Diagram Arsitektur

Komposisi API Paralel di AWS



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Komposisi API Paralel di AWS: Diagram Arsitektur

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

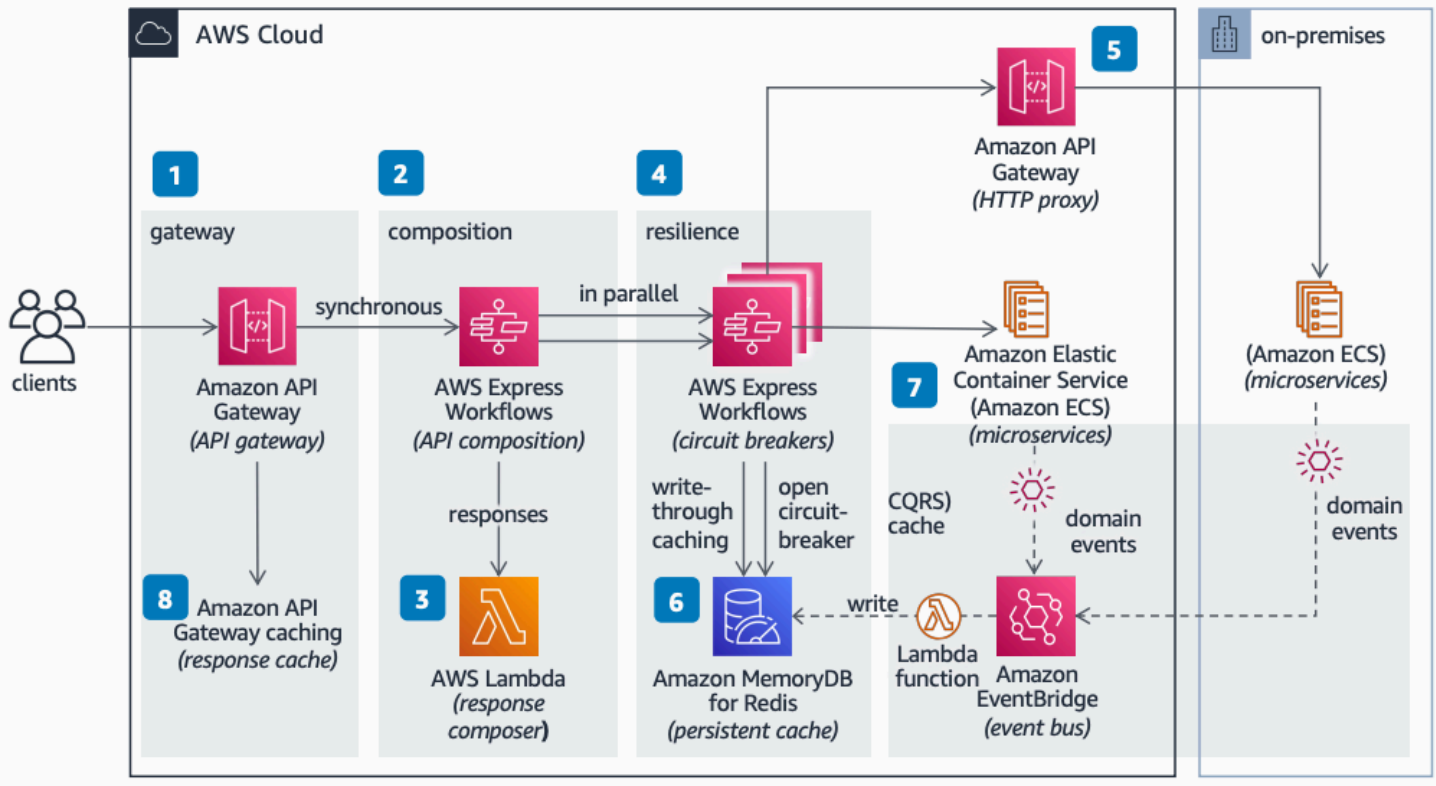
Komposisi API Paralel i
Komposisi API Paralel di AWS 1
Sumber bacaan lebih lanjut 2
Sejarah diagram 2
..... iv

Komposisi API Paralel di AWS

Tanggal publikasi: 17 Mei 2022 (Sejarah diagram)

Arsitektur ini menunjukkan cara memanggil beberapa titik akhir API hilir, secara paralel atau berurutan, untuk menyusun respons agregat tunggal. Anda membangun alur kerja pemutus sirkuit yang umum dan berparameter menggunakan Alur Kerja Ek AWS Step Functions spres dan menggunakan pemisahan tanggung jawab kueri perintah (CQRS) untuk mempertahankan cache yang konsisten.

Komposisi API Paralel di AWS



Langkah-langkah berikut menjelaskan arsitektur:

1. Buat gateway API menggunakan Amazon API Gateway untuk memungkinkan klien mengirim permintaan web sinkron ke layanan mikro Anda.
2. Gunakan Step Functions Express Workflow untuk membuat alur kerja dengan langkah-langkah paralel yang memanggil beberapa layanan mikro secara bersamaan.
3. Gunakan AWS Lambda fungsi untuk menyusun beberapa respons dari layanan mikro hilir menjadi satu respons agregat untuk klien.

4. Tangani permintaan paralel dengan pemutus sirkuit yang dibangun dengan alur kerja Step Functions Express bersarang berparameter untuk meningkatkan ketahanan.
5. Gunakan API Gateway untuk memproksi permintaan HTTP saat memanggil layanan mikro lokal.
6. Buat database Amazon MemoryDB for Redis untuk menyimpan respons microservice menggunakan pola caching write-through. https://docs.aws.amazon.com/whitepapers/latest/database-caching-strategies-using-redis/caching-patterns.html Ketika layanan mikro hilir menjadi tidak tersedia atau latensinya mencapai ambang batas, sirkuit menutup dan membaca semua respons langsung dari cache.
7. Lengkapi data cache dengan peristiwa domain dari layanan mikro hilir menggunakan pola CQRS. Buat bus acara dengan Amazon EventBridge, lalu tangani peristiwa dengan fungsi Lambda untuk mempertahankan agregat domain di MemoryDB.
8. Aktifkan cache respons API Gateway dan sesuaikan nilai dan filter time-to-live (TTL) sesuai dengan setiap jenis permintaan untuk meningkatkan kinerja.

Sumber bacaan lebih lanjut

Untuk informasi tambahan, lihat sumber daya berikut:

- AWS Ikon Arsitektur
- AWS Pusat Arsitektur
- AWS Well-Architected

Sejarah diagram

Untuk diberitahu tentang pembaruan diagram arsitektur referensi ini, berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Publikasi awal	Diagram arsitektur referensi pertama kali diterbitkan.	Mei 17, 2022

> **Note**
>
> Untuk berlangganan pembaruan RSS, Anda harus mengaktifkan plugin RSS untuk browser yang Anda gunakan.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.