



Diagramas de arquitetura

Composição de API paralela na AWS



Composição de API paralela na AWS: Diagramas de arquitetura

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

Composição de API paralela i

Composição de API paralela na AWS 1

Outras fontes de leitura 2

Histórico do diagrama 2

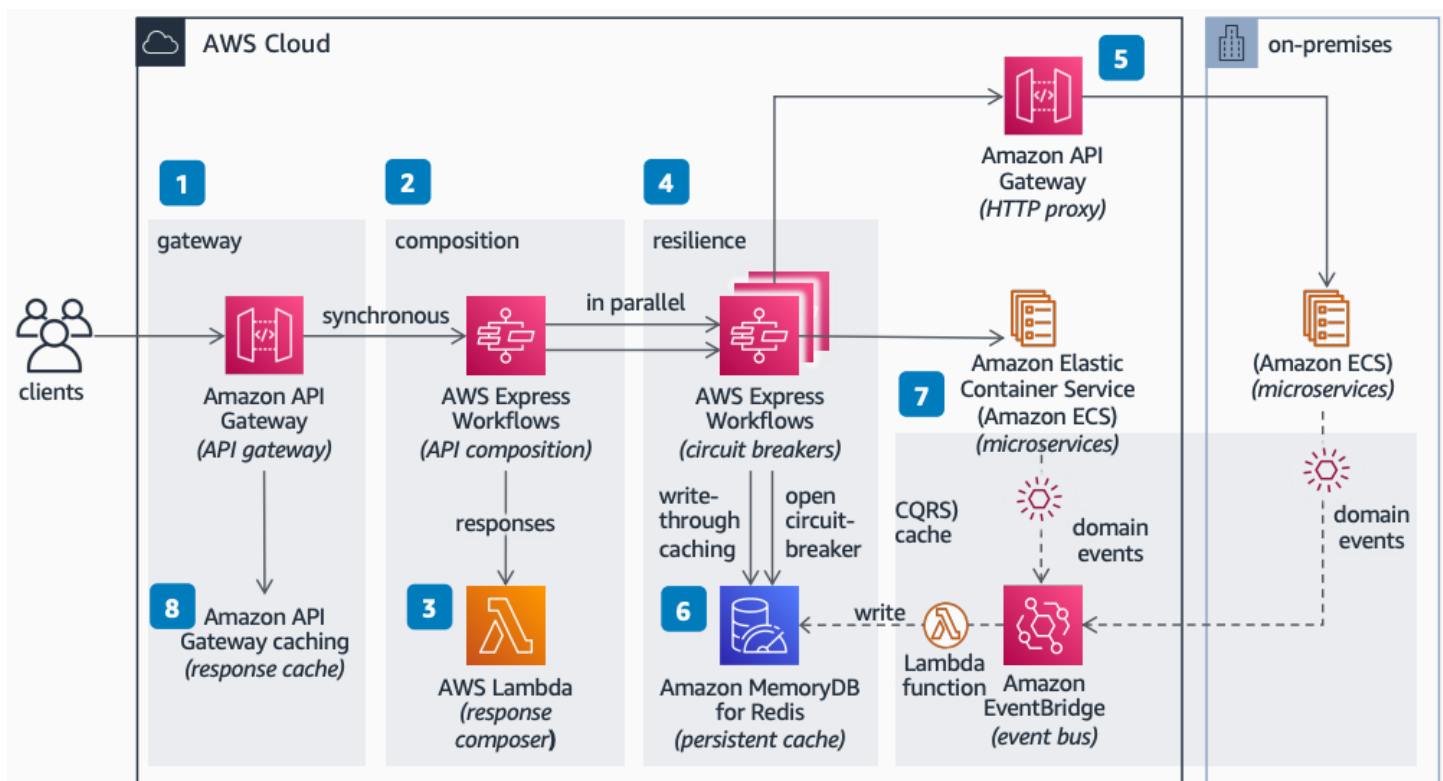
..... iv

Composição de API paralela na AWS

Data de publicação: 17 de maio de 2022 ([Histórico do diagrama](#))

Essa arquitetura mostra como chamar vários endpoints de API downstream, em paralelo ou em sequência, para compor uma única resposta agregada. Você cria um fluxo de trabalho genérico e parametrizado de disjuntores usando fluxos de trabalho [AWS Step Functions](#) expressos e usa a segregação de responsabilidade por consulta de comando (CQRS) para manter um cache persistente e eventualmente consistente.

Composição de API paralela na AWS



As etapas a seguir descrevem a arquitetura:

1. Crie um gateway de API usando o [Amazon API Gateway](#) para permitir que os clientes enviem solicitações síncronas da web para seus microserviços.
2. Use os fluxos de trabalho do Step Functions Express para criar um fluxo de trabalho com etapas paralelas que invocam vários microserviços ao mesmo tempo.
3. Use uma [AWS Lambda](#) função para compor as várias respostas dos microserviços downstream em uma única resposta agregada para clientes.

4. Gerencie solicitações paralelas com disjuntores criados com fluxos de trabalho Step Functions Express aninhados e parametrizados para aumentar a resiliência.
5. Use o API Gateway para intermediar solicitações HTTP ao invocar microsserviços locais.
6. Crie um banco de dados Amazon MemoryDB for Redis para armazenar em cache respostas de microsserviços usando o padrão de armazenamento em cache de gravação contínua. <https://docs.aws.amazon.com/whitepapers/latest/database-caching-strategies-using-redis/caching-patterns.html> Quando um microsserviço downstream fica indisponível ou sua latência atinge um limite, o circuito fecha e lê todas as respostas diretamente do cache.
7. Complemente os dados em cache com eventos de domínio de microsserviços downstream usando o padrão CQRS. Crie um barramento de eventos com a [Amazon e EventBridge](#), em seguida, gerencie eventos com uma função Lambda para manter os agregados de domínio no MemoryDB.
8. Ative o cache de resposta do API Gateway e ajuste os valores e filtros de tempo de vida (TTL) de acordo com cada tipo de solicitação para aumentar o desempenho.

Outras fontes de leitura

Para obter informações adicionais, consulte os seguintes recursos:

- [AWS Ícones de arquitetura](#)
- [AWS Centro de Arquitetura](#)
- [AWS Well-Architected](#)

Histórico do diagrama

Para ser notificado sobre atualizações nesse diagrama de arquitetura de referência, assine o feed RSS.

Alteração	Descrição	Data
Publicação inicial	Diagrama de arquitetura de referência publicado pela primeira vez.	17 de maio de 2022

 Note

Para assinar as atualizações de RSS, você deve ter um plug-in RSS ativado para o navegador que está usando.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.