Demystifying PostgreSQL High Availability in the Cloud

Arpitha Vasudeva (She/Her)

Technical Account Manager
Amazon Web Services

October 17, 2025
PGDU, Sydney, Australia

Agenda

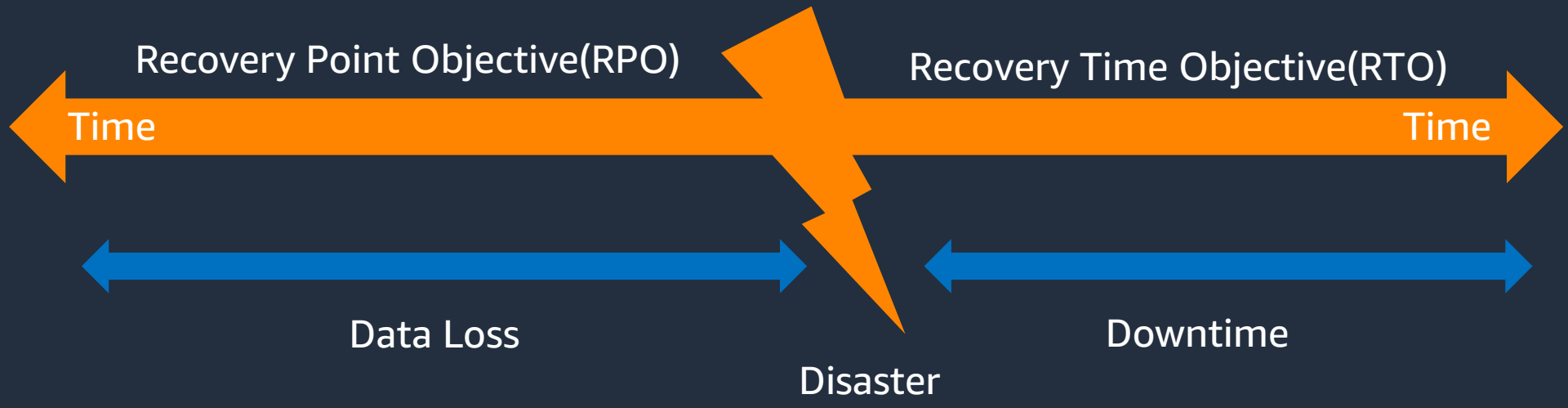
- What is High Availability?
- Metrics for Evaluation
- Native PostgreSQL Architectures
    - Log Shipping
    - Log Streaming
- Cloud Native Architectures
    - Amazon RDS
    - Amazon Aurora
- Decision Matrix
- Common Use Cases

What is High Availability (HA)?

“High availability isn't just about preventing failures — it's about maintaining service despite them.”

How much data can you afford to recreate or lose?

How quickly must you recover? What is the cost of Downtime?



Metrics for Evaluation

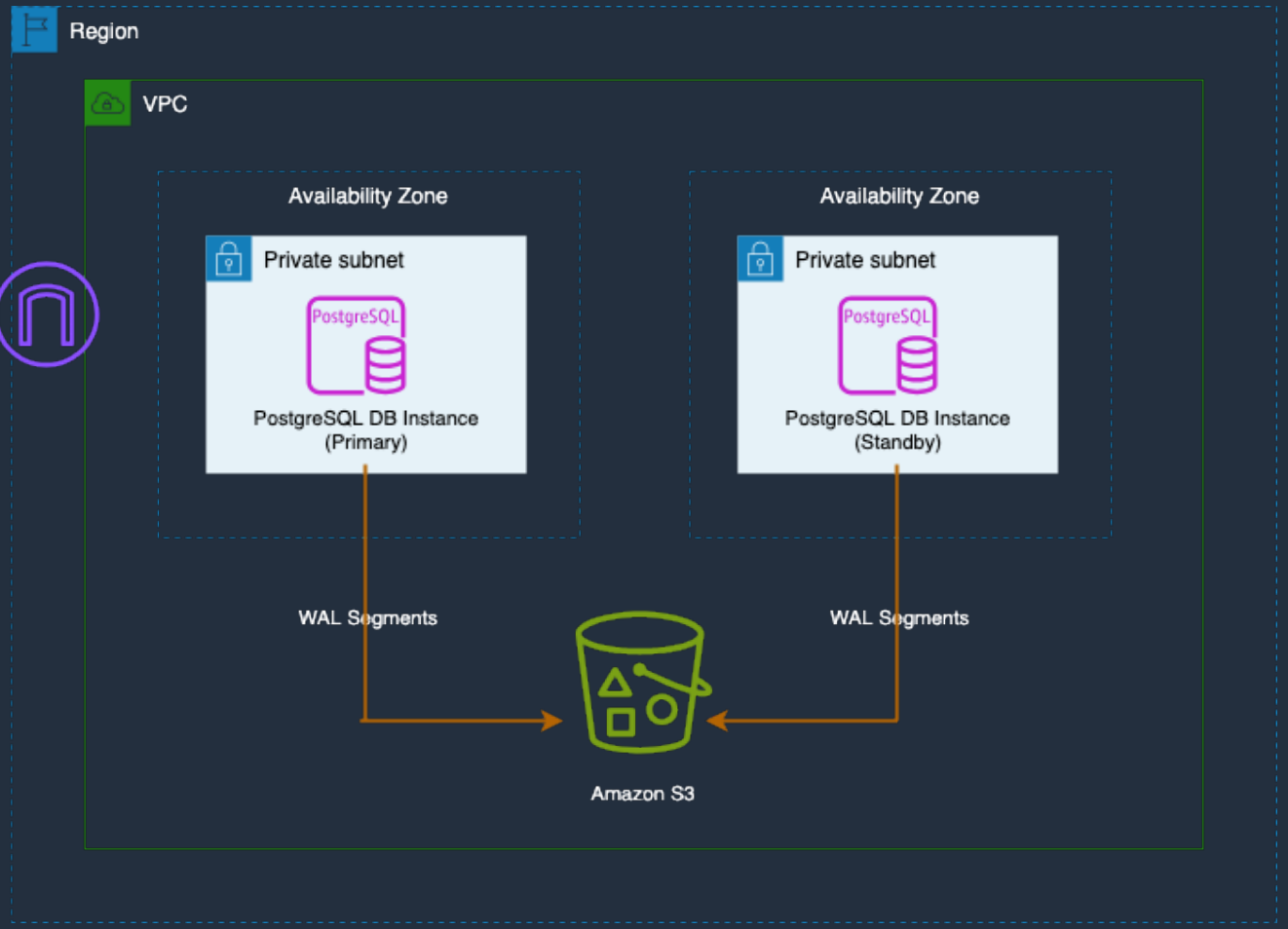
Design Pillers	Metrics
Availability & Recovery	RTO, RPO, Failover Time
Scale	Read/Write Scaling
Resilience	Geo Distribution
Operations & Flexibility	Effort, Extensibility
Business	Cost

