

Cara Menyediakan Sumber Daya AWS

Sampai di sini, kita telah membicarakan tentang beberapa sumber daya serta infrastruktur global AWS. Melihat materi-materi sebelumnya, muncullah sebuah pertanyaan, bagaimana sebenarnya kita dapat berinteraksi dengan layanan-layanan tersebut?

Jawabannya adalah API--sudah kita singgung sedikit di modul sebelumnya. Di AWS semua aktivitas adalah panggilan API. API adalah application programming interface atau antarmuka pemrograman aplikasi. Dengan kata lain, ada cara yang telah ditentukan sebelumnya untuk Anda sehingga dapat berinteraksi dengan layanan AWS.

Anda dapat memanggil API ini untuk menyediakan, mengonfigurasi, dan mengelola sumber daya AWS. Misal untuk meluncurkan EC2 instance atau membuat AWS Lambda *function*. Masing-masing akan menjadi permintaan dan panggilan API yang berbeda ke AWS.

Jadi, untuk berinteraksi dengan layanan AWS, Anda dapat menggunakan AWS Management Console, AWS Command Line Interface (CLI), dan AWS Software Development Kit (SDK). Atau berbagai alat lain seperti AWS Elastic Beanstalk dan AWS CloudFormation (layanan yang dapat membuat permintaan untuk dikirim ke AWS API guna membuat dan mengelola sumber daya AWS).

AWS Management Console



AWS Management Console adalah antarmuka berbasis browser yang dapat digunakan untuk mengakses dan mengelola layanan AWS. Melalui *console* (konsol), Anda dapat mengelola sumber daya AWS secara visual dan dengan cara yang mudah dipahami. Tentu ini adalah cara yang ideal untuk memulai dan membangun pengetahuan Anda tentang layanan AWS.

Dengan AWS Management Console, Anda dapat

- mencari layanan AWS dari nama, kata kunci, atau akronim;
- membangun lingkungan pengujian;
- melihat tagihan AWS;
- melakukan pemantauan; atau

- bekerja dengan sumber daya nonteknis lainnya.

AWS Console versi aplikasi seluler juga tersedia dan dapat Anda gunakan untuk melakukan tugas seperti memantau sumber daya, melihat alarm, dan mengakses informasi penagihan.

AWS Management Console adalah tempat pertama terbaik yang perlu Anda tuju ketika ingin mempelajari tentang AWS.

AWS Command Line Interface



Jika Anda akan menjalankan sumber daya di lingkungan produksi, tentu Anda tak ingin bergantung dengan cara *point and click* (tunjuk dan klik) yang diberikan console untuk membuat dan mengelola sumber daya AWS.

Ambil contoh pembuatan Amazon EC2 instance. Dengan AWS Management Console, Anda perlu klik berkali-kali, mengatur semua konfigurasi, barulah instance Anda dapat diluncurkan. Jika nantinya ingin meluncurkan EC2 instance lain, Anda harus kembali ke console dan melakukan proses klik yang sama terhadap layar-layar itu lagi. Ini akan membuang banyak waktu.

Jika seorang manusia terus-menerus melakukan penyediaan manual semacam ini, artinya Anda membuka peluang terhadap potensi kesalahan karena manusia sangat mudah untuk lupa mengeklik kotak centang atau salah mengeja sesuatu.

Jawaban untuk masalah ini adalah dengan menggunakan sarana yang memungkinkan Anda membuat skrip atau memprogram panggilan API, yaitu dengan AWS Command Line Interface atau CLI.

CLI memungkinkan Anda untuk mengendalikan layanan AWS dengan baris perintah melalui satu alat. Jelas ini berbeda dengan gaya navigasi visual dari AWS Management Console. AWS CLI tersedia untuk pengguna Windows, macOS, dan Linux.

Dengan menulis perintah menggunakan CLI, Anda dapat membuat tindakan yang dapat ditulis berkali-kali. Misal Anda menulis dan menjalankan perintah untuk meluncurkan EC2 instance. Nah, jika Anda ingin meluncurkan instance lain, cukup jalankan kembali perintah tersebut. Dengan demikian, Anda akan terhindar dari *human error* alias kesalahan manusia.

Selain itu, Anda juga dapat menjalankan skrip tersebut secara otomatis, contohnya dengan berdasarkan jadwal atau dipicu oleh proses lain.

AWS Software Development Kit



Cara lain untuk berkomunikasi dengan AWS adalah melalui AWS Software Development Kit atau SDK. SDK memudahkan Anda untuk berinteraksi dengan sumber daya AWS melalui berbagai bahasa pemrograman.

