pgEdge

How to Unleash Ultra-High Availability and Zero Downtime Maintenance with Distributed PostgreSQL

Presenters

Ibrar Ahmed
Principal Software Engineer

Mike Josephson
Director Solution Engineer

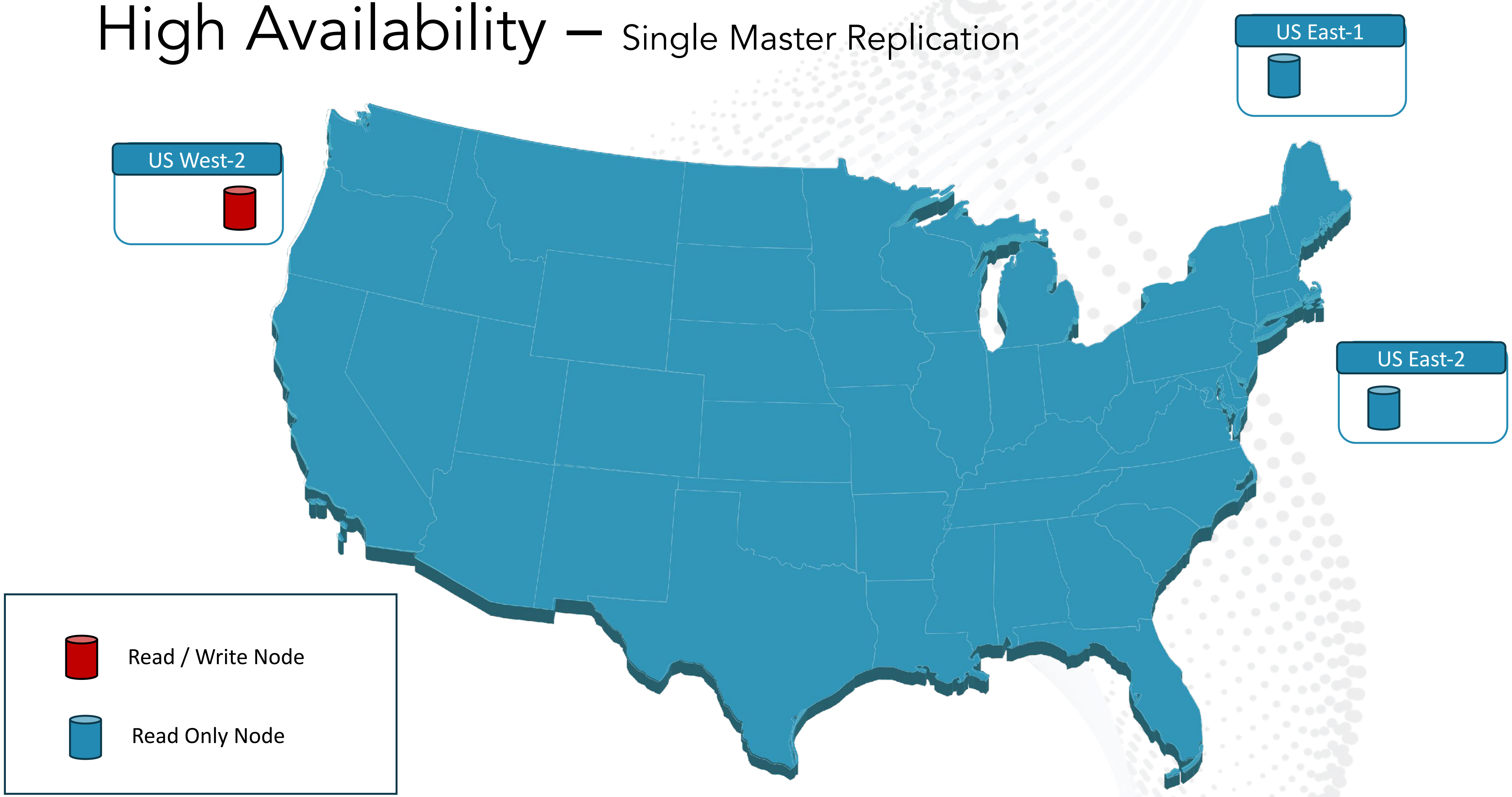
Agenda

High Availability

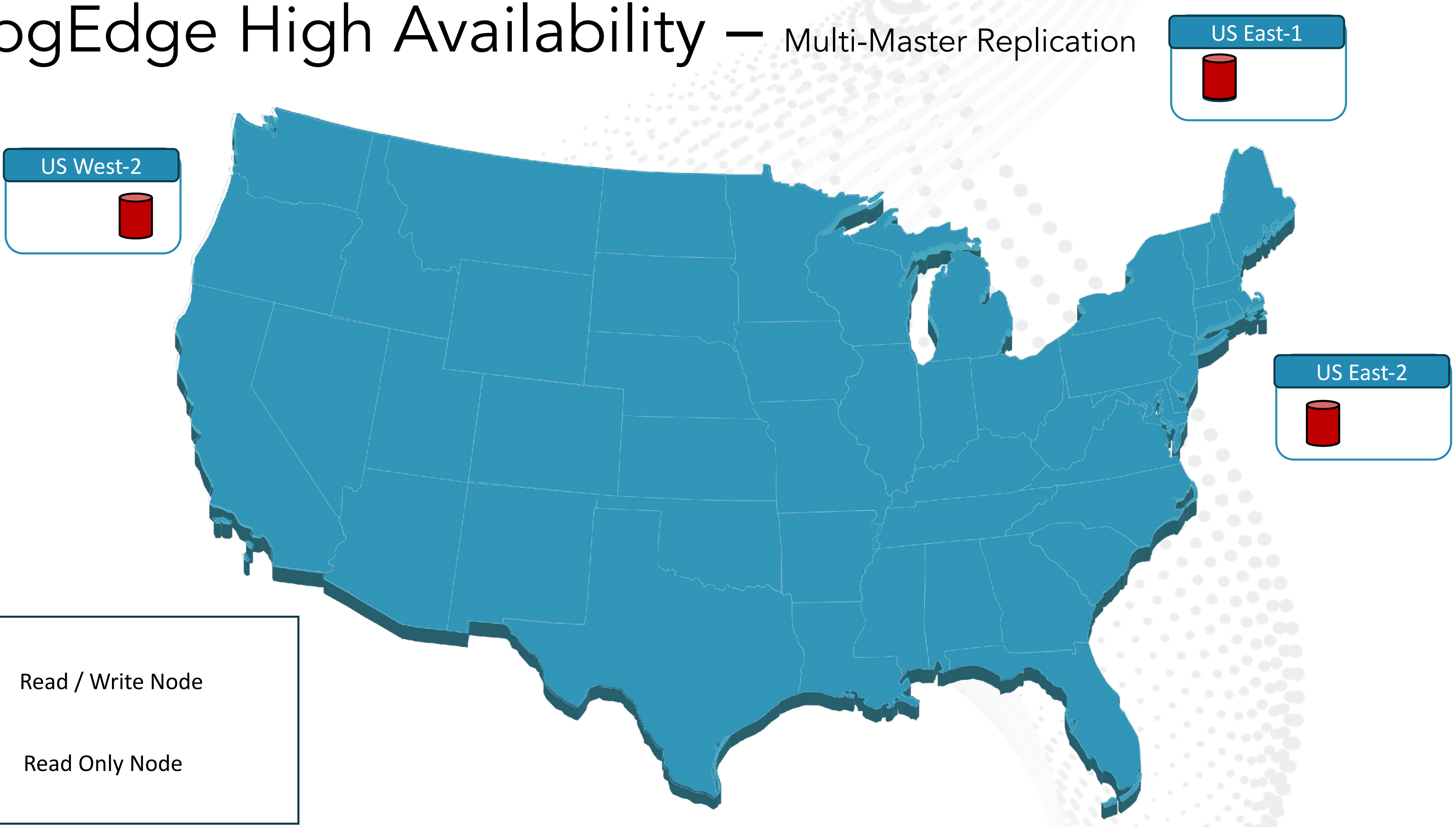