Native PostgreSQL Options

Native PostgreSQL Architectures

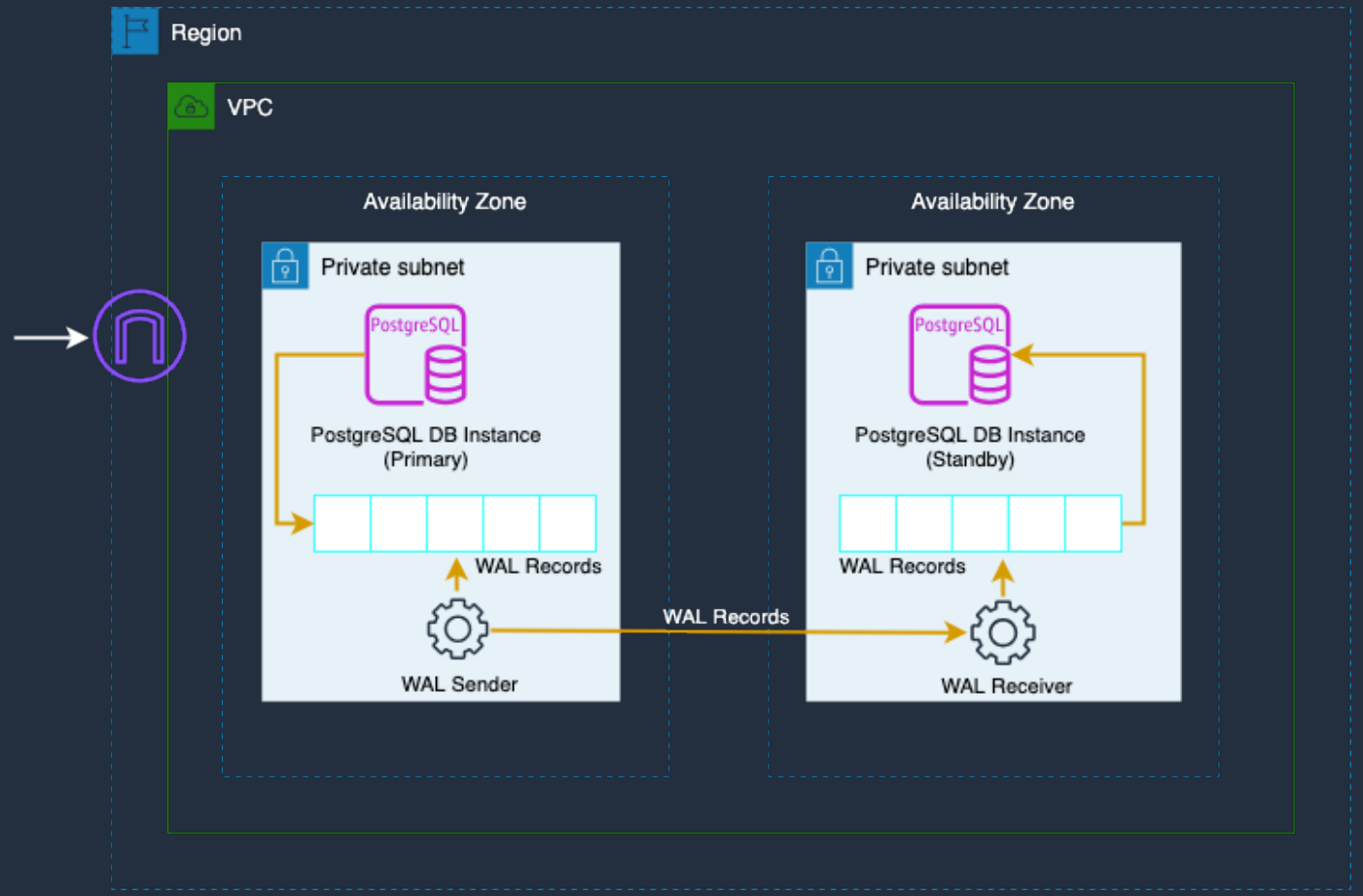
1. Log shipping
2. Streaming replication

