
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

FSI301

Optimizing database resilience in a multi-Region universe

Using RDS Proxy and Aurora Global Database cross-region failover to achieve high availability and fast disaster recovery

Max Winter

Solutions Architect
Amazon

winmaxim@amazon.com
https://www.linkedin.com/in/maxwinter

Adrian Tarjoianu

Solutions Architect
Amazon

tarjoadi@amazon.com
https://www.linkedin.com/in/adriantarjoianu

Brooke Mullen

Resiliency Architect
Amazon

mulbrook@amazon.com
https://www.linkedin.com/in/brookeemullen

Agenda

1. Use Case Overview
2. In-Region Failover + Demo
3. Warm Standby Regional Failover + Demo
4. Hot Standby, Fully Synchronized Regional Failover
5. Q&A

Use case overview

Customer Use Case: Broadridge Financial Advisor Workstation

Broadridge's Advisor Workstation is the ecosystem in which all Broadridge, firm-proprietary, and 3rd party applications are rendered through a widget based interface for wealth managers and advisors.

The Workstation, powered by our open architecture integration platform with a wealth data layer, APIs, messaging and micro-services (Enterprise Integration Services Layer - EISL).

The Workstation provides access to all the wealth tools that an advisor needs to personalize their interaction with clients, monitor communication, determine next best actions and recommendations, efficient access to brokerage and advisory business needs, such as performance reporting, client maintenance workflow actions, and advanced dashboard elements, including trade initiation, financial planning, CRM, managed accounts, etc.