pgEdge High Availability Demo

Ultra High Availability & Zero Down Time

High Availability – Single Master Replication



pgEdge High Availability – Multi-Master Replication



🟥 Read / Write Node

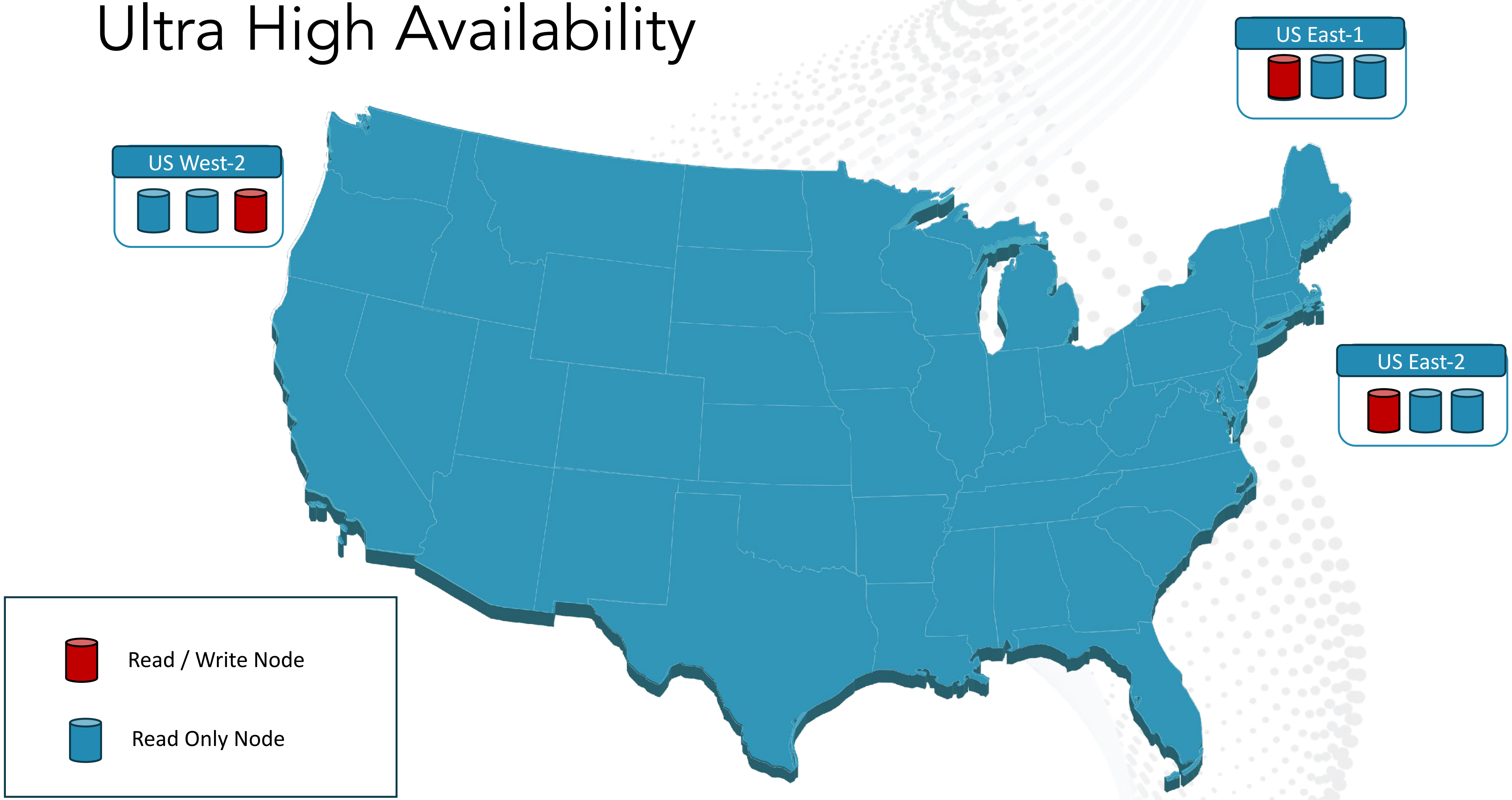
🟦 Read Only Node

High Availability Demo by

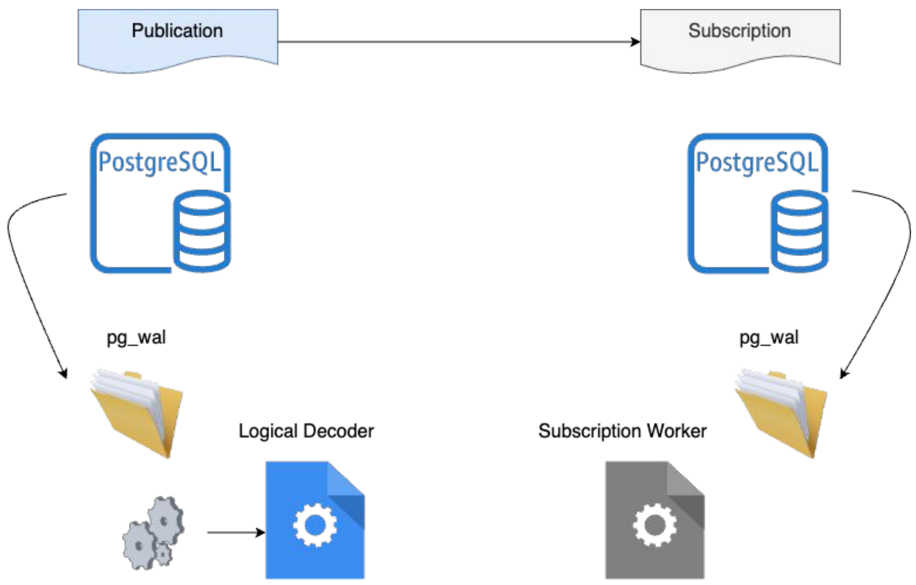