Log Shipping



- Decoupled Solution
- Primary is in continuous archive mode
- WAL segments are fetched and applied to the standby
- Asynchronous Solution
- Possible data loss during failover, use *archive_timeout* parameter to reduce the data loss window
- Standby can be used as hot standby for read scaling

Streaming Replication



- Supports synchronous and asynchronous (default) replication
- Maintains direct connection between primary and standby
- Replication slots ensure WAL retention until consumed
- Near real-time compared to log shipping solution
- Used for High Availability and Disaster recovery

Challenges

- With Log Shipping
    1. Shared Storage Management
    2. Manual Failover Process
- With Streaming Replications
    1. Network Connectivity & Management
    2. Complex Infrastructure Orchestration

Patroni for Automated Failover

- Complete Control and Flexibility
- Cloud Agnostic Solution
- Automated Failover with Customizable Timing
- Switchover Capabilities for Maintenance

Challenges:

- Operational Complexity
- Infrastructure Management

Cloud Native Solutions

Cloud Native Architectures

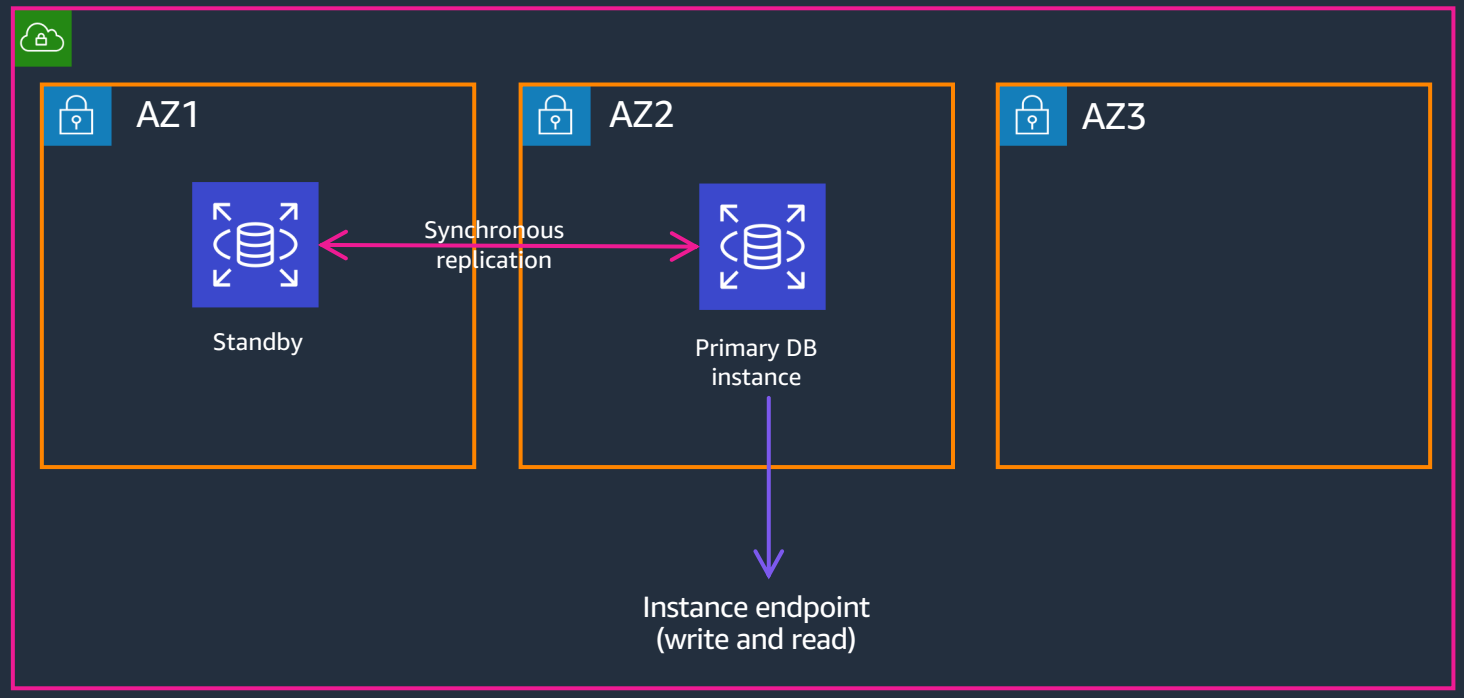
Amazon RDS Specific Architectures

- RDS Multi-AZ with One Standby
- RDS Multi-AZ with Two Readable Standbys
- RDS with In-Region Read Replicas
- RDS with Cross Region Read Replicas

Amazon Aurora Specific Architectures

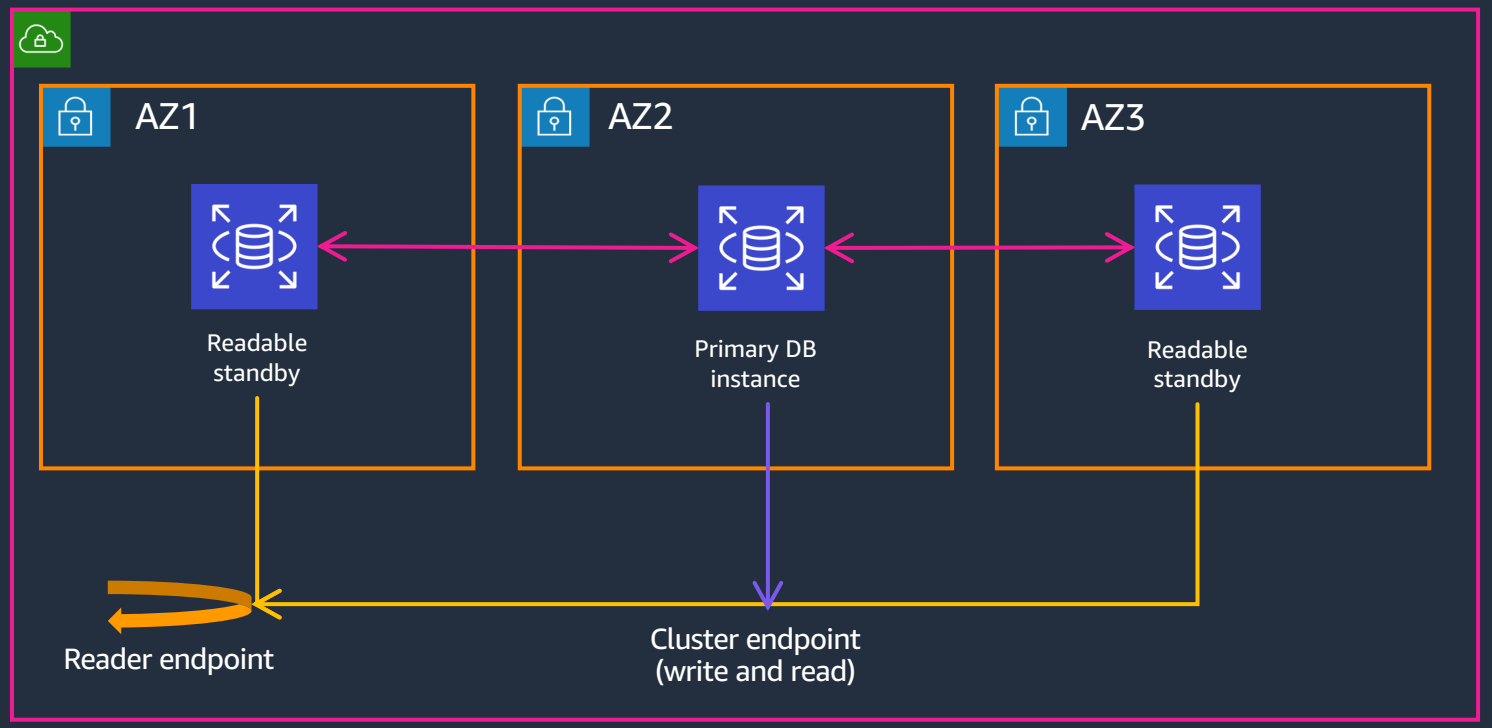
- Amazon Aurora Cluster
- Amazon Aurora Global Database Cluster