Broadridge®

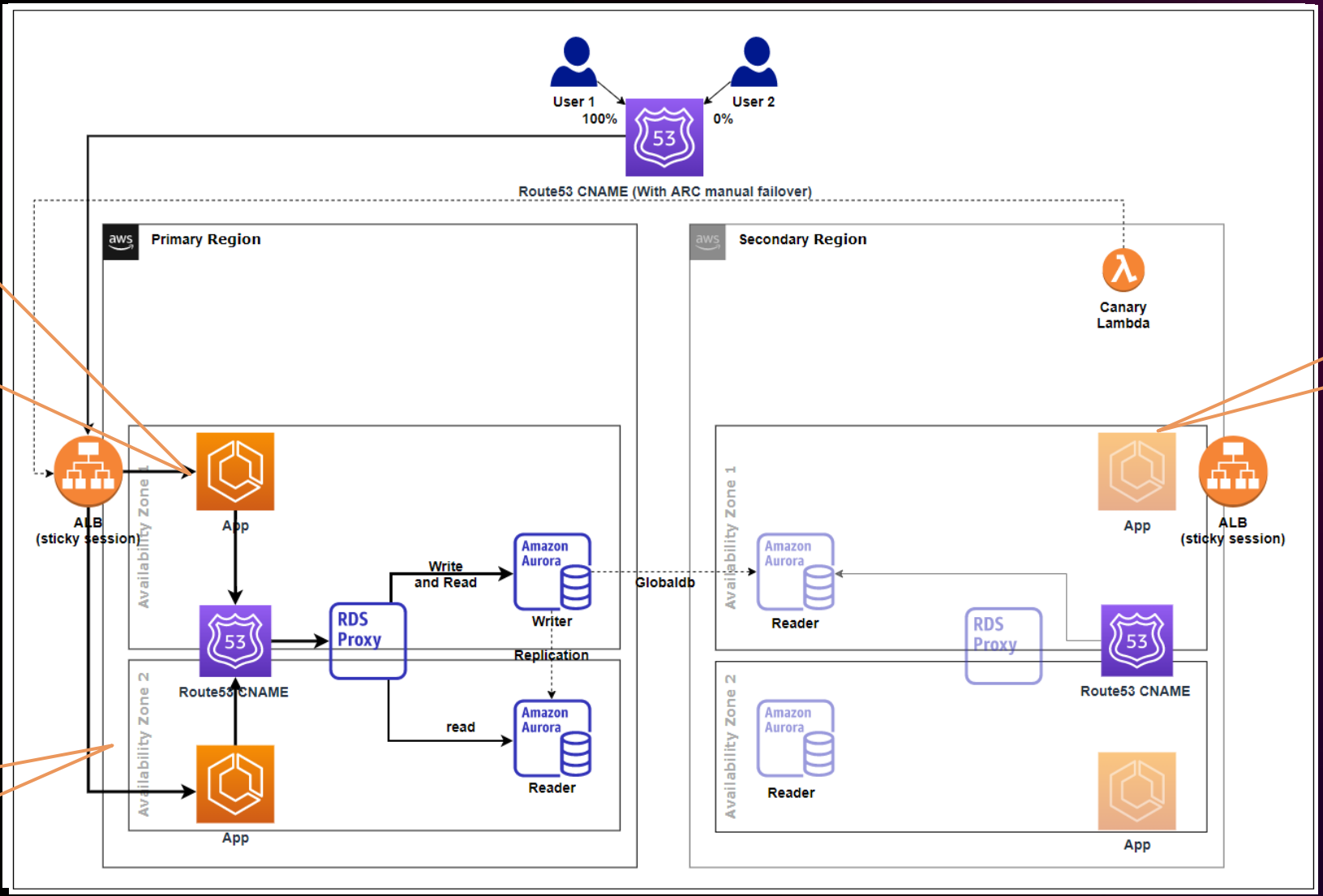
Use case

API service

Application and database

High availability required

Regional disaster recovery required



Multi-Region for disaster recovery

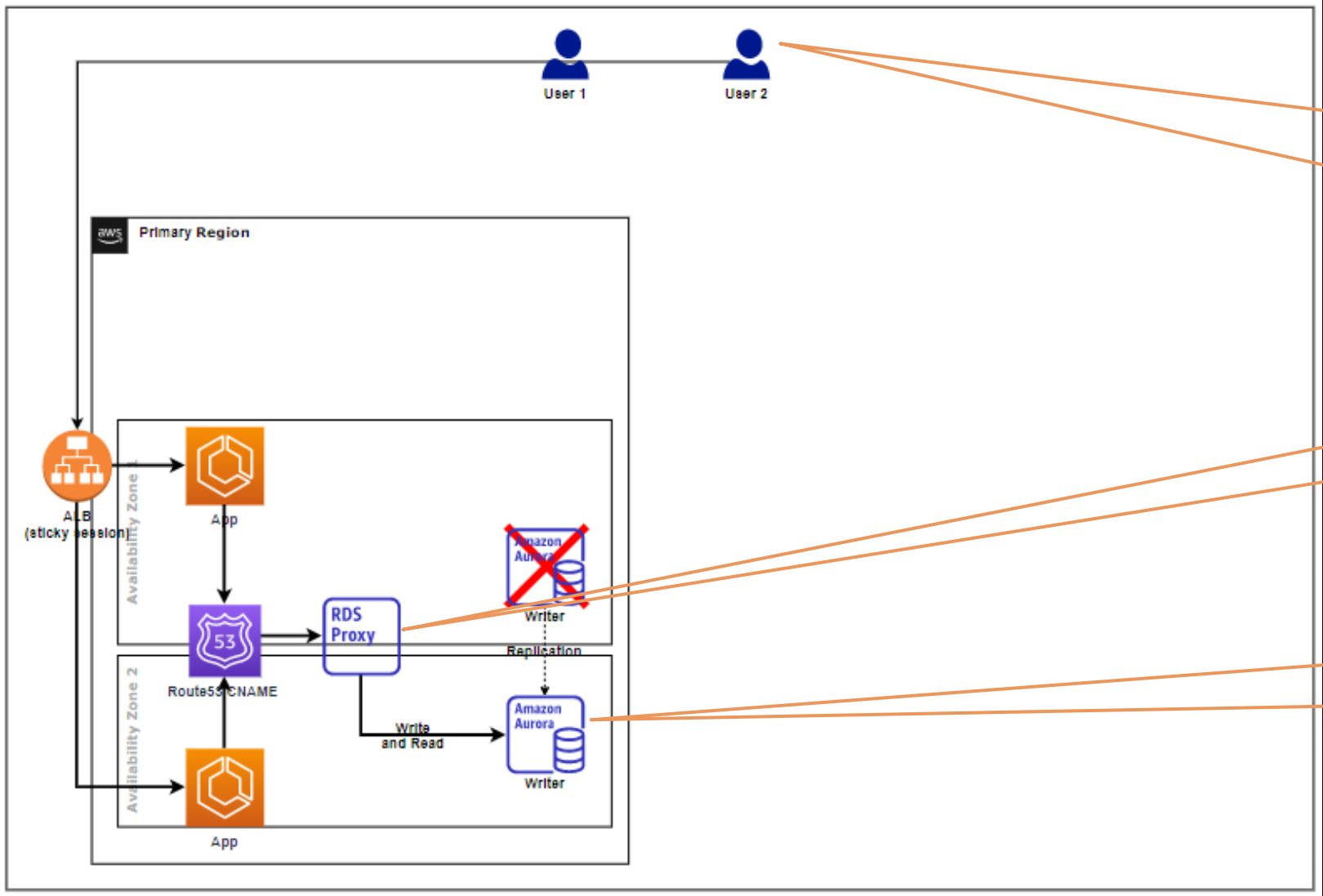
Multi-AZ for high availability

Objectives

- Minimal client/user impact during service or Regional outage
- Every minute counts, so no time for human operator actions
- Minimal RTO/RPO failover within and between Regions
- Automated failover
- Reusable resiliency pattern
- AWS managed replications options
- Work-around for control plane outage