Director Solution Engineer

Mike Josephson

Ultra High Availability

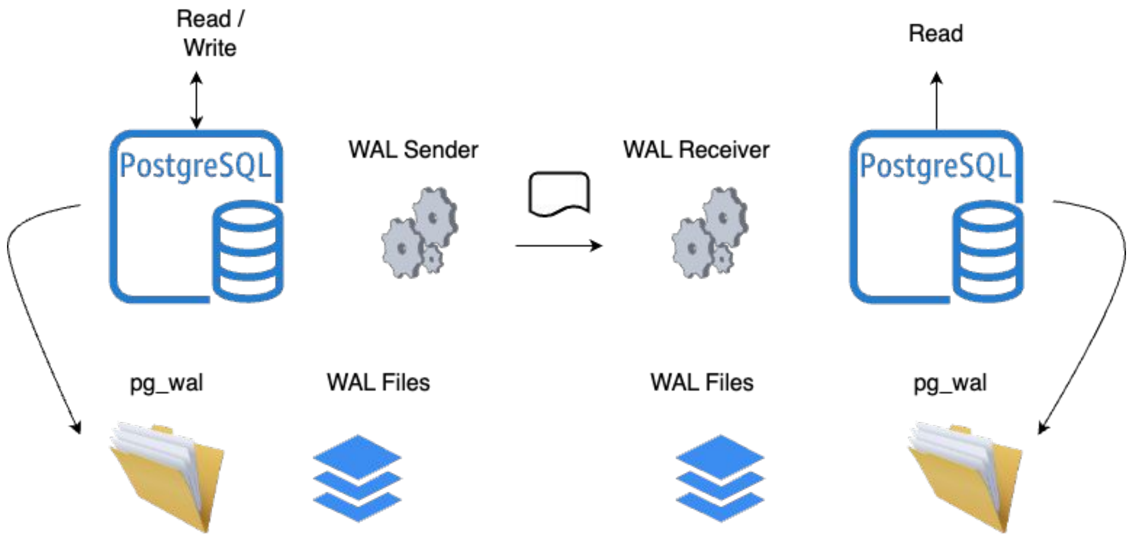


Logical Replication



- Logical replication is method of copying data objects and changes based on replication identity.
- Provides fine grained control over data replication and security.
- Publisher / Subscriber model - one or more subscriber subscribe to one or more publisher.
- Copy data in format that can be interpreted by other systems using logical decoding plugins.
- Publication is set of changes generated from a table or group of tables.
- Subscription is the downstream end of logical replication.