Hal ini memudahkan *developer* (pengembang) untuk membuat program di AWS *tanpa* menggunakan *low-level* API serta menghindari pembuatan sumber daya secara manual yang dari tadi kita bincangkan.

Sederhananya, *low-level* API memungkinkan Anda untuk memanipulasi fungsi di dalam API secara terperinci sesuai dengan kebutuhan. Lawannya adalah *high-level* API, yang memberikan lebih banyak fungsi dalam satu perintah dan lebih mudah digunakan sehingga Anda tak perlu mendalami detail teknis dan struktur API-nya.

Untuk membantu Anda memulai menggunakan SDK, AWS menyediakan dokumentasi dan sampel kode untuk setiap bahasa pemrograman yang didukung, yaitu mencakup C++, Go, Java, JavaScript, .NET, Node.js, PHP, Python, dan Ruby.

AWS Elastic Beanstalk

Opsi-opsi sebelumnya, yaitu AWS Management Console, CLI, dan SDK adalah cara penyediaan dan pengelolaan lingkungan AWS yang harus *Anda lakukan sendiri*.

Kita telah belajar, bahwa jika ingin menyediakan sumber daya, kita dapat melakukan salah satu cara di bawah ini:

- Masuk ke konsol, arahkan dan klik.
- Menulis beberapa perintah.
- Menulis beberapa program untuk melakukannya.

Nah, selain yang disebutkan di atas, ada juga cara lain untuk mengelola lingkungan AWS Anda menggunakan *managed service* (layanan terkelola) seperti AWS Elastic Beanstalk.

AWS Elastic Beanstalk adalah layanan yang dapat membantu Anda menyediakan lingkungan berbasis Amazon EC2. Tak perlu lagi harus klik sana sini di console atau menulis beberapa baris perintah untuk membangun jaringan, EC2 instance, *scaling* (penyesuaian kapasitas), dan Elastic Load Balancer.

Ucapkan selamat tinggal kepada semua itu. AWS Elastic Beanstalk dapat menyediakan dan mengelola semua infrastruktur tersebut sembari tetap memberi visibilitas dan kendali atas sumber daya yang mendasarinya. Dengan begitu, Anda bisa fokus pada aplikasi bisnis bukan infrastrukturnya.

Cukup unggah kode dan tentukan konfigurasi yang Anda inginkan, maka AWS Elastic Beanstalk pun akan mengolah informasi tersebut dan membangun lingkungan AWS-nya untuk Anda.

Lingkungan yang dimaksud adalah:

- Penyesuaian Kapasitas
- *Load balancing* (Penyeimbang beban)
- *Auto-scaling* (Penskalaan otomatis)
- Pemantauan kesehatan aplikasi

Simpan konfigurasi lingkungan tersebut sehingga nantinya akan memudahkan Anda saat menerapkannya kembali.

AWS CloudFormation

Layanan lain yang dapat Anda gunakan untuk membantu membuat penerapan (*deployment*) otomatis dan berulang adalah AWS CloudFormation.

AWS CloudFormation adalah layanan *infrastructure as code* (infrastruktur sebagai kode) yang memungkinkan Anda untuk menentukan berbagai sumber daya AWS dengan cara deklaratif menggunakan dokumen berbasis teks JSON atau YAML yang disebut sebagai CloudFormation template.

Format deklaratif memudahkan Anda untuk memberikan spesifikasi apa yang ingin dibangun tanpa harus mendeskripsikan detail bagaimana caranya karena mesin CloudFormation yang akan mengurusnya. Anda hanya perlu menulis baris kode untuk membangun lingkungan ketimbang menggunakan AWS Management Console.

AWS CloudFormation tak hanya terbatas pada solusi berbasis EC2, melainkan mendukung banyak sumber daya AWS, seperti penyimpanan, database, analitik, *machine learning*, dan banyak lagi.

Oke, setelah Anda menentukan sumber daya dalam CloudFormation template, AWS CloudFormation akan menguraikannya dan mulai menyediakan semua sumber daya tersebut secara paralel.

AWS CloudFormation mengelola semua panggilan API. Anda dapat menjalankan CloudFormation template yang sama di beberapa akun atau Region. Dengan begitu, layanan ini akan membuat lingkungan yang identik di dalamnya. Tak akan ada lagi *human error* karena semua proses berjalan secara otomatis dengan sepenuhnya.

[< Sebelumnya](#)

[Selanjutnya >](#)