RDS Multi-AZ with One Standby



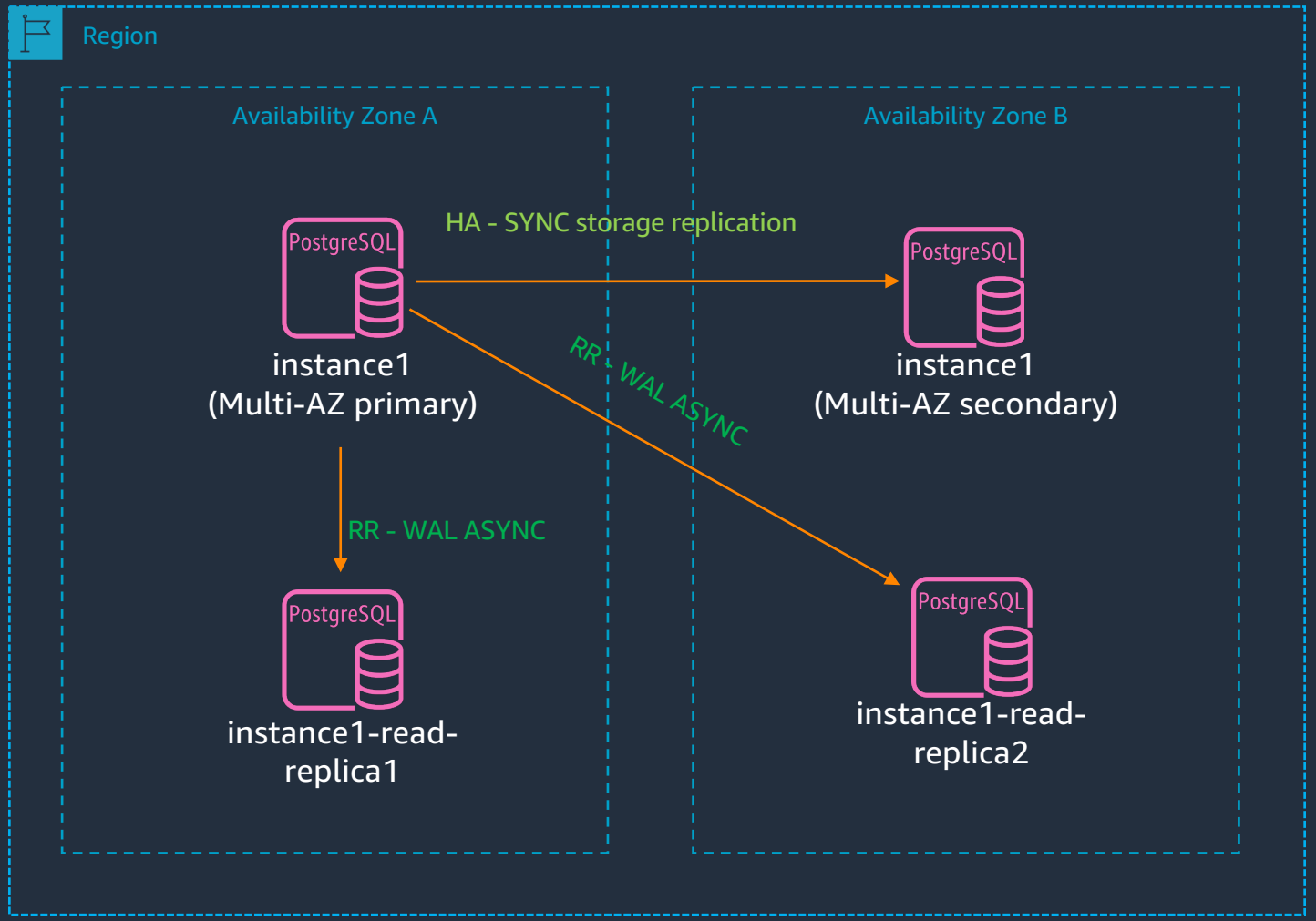
- Synchronous replication between primary and standby
- Zero data loss during failover (RPO near zero)
- Automatic Failover ~60-120 seconds
- Automatic backups and point-in-time recovery (PITR)
- Automatic Storage scaling