Physical Replication

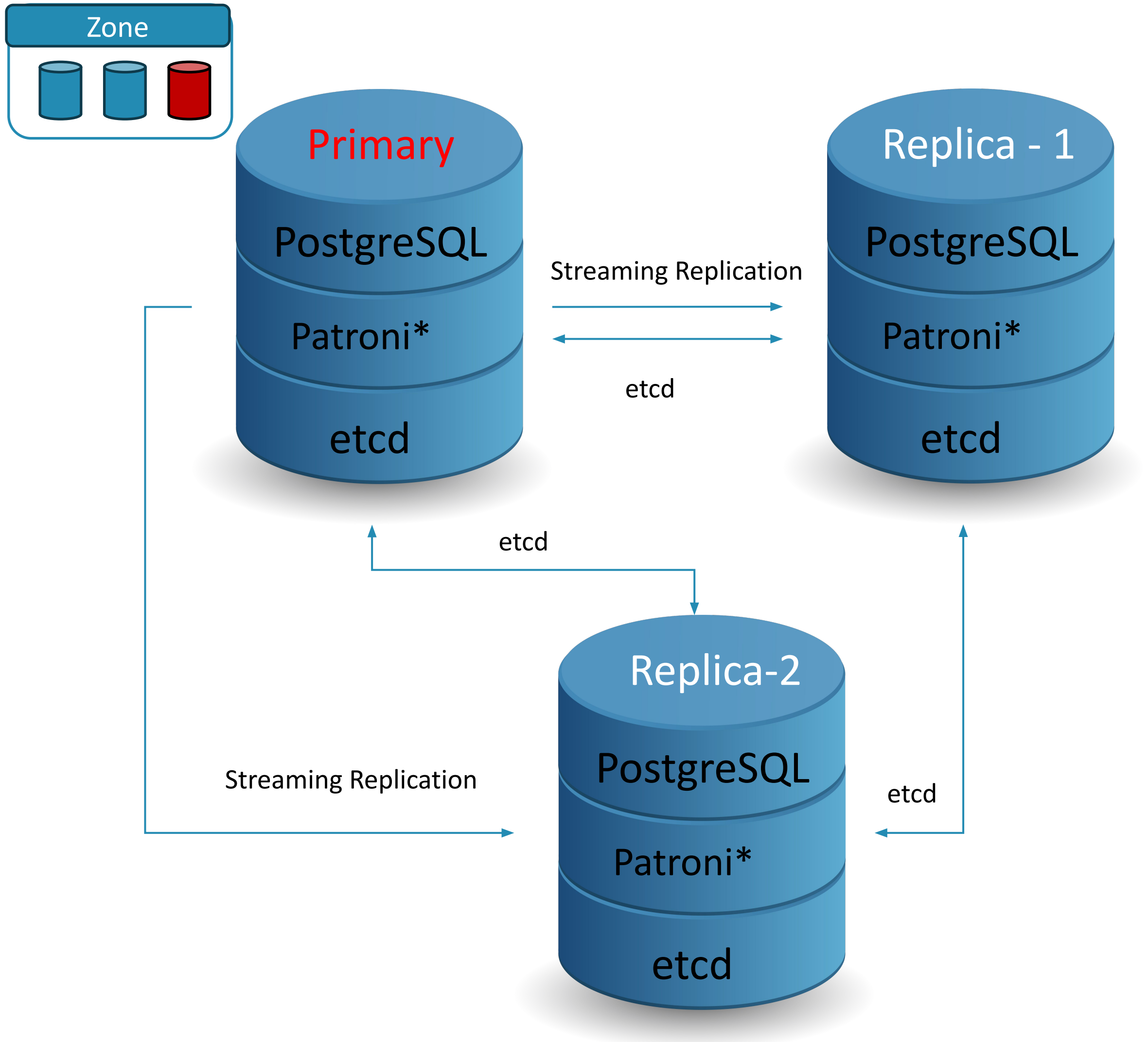


- Physical replication in PostgreSQL is a method for copying and synchronizing data from a primary server to standby servers in real-time.
- Real-time transfer of WAL records from a primary to standby servers to ensure data consistency and up-to-date replicas.
- Standby servers can run in hot standby mode, allowing them to handle read-only queries while replicating changes.
- Configurable as either synchronous, for strict data integrity, or asynchronous, for improved write performance.
- Facilitates automatic failover by promoting a standby to primary in case of primary server failure.

Physical vs Logical Replication

Aspect	Physical Replication	Logical Replication
Data Replication Level	Replicates data blocks on disk.	Replicates logical changes to data.
Granularity	Operates at the page level.	Operates at the row or transaction level.
Use Cases	High availability, disaster recovery.	Data distribution, selective replication, data warehousing.
Setup Complexity	Typically more complex.	Generally easier to set up and manage.
Compatibility	Requires same PostgreSQL version and often the same platform.	Can replicate data between different PostgreSQL versions or even different database systems.
Performance	Lower replication overhead, suitable for low-latency scenarios.	May have higher overhead due to additional processing.
Conflict Resolution	Typically used in master-slave setup.	Used for multi-master replication, may require conflict resolution mechanisms