In-Region failover

In-Region failover



If the Aurora writer fails

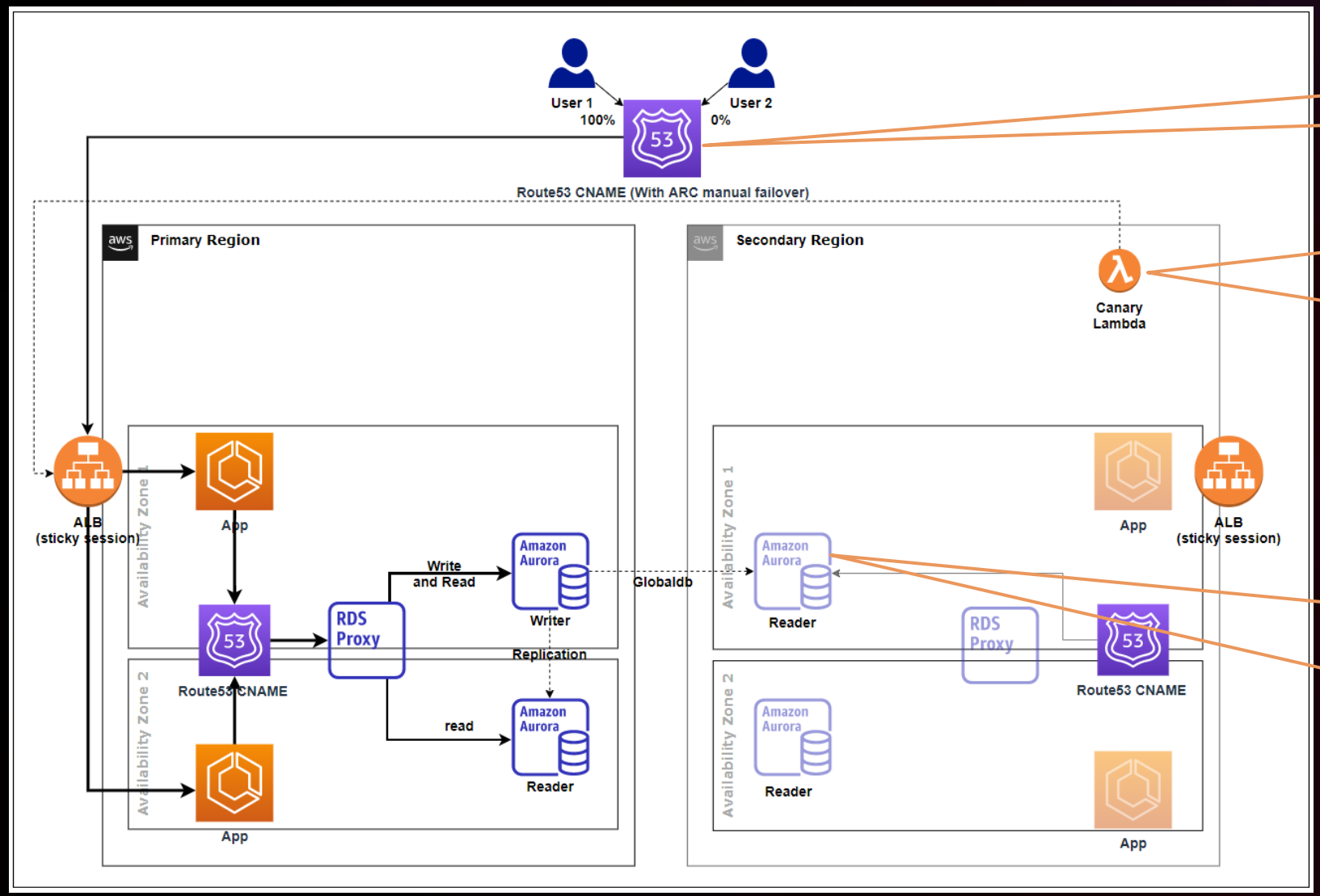
The application needs to retry any writes and long-running reads and transactions that were running on the failing writer (those have to return errors)

Amazon RDS Proxy queues writes/reads until a reader is promoted to writer

Near-synchronous replication ensures the reader is up to date (tens of ms before failover)

Warm standby
Regional failover

Warm standby – Normal operation



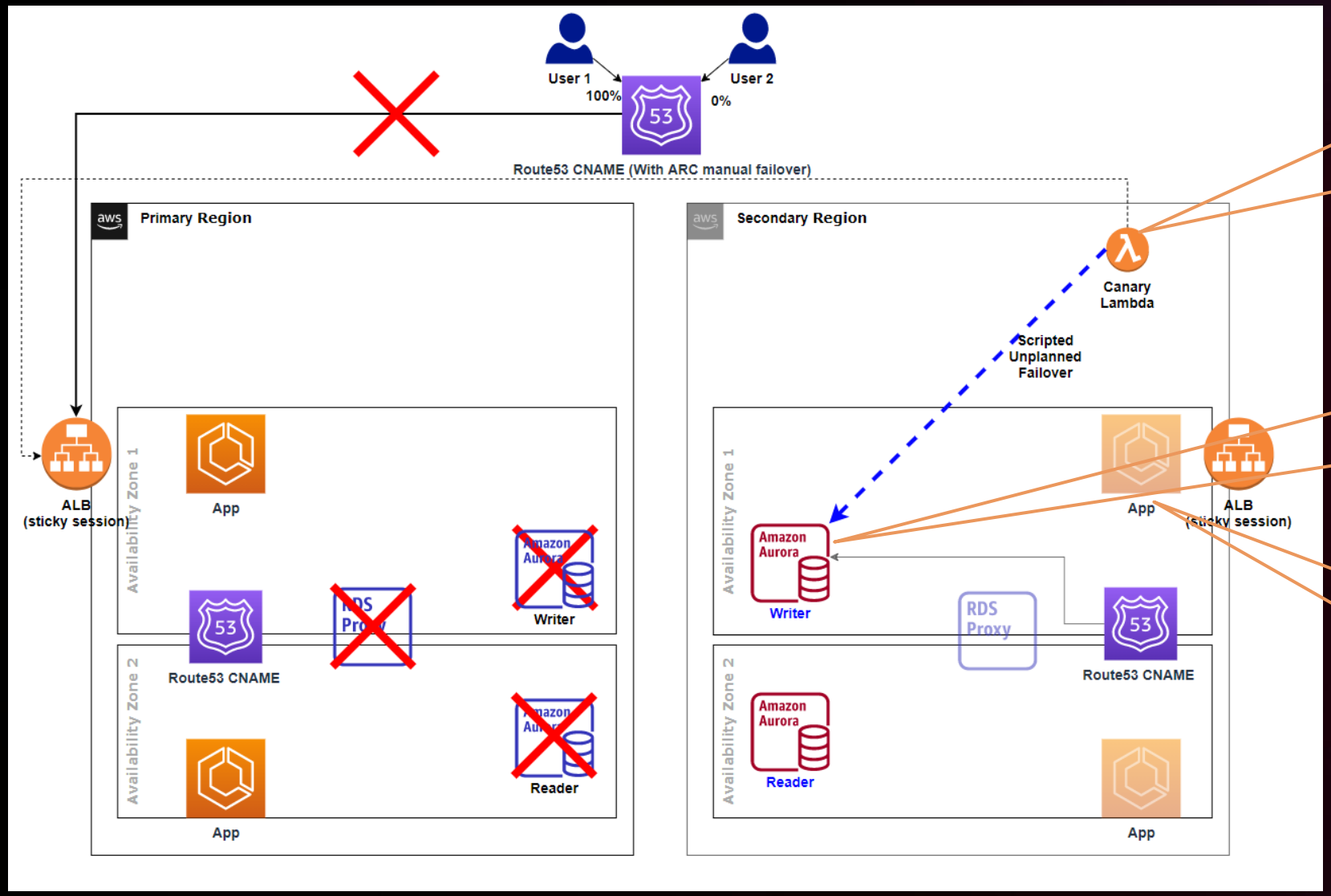
Route 53 CNAME weighted to 100/0, so all traffic goes to the primary Region