RDS Multi-AZ with Two Readable Standbys



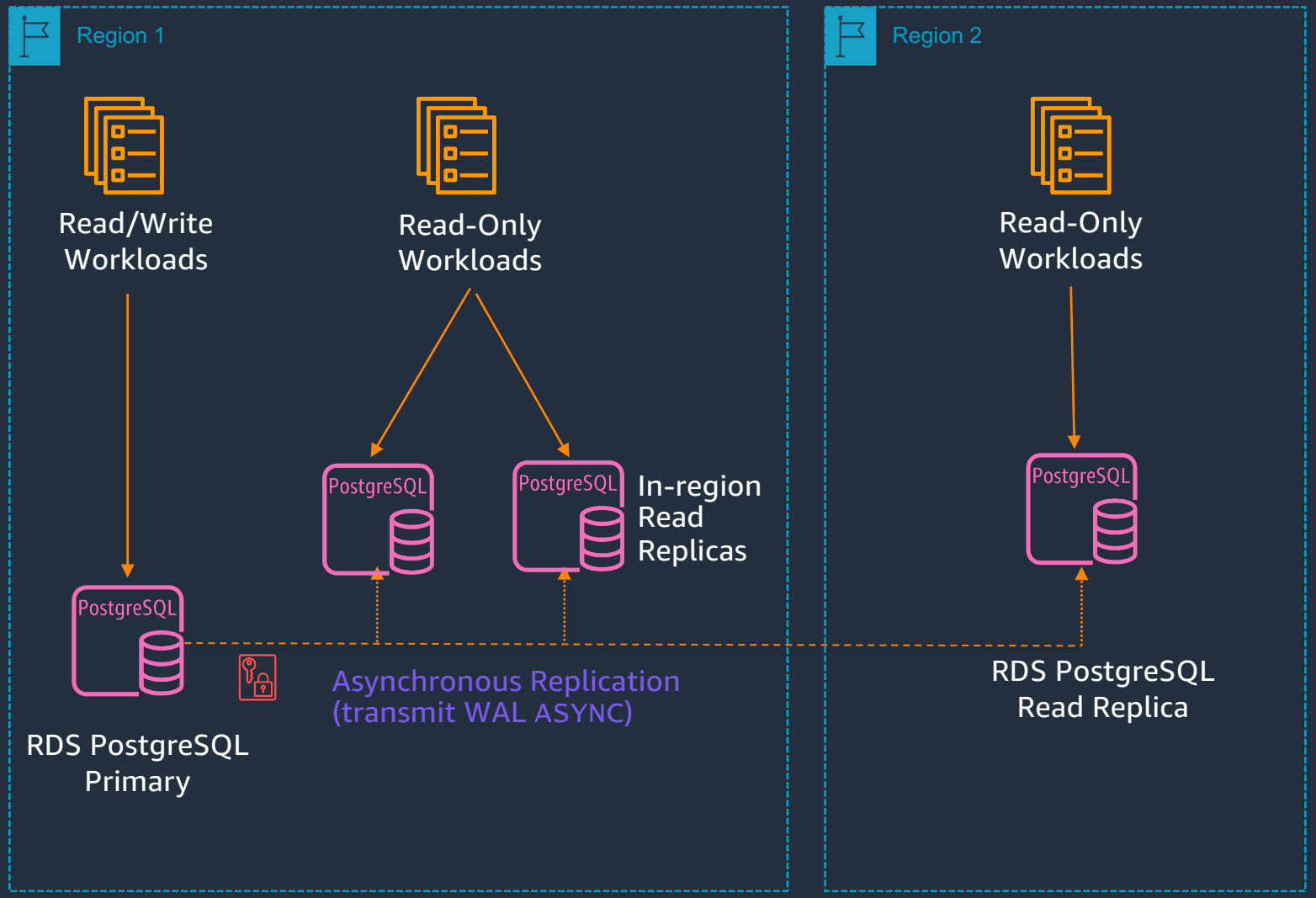
- One primary two readable standbys
- Standbys across different AZs and serve read traffic
- Synchronous replication between both standbys
- Both standbys participate in automatic failover
- Improved availability during maintenance
- Zero RPO during Failover
- Reduced RTO (typically <35 seconds)
- Automatic backups, PITR and storage scaling

