



Architecture Diagrams

Multi-Region Custom-Domain AppSync Proxy API



Multi-Region Custom-Domain AppSync Proxy API: Architecture Diagrams

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Multi-Region AppSync Proxy **i**

Multi-Region Custom-Domain AppSync Proxy API 1

Further reading 2

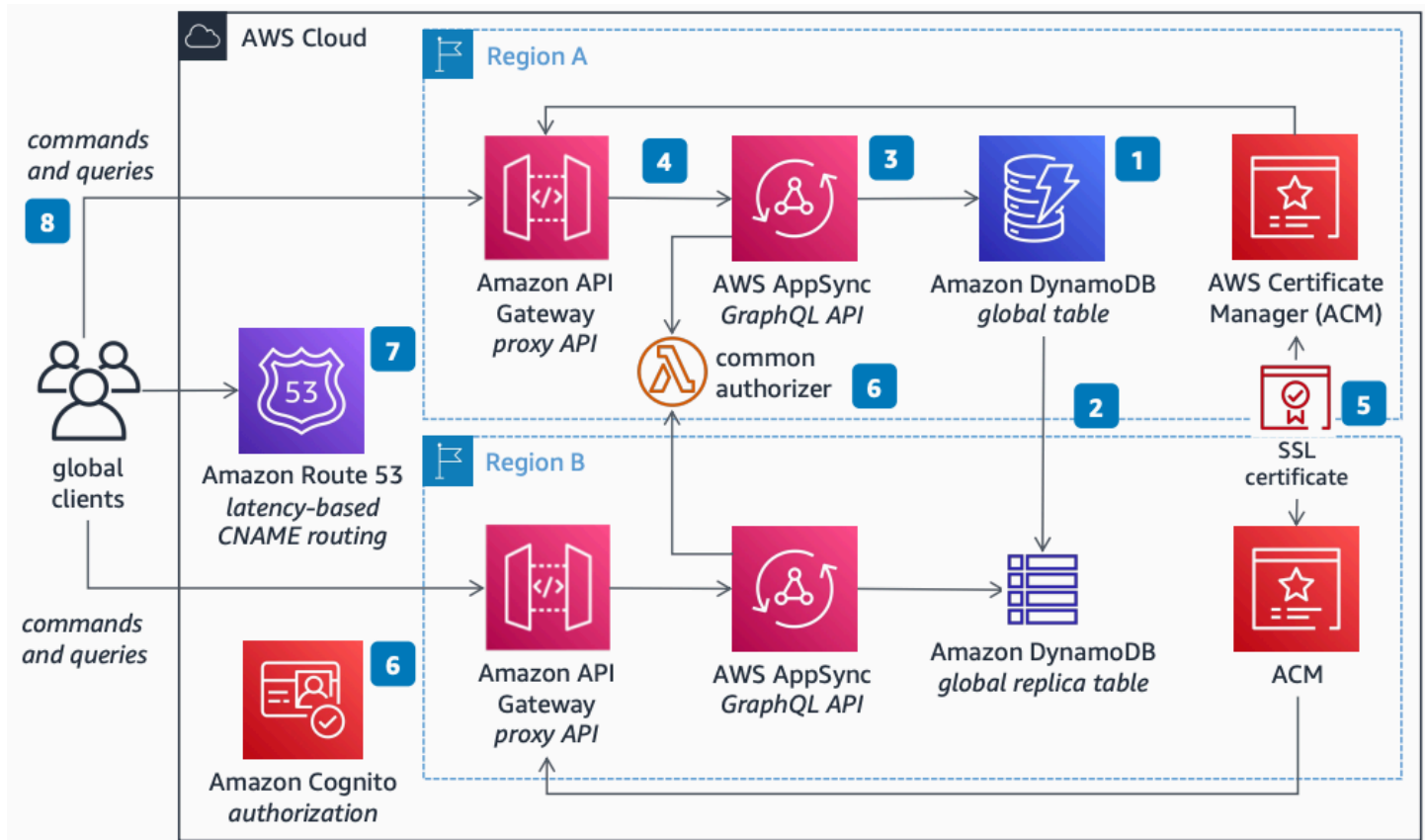
Diagram history 2

Multi-Region Custom-Domain AppSync Proxy API

Publication date: May 6, 2022 ([Diagram history](#))

This architecture shows how to reduce latency for end users while increasing availability by providing GraphQL API endpoints in multiple AWS Regions. You use [Amazon AppSync](#) with active-active real-time data synchronization supported by [Amazon DynamoDB](#) Global Tables.

Multi-Region Custom-Domain AppSync Proxy API



The following steps describe the architecture:

1. Create one or more DynamoDB database tables in the first AWS Region to store your workload's data. Enable the DynamoDB Global Table feature to propagate data changes bidirectionally in multi-active mode to all other Regions.
2. Create an Amazon AppSync GraphQL API endpoint in each Region in scope, using DynamoDB Global Tables to store and replicate your workload's data.

3. Create an [Amazon API Gateway](#) proxy API endpoint in each Region in scope, forwarding all requests to the same Regional AppSync endpoint.
4. Create or import an SSL certificate for your domain to the AWS Certificate Manager (ACM) of each Region. Configure each Regional API Gateway to support HTTPS requests by referencing your domain's SSL certificate.
5. Configure all Regional AppSync endpoints to use a common authorization mechanism using [Amazon Cognito](#) or a common [AWS Lambda](#) authorizer for seamless client authorization.
6. Configure [Amazon Route 53](#) to forward your domain's requests to the Regional API Gateway endpoint with the lowest latency to the client's location.
7. Clients across the world send requests for commands and queries to the nearest Region.
8. DynamoDB Global Tables automatically replicate data changes to all configured Regions.

Further reading

For additional information, refer to the following resources:

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	May 6, 2022

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.