Canary Lambda checks the entire application works in the primary Region, and fails it over if the application is unavailable for over 30 seconds (3 consecutive failures, each 10 seconds apart)

Aurora global DB replication

- AWS managed replication
- RTO: minutes, RPO: seconds
- Very simple to implement/maintain

Warm standby – Full Region (or app) failure

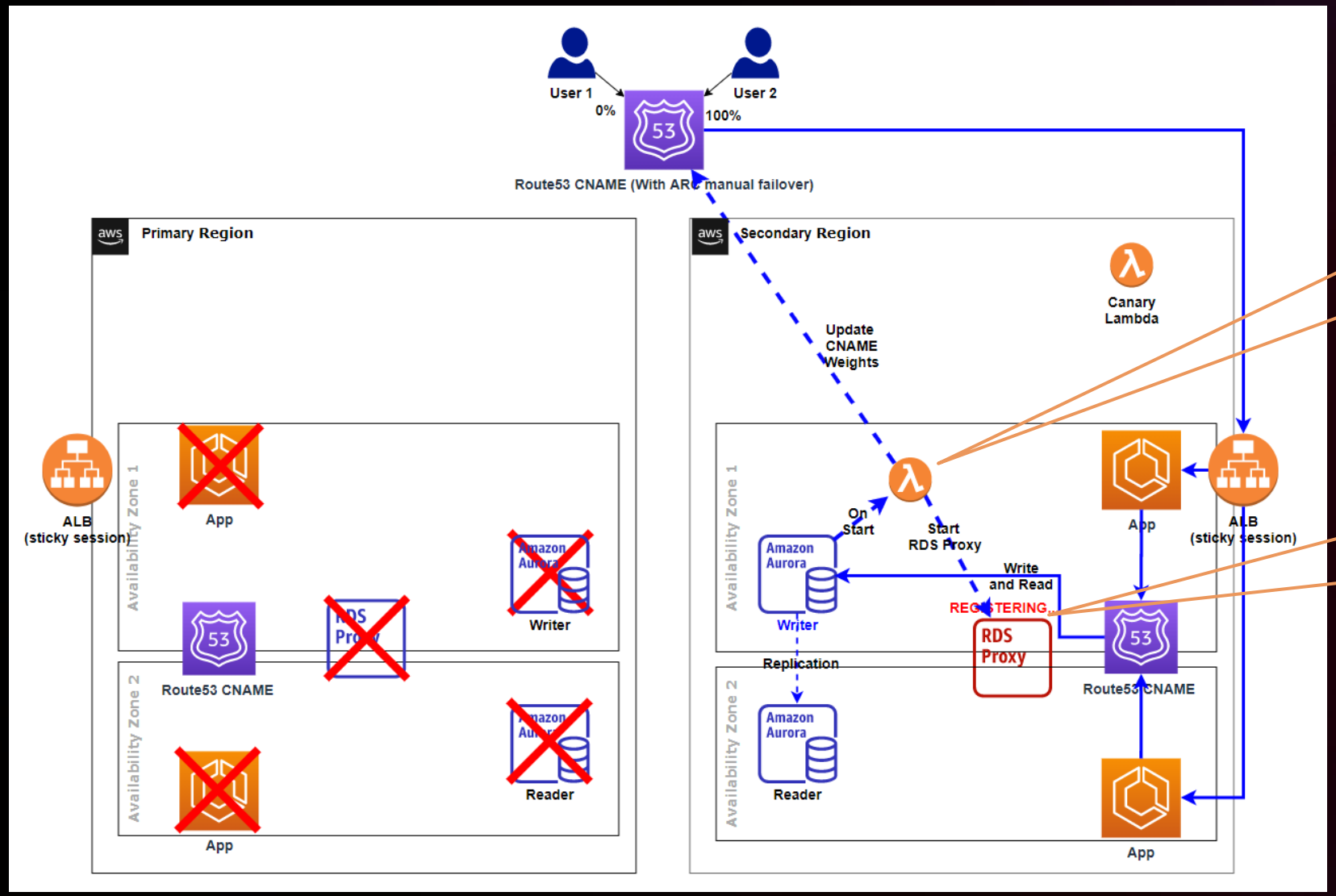


Since the whole primary Region is unable to serve requests, the canary begins failover of the database cluster to the secondary Region

