadangan Aurora bersifat kontinu dan inkremental sehingga Anda dapat dengan cepat memulihkan ke titik mana pun dalam periode retensi cadangan. Tidak ada dampak performa atau gangguan layanan basis data yang terjadi saat data cadangan sedang ditulis. Anda dapat menentukan periode retensi cadangan dari 1 hingga 35 hari. Anda juga dapat membuat [snapshot basis data manual](#), yang dipersistensi sampai Anda menghapusnya. Snapshot basis data manual berguna untuk pencadangan dan pengarsipan jangka panjang.

Lampiran D: Men-deploy plugin dan tema baru

Beberapa situs web tetap statis. Dalam kebanyakan kasus, Anda secara berkala akan menambahkan tema dan plugin WordPress yang tersedia untuk umum atau meng-upgrade ke versi WordPress yang lebih baru. Dalam kasus lain, Anda akan mengembangkan tema dan plugin kustom Anda sendiri dari awal.

Setiap kali Anda membuat perubahan struktural pada penginstalan WordPress Anda, ada risiko tertentu untuk menimbulkan masalah yang tak terduga. Paling tidak, ambil cadangan kode, konfigurasi, dan basis data aplikasi Anda sebelum menerapkan perubahan signifikan (seperti menginstal plugin baru). Untuk situs web bisnis atau nilai lainnya, uji perubahan tersebut di lingkungan penahanan terpisah terlebih dahulu. Dengan AWS, mudah untuk mereplikasi konfigurasi lingkungan produksi Anda dan menjalankan seluruh proses deployment dengan cara yang aman. Setelah selesai dengan pengujian Anda, Anda cukup menghapus lingkungan pengujian Anda dan berhenti membayar sumber daya tersebut. Selanjutnya, laporan resmi ini membahas beberapa pertimbangan khusus WordPress.

Beberapa plugin menulis informasi konfigurasi ke tabel basis data `wp_options` (atau membuat perubahan skema basis data), sedangkan yang lain membuat file konfigurasi di direktori penginstalan WordPress. Karena kami telah memindahkan basis data dan penyimpanan ke platform bersama, perubahan ini segera tersedia untuk semua instans yang berjalan tanpa upaya lebih lanjut dari pihak Anda.

Saat men-deploy tema baru di WordPress, mungkin memerlukan sedikit upaya. Jika Anda hanya menggunakan Amazon EFS untuk menyimpan semua file penginstalan WordPress Anda, maka tema baru akan segera tersedia untuk semua instans yang berjalan. Namun, jika Anda melepas konten statis ke Amazon S3, Anda harus memproses salinannya ke lokasi bucket yang tepat. Plugin seperti W3 Total Cache menyediakan cara bagi Anda untuk secara manual menginisiasikan tugas ini. Atau, Anda dapat mengotomatisasi langkah ini sebagai bagian dari proses pembangunan.

Karena aset tema dapat di-cache di CloudFront dan di browser, Anda memerlukan cara untuk menginvalidasi versi lama saat men-deploy perubahan. Cara terbaik untuk mencapai ini adalah dengan menyertakan semacam pengidentifikasi versi di objek Anda. Pengidentifikasi ini dapat berupa string kueri dengan stempel tanggal-waktu atau string acak. Jika Anda menggunakan plugin W3 Total Cache, Anda dapat memperbarui string kueri media yang ditambahkan ke URL file media.

Pemberitahuan

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

© 2021 Amazon Web Services, Inc. atau afiliasinya. Semua hak dilindungi undang-undang.