RDS PostgreSQL Read Replicas



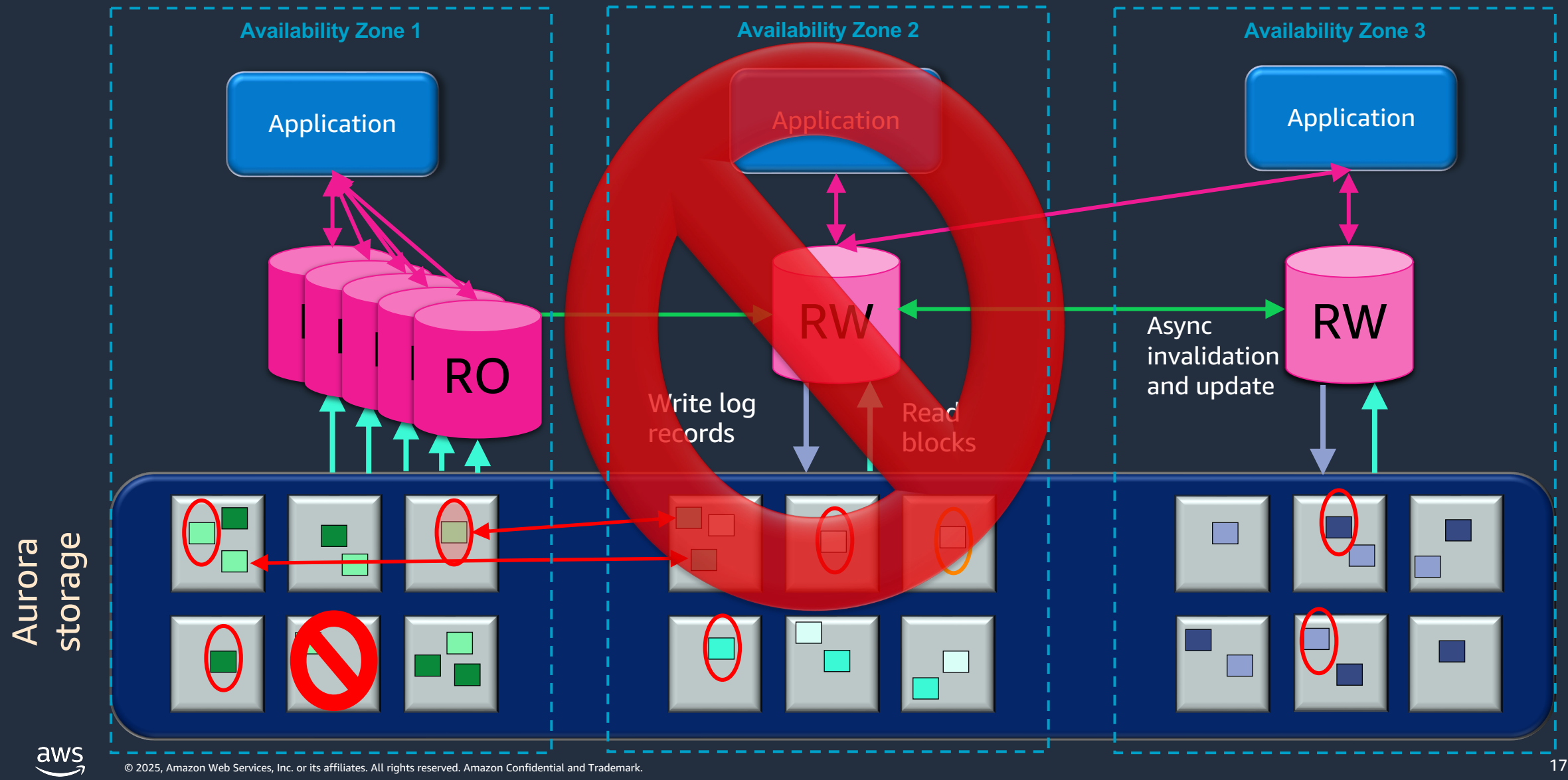
- Asynchronous Streaming Replication – some lag expected
- Scale read and serve read-heavy workloads
- Manual failover process
- Connection management needed