Asynchronous replication likely delivers any in-flight writes to the secondary region's database (unless the replication itself is interrupted)

Until the Aurora writer instance is up, applications continue to get DB connection errors

Warm standby – Secondary Region Amazon Aurora writer starts

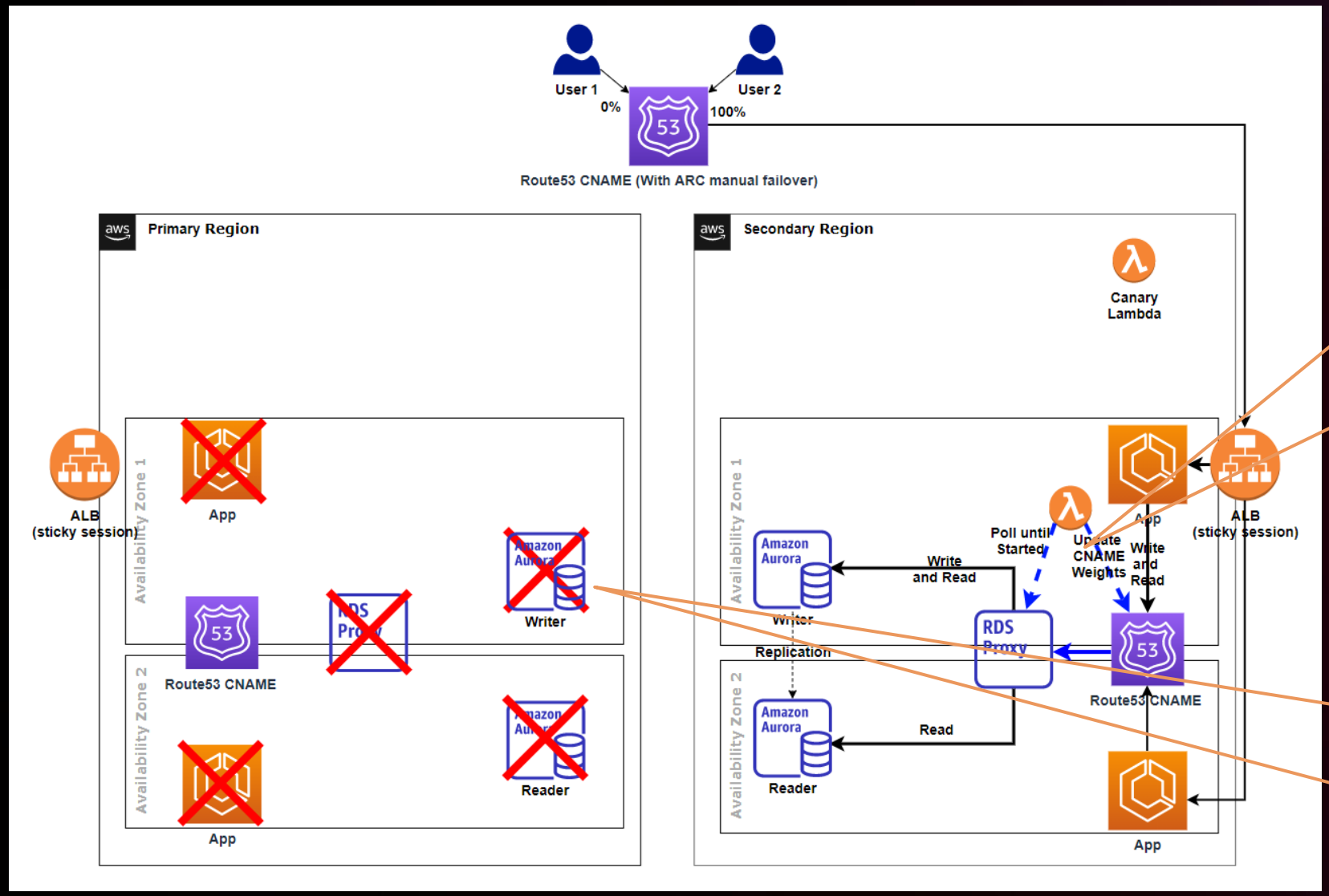


Once the backup Aurora writer starts

Startup event triggers Lambda that updates the Route 53 CNAME weighting to direct all traffic to the backup Region

The Lambda starts the Amazon RDS Proxy connection process while the CNAME sends traffic directly to Aurora

