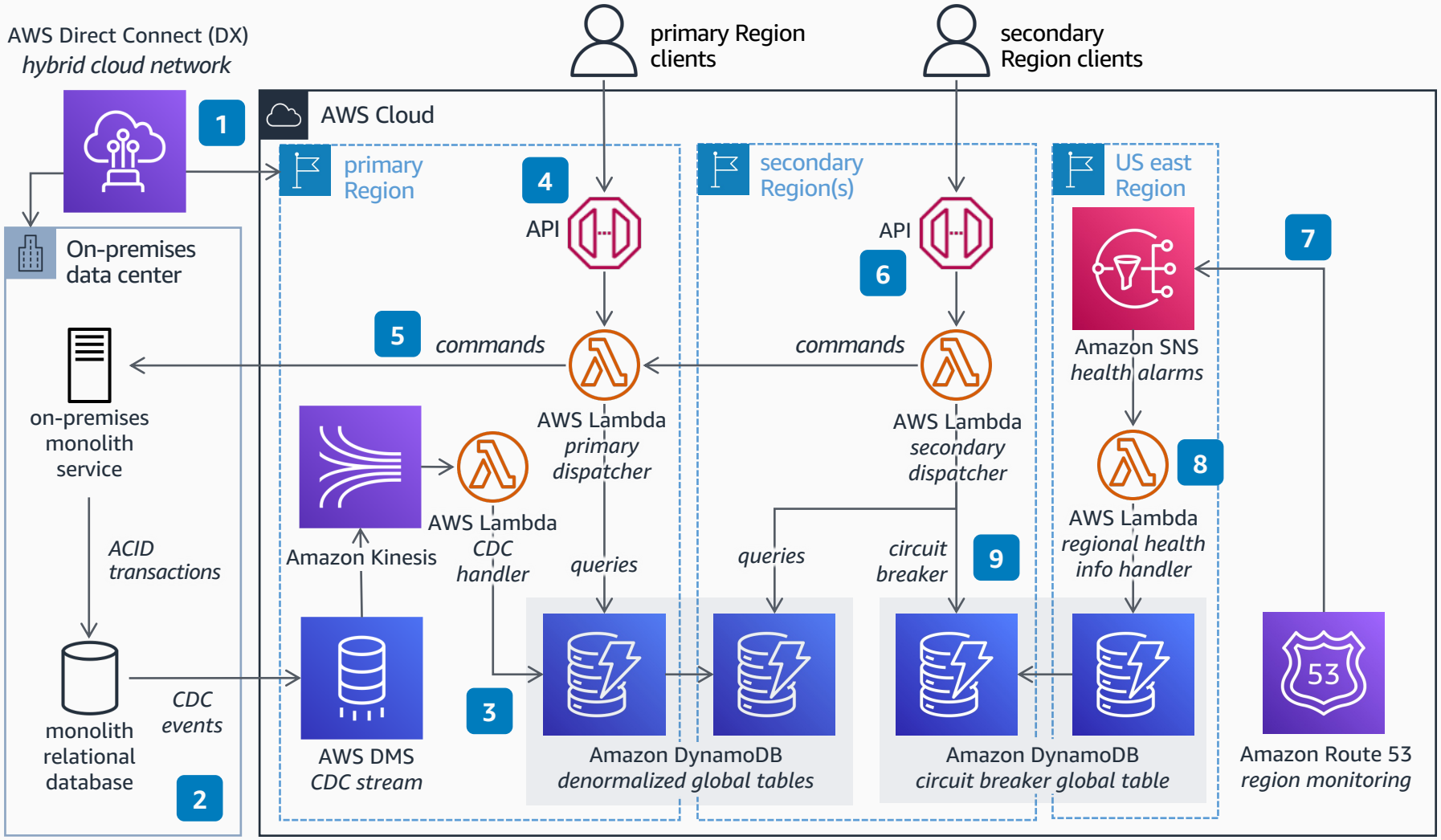
Multi-Region CQRS for On-premises Monoliths

Expand your on-premises transactional monolith to the cloud using the AWS global network. Reduce latency for your end-users around the world, while maintaining the transactional isolation of your monolith and on-premises services, using the Command Query Responsibility Segregation (CQRS) and the read-local write-global architectural patterns.



1. Ensure low-latency hybrid connectivity by deploying an **AWS Direct Connect** (DX) connection between your on-premises installations and the nearest AWS Region.
2. To capture the changes resulting from atomicity, consistency, isolation, and durability (ACID) transactions,, set up a change data capture (CDC) stream from your on-premises relational database by attaching **AWS Database Migration Service** (AWS DMS) to the source database, and then streaming the events with **Amazon Kinesis Data Streams**.
3. Handle **Kinesis** CDC events with **AWS Lambda**, storing them in denormalized **Amazon DynamoDB** global tables.
4. Expose an API to your customers in the Region with **Direct Connect** (the primary Region), and handle query requests by reading the denormalized data in **DynamoDB**.
5. Handle the client's command requests by dispatching them to the on-premises monolith service, to continue to support transactional isolation and atomicity.
6. Expose additional APIs in one or more secondary Regions, serving read requests with the data replicated to the local **DynamoDB** table, and dispatching command requests to the primary Region.
7. On the us-east-1 Region, set up **Amazon Route 53** monitoring of your Regions, sending health alarms to **Amazon Simple Notification Service** (Amazon SNS).
8. Handle the health alarms with **Lambda**, storing the state and location of Regional APIs in a **DynamoDB** global table.
9. Set up circuit breakers on the API dispatchers, based on the state of the Regions available in **DynamoDB**.

Reviewed for technical accuracy July 1, 2022

AWS Reference Architecture