Patroni-ETCD



etcd

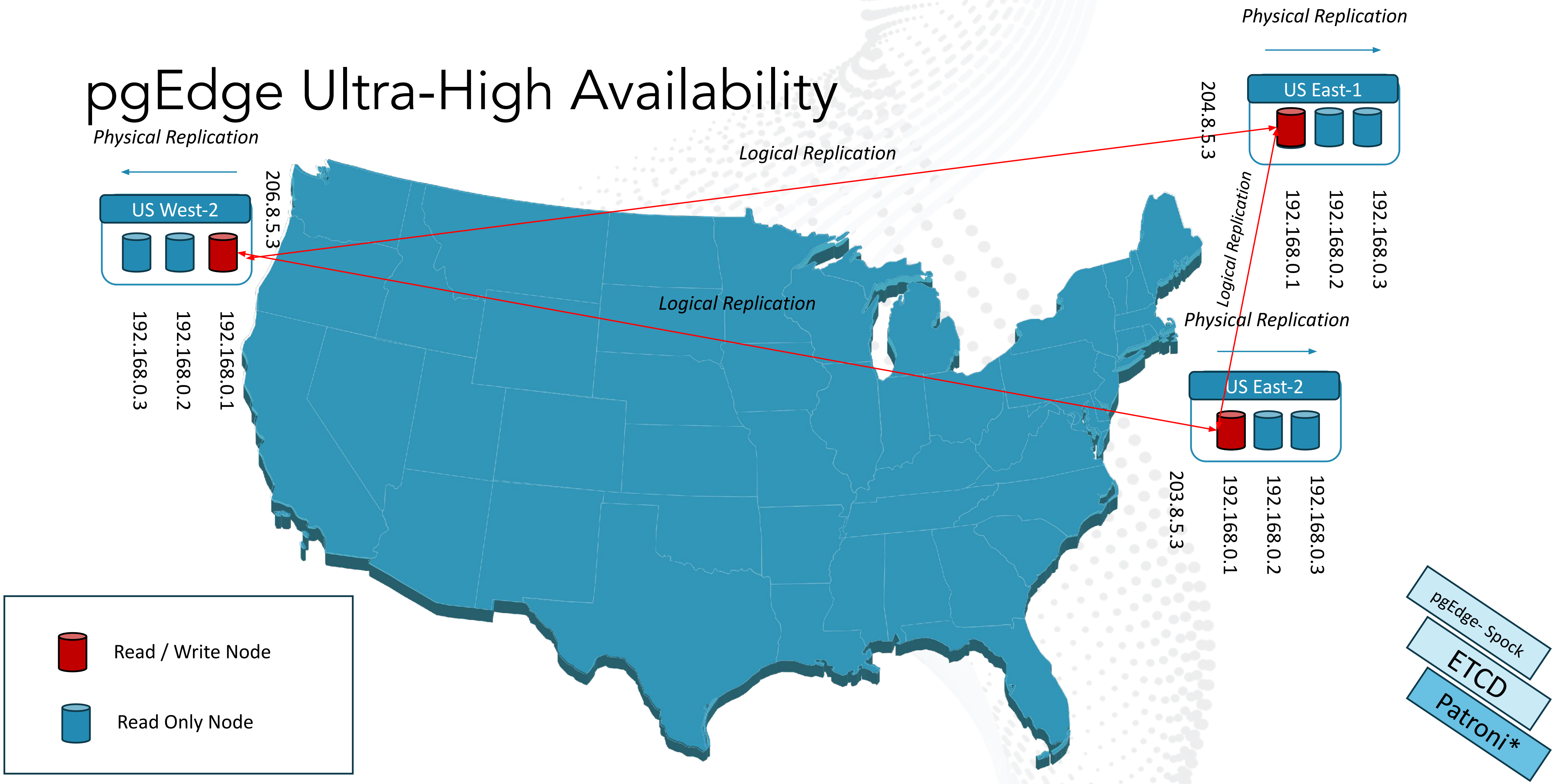
- etcd is designed to provide strong consistency and high availability, even in the face of network partitions, using the Raft consensus algorithm.
- It is often used for service discovery in clustered systems, where it manages and stores the metadata and configuration data for services running across various nodes

Patroni*

- Patroni automates the failover process, allowing for a standby server to be promoted to primary automatically in case the current primary fails.
- It allows changes to the PostgreSQL instance configuration in a dynamic way, without needing to restart the cluster.

** pgEdge has its own fork of Patroni compatible with multimaster replication solution*

pgEdge Ultra-High Availability



Q&A

In Summary …

- pgEdge Distributed PostgreSQL makes achieving high availability easier

- Want to learn more? Book a demo of pgEdge Cloud at https://pages.pgedge.com/schedule-demo

- Download a copy of pgEdge Platform at www.pgedge.com/download

- Or to speak with a Solution Architect contact us at www.pgedge.com/contact

pgEdge

Thank You!