Warm standby – Secondary Region Amazon RDS Proxy takes over



Once Amazon RDS Proxy starts

AWS Lambda detects Amazon RDS Proxy is up, and re-points the CNAME to it

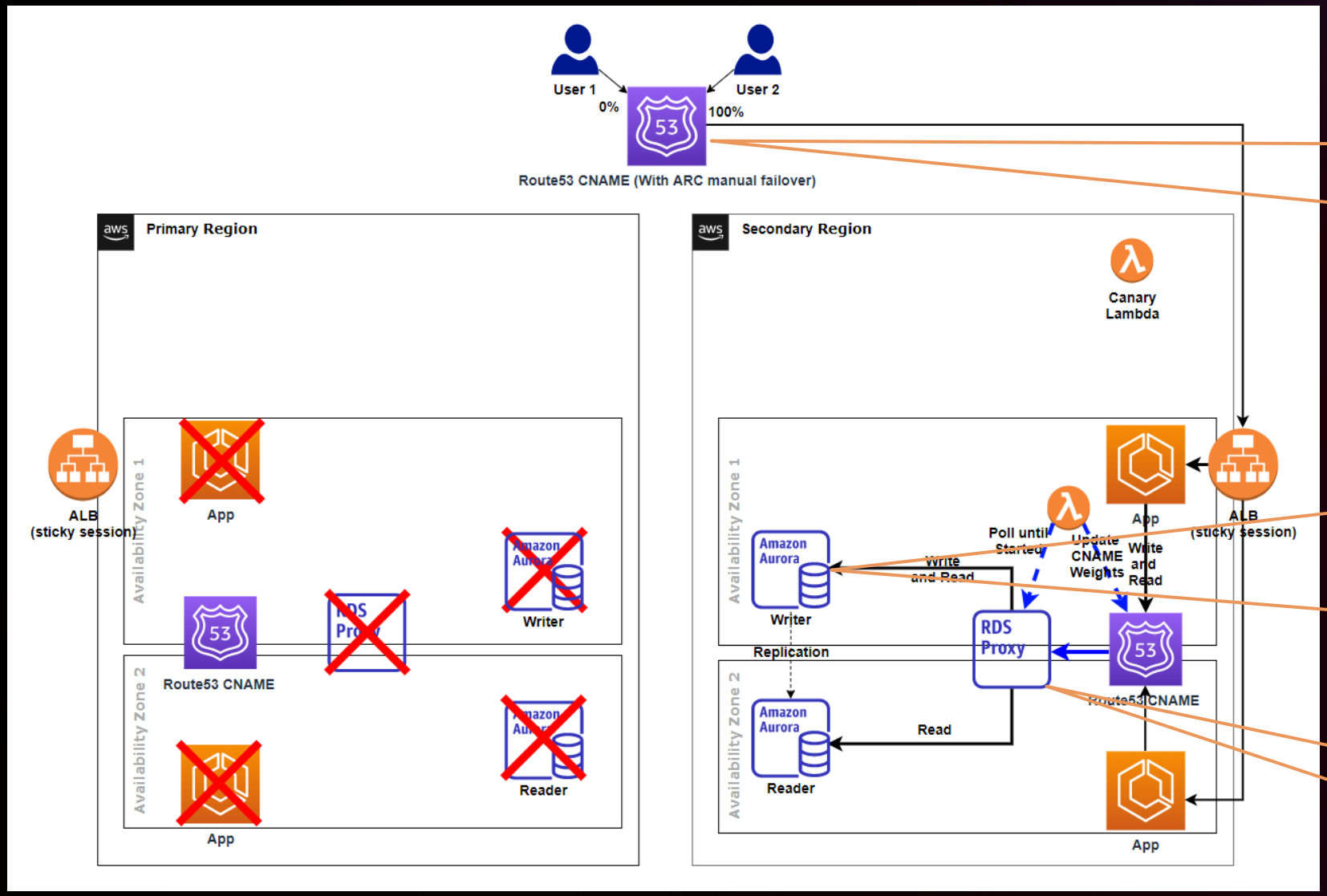
This re-enables high-availability cross-AZ failover, and avoids several more minutes of Amazon RDS Proxy startup time impacting the app

Once primary Region event is over

Aurora databases there should be recreated as global DB headless nodes

Fail-back to the original Region would happen identically to the failover

Warm standby – Caveats



If the primary region's Route 53 control plane is down, an ARC (Amazon Route 53 Application Recovery Controller) manual failover is pre-configured as a last-ditch override to re-route traffic to the secondary region in minutes

Aurora reader startup as a writer can take up to 8 minutes (usually 1–2)

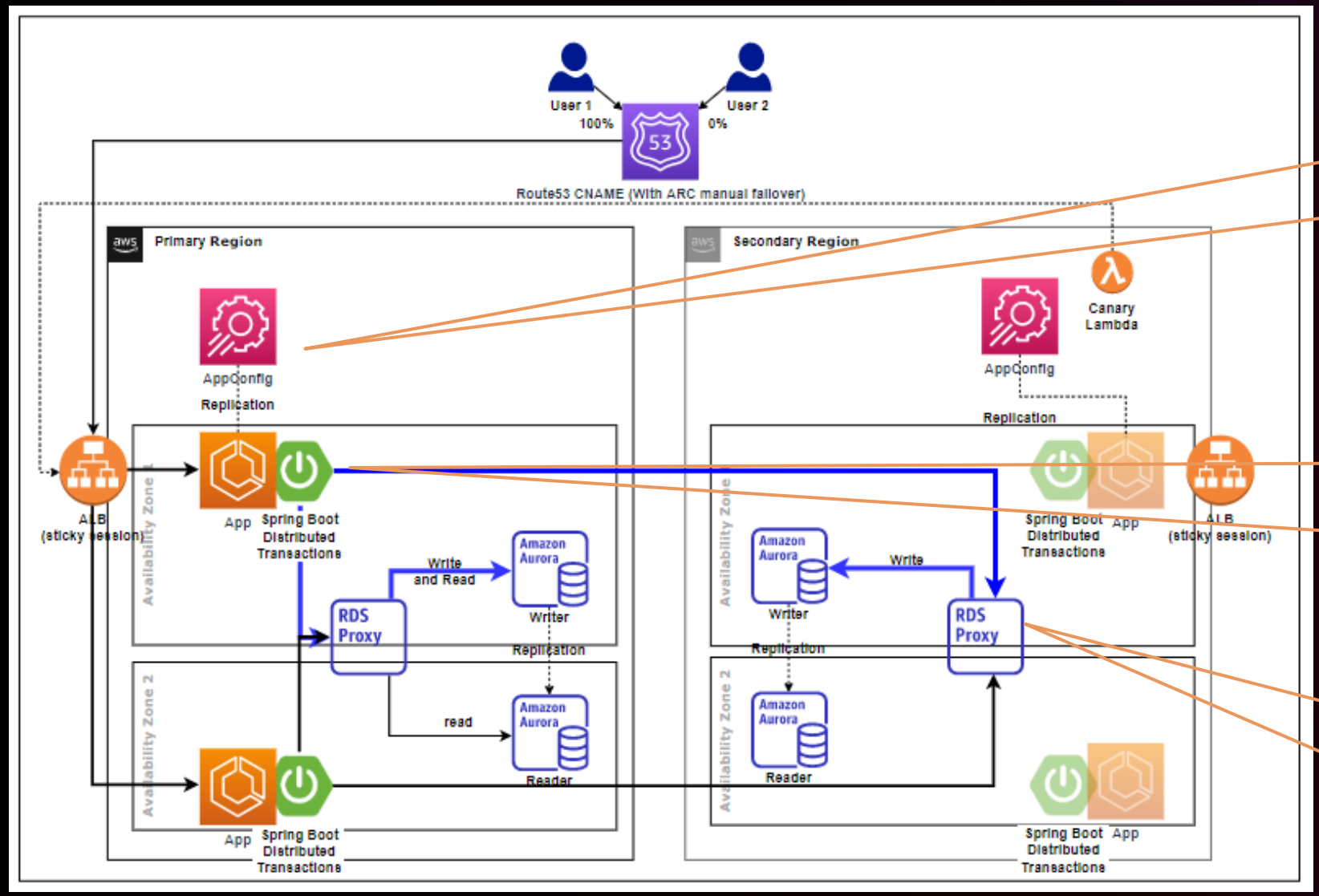
If using "headless" rather than "reader" Global DB resilience to save cost, startup will need to provision an instance, which will take several minutes longer