RDS with Cross Region Read Replica

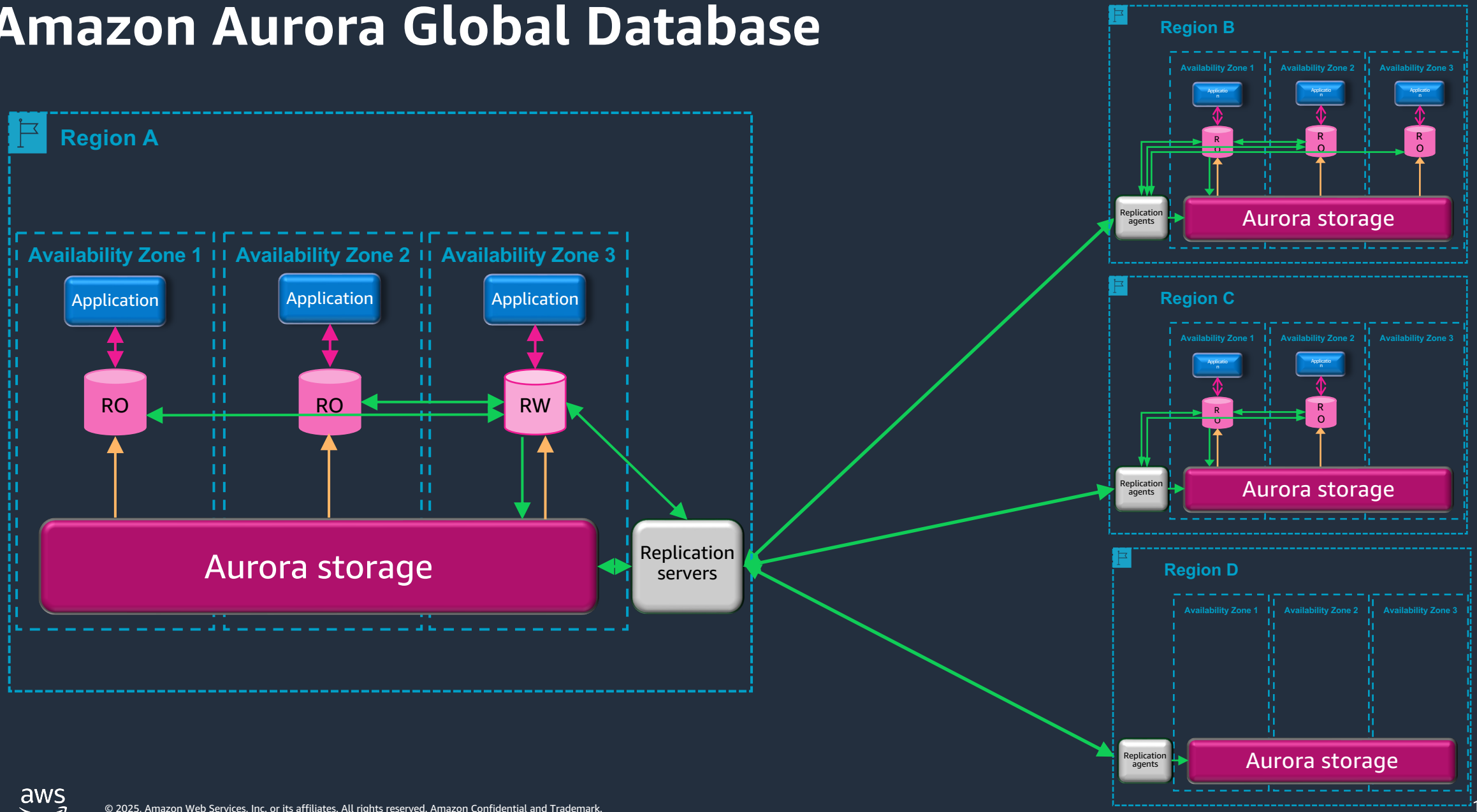


- Global read scaling
- Disaster recovery Option
- Asynchronous Replication
- Adds Network Latency
- Adds Replication Lag
- RPO an be minutes
- Manual promotion required
- RTO upwards of 15minutes

Amazon Aurora Cluster



Amazon Aurora Global Database



Decision Matrix and Common Use Cases

Architecture	Availability & Recovery			Scaling	Resilience	Operations & Flexibility		Business
	RPO	RTO	Failover Time	Read Scale	Geo-Distribution	Operations Effort	Extensibility	Cost
Single Instance PostgreSQL (Standalone)	⚠ seconds–mins	❌ >30 min manual	❌ >20 min, manual	❌ None	❌ Single host	🔴 High	🟢 Full	🟢 Low
Self-Managed PostgreSQL with Streaming Replication	⚠ <10 s async	⚠ 5–15 min, manual	⚠ 5–10 min, manual	🟢 Manual routing	❌ Single region	🔴 High	🟢 Full	🟢 Low
RDS for PostgreSQL – Multi-AZ Deployment	🟢 <1 s, sync	🟢 ~30-60 s, auto	🟢 ~30–60 s, auto	⚠ Limited standby	⚠ AZ-level	🟢 Low	⚠ Limited	⚠ Medium
RDS for PostgreSQL – Multi-AZ with Two Readable Standbys	🟢 <1 s, sync	🟢 <35 s	🟢 ~30–60 s, auto	🟢 Good (x2 reader)	⚠ AZ-local	🟢 Low	⚠ Limited	⚠ Medium–High
RDS for PostgreSQL with Read Replica(s) In-Region	⚠ seconds; async	⚠ minutes, manual	⚠ minutes, manual	🟢 Excellent	⚠ Multi-AZ only	🟠 Medium	⚠ Limited	⚠ Medium
RDS for PostgreSQL with Cross-Region Read Replica(s)	⚠ seconds–minutes	⚠ manual promotion	⚠ minutes, manual	🟢 Excellent	🟢 Cross-region	🟠 Medium	⚠ Limited	⚠ Medium–High
Amazon Aurora	🟢 ≈0 RPO	🟢 20–30 s	🟢 <30 s, auto	🟢 Excellent	⚠ AZ-level	🟢 Very Low	⚠ Limited	⚠ High
Amazon Aurora Global	🟢 <1 s inter-region	🟢 <1 min	🟢 ≈30 s, auto	🟢 Excellent	🟢 Cross-region	🟢 Very Low	⚠ Limited	🔴 Very High

🟢 Optimal/automated, ⚠ Limited/needs attention, 🔴 Poor/manual, ❌ Not available

Common Use-Cases

Use Case	Typical Team	Recommended Architecture	Why It Fits
Works usually for fast growing Startups/lean ops teams	Early stage/Fast moving engineering teams with minimal DBA bandwidth (99.9% availability, moderate traffic)	RDS Muti-AZ with single standby	Minimal Operational Overhead + Automated Backups and Failover + Reasonable Cost & Simplicity
Scalable SaaS/ Regional High Traffic App	Mature Product, multiple environments and growing read workloads (99.95%+ availability, low latency within a region)	RDS Multi AZ with Read Replicas (In region)	Automated Failover + Read Scalability + Seamless Regional Performance
Global / mission critical Platforms	Fin-tech, e-commerce, or SaaS at global scale (99.99%+ availability , near 0 RPO, Cross region DR)	Aurora Global Database	Fast Failover + Muti-Region Replication + Near-Zero Data Loss

Thank you!

Let’s connect!