A serverless endpoint is also an option to save cost, but can't scale up as high as provisioned

RDS Proxy takes significant time to set up, so secondary region initially uses Aurora directly

Hot standby, fully synchronized Regional failover

Synchronized hot standby (work in progress)

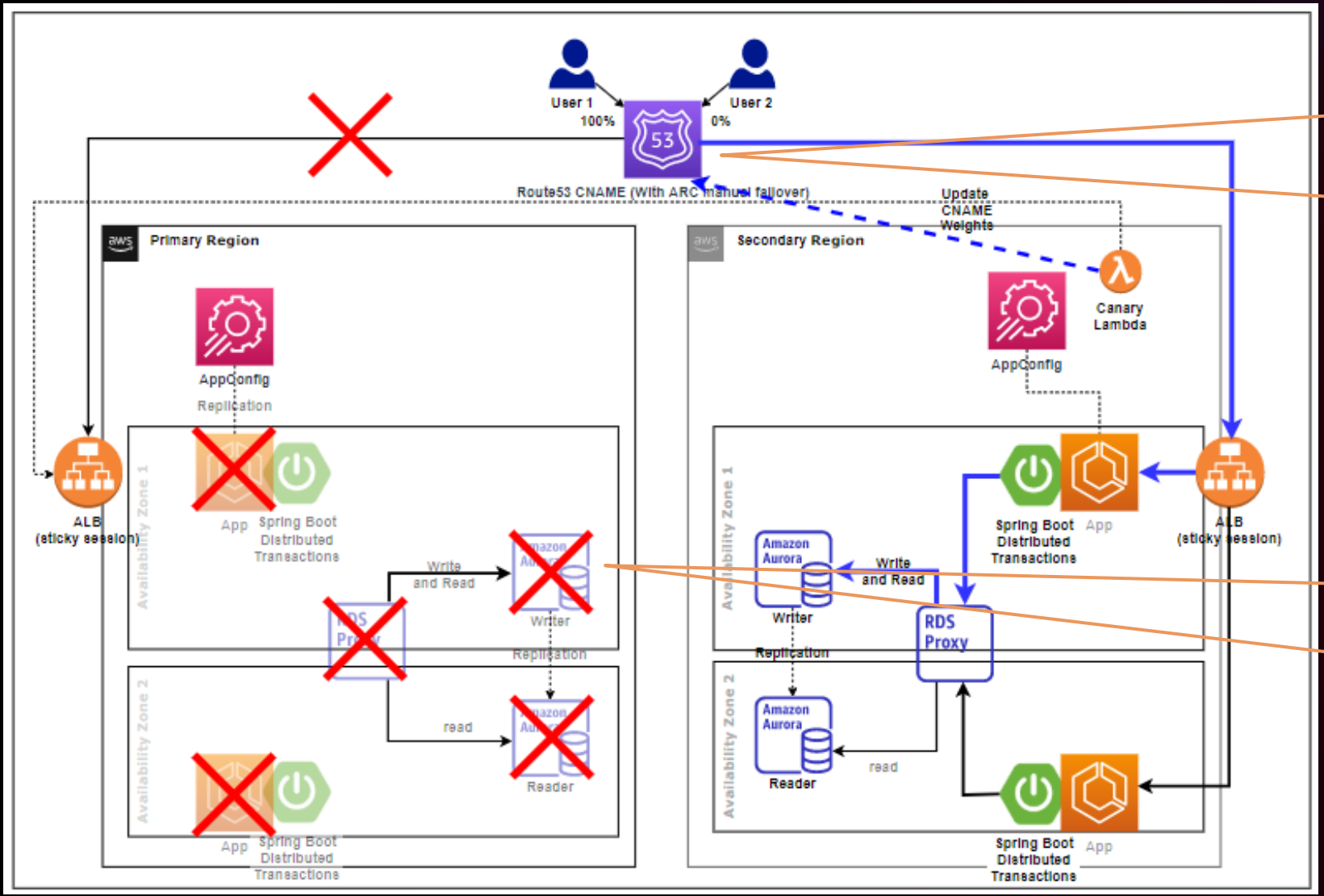


AWS AppConfig contains a Region list that must be written for a transaction to be considered to be committed successfully

Application makes database updates via a distributed transaction library (i.e., Narayana) to write to all Regions listed in AppConfig to achieve RPO=0

Caveat: Synchronous replication means a round trip to the secondary Region for every write, adding tens or hundreds of milliseconds to each write

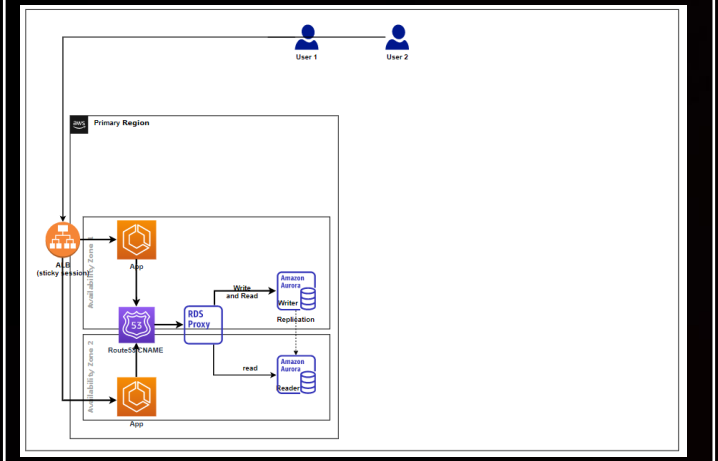
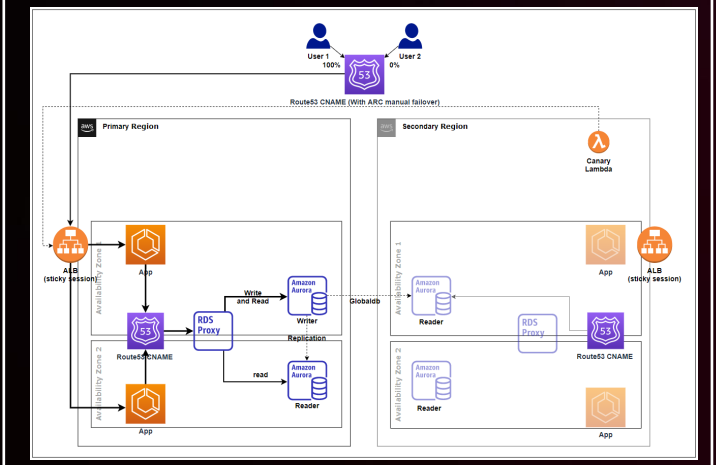
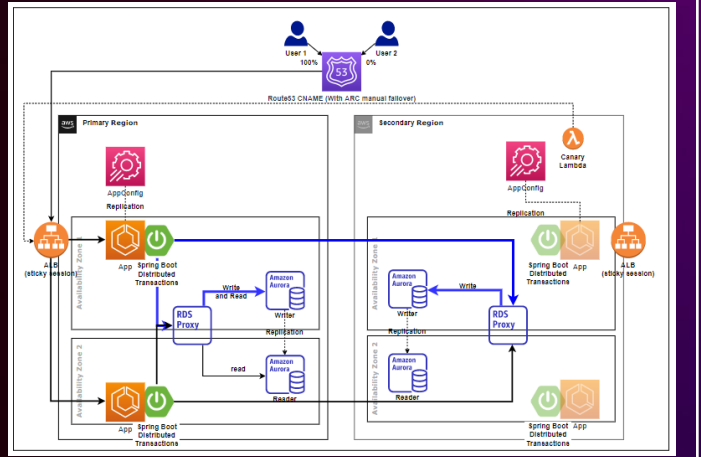
Synchronized hot standby (work in progress)



In the event of continuous errors in the primary Region, we can switch over the application routing via the CNAME in seconds and have immediate write access to an identical secondary database

To restore normal (2-Region) operation, we'd need to stop the application writes, re-sync the databases, and restart transaction replication during the next maintenance window

Pros and cons

	Multi-AZ Amazon RDS Proxy	Multi-Region warm standby	Multi-Region hot standby
			
X-Region	No	Yes	Yes
RTO	Seconds	Single-digit minutes	Seconds
RPO	Milliseconds	Seconds	Zero
Latency	Milliseconds	Milliseconds	10s-100s of milliseconds (Based on region distance)
Fail-back	Automatic	Easy (add reader)	Manual (maintenance window)

Q&A

You can find the template to recreate this demo here:

https://github.com/aws-samples/amazon-aurora-postgresql-fast-failover-demo

Thank you!

Max Winter

Solutions Architect
Amazon

winmaxim@amazon.com

https://www.linkedin.com/in/maxwinter

Adrian Tarjoianu

Solutions Architect
Amazon

https://www.linkedin.com/in/adriantarjoianu

Brooke Mullen

Resiliency Architect
Amazon

mulbrook@amazon.com

https://www.linkedin.com/in/brookeemullen

Please complete the session survey in the **mobile app**