



SUMMIT  
ONLINE/2021

# POSTGRES SQL HIGH AVAILABILITY AND BACKUP



**Dimitri Bellini**  
Quadrata CEO



**QUADRATA**  
SERVICE GROUP s.r.l.

## *Questions from many customers*

- **Is my Zabbix DB architecture highly available?**
- **Does it solve my Zabbix slowness during housekeeping?**
- **Does it save my precious NVMe space?**
- **What about Backup or Disaster Recovery?**



## *Our final solution....*



PostgreSQL



etcd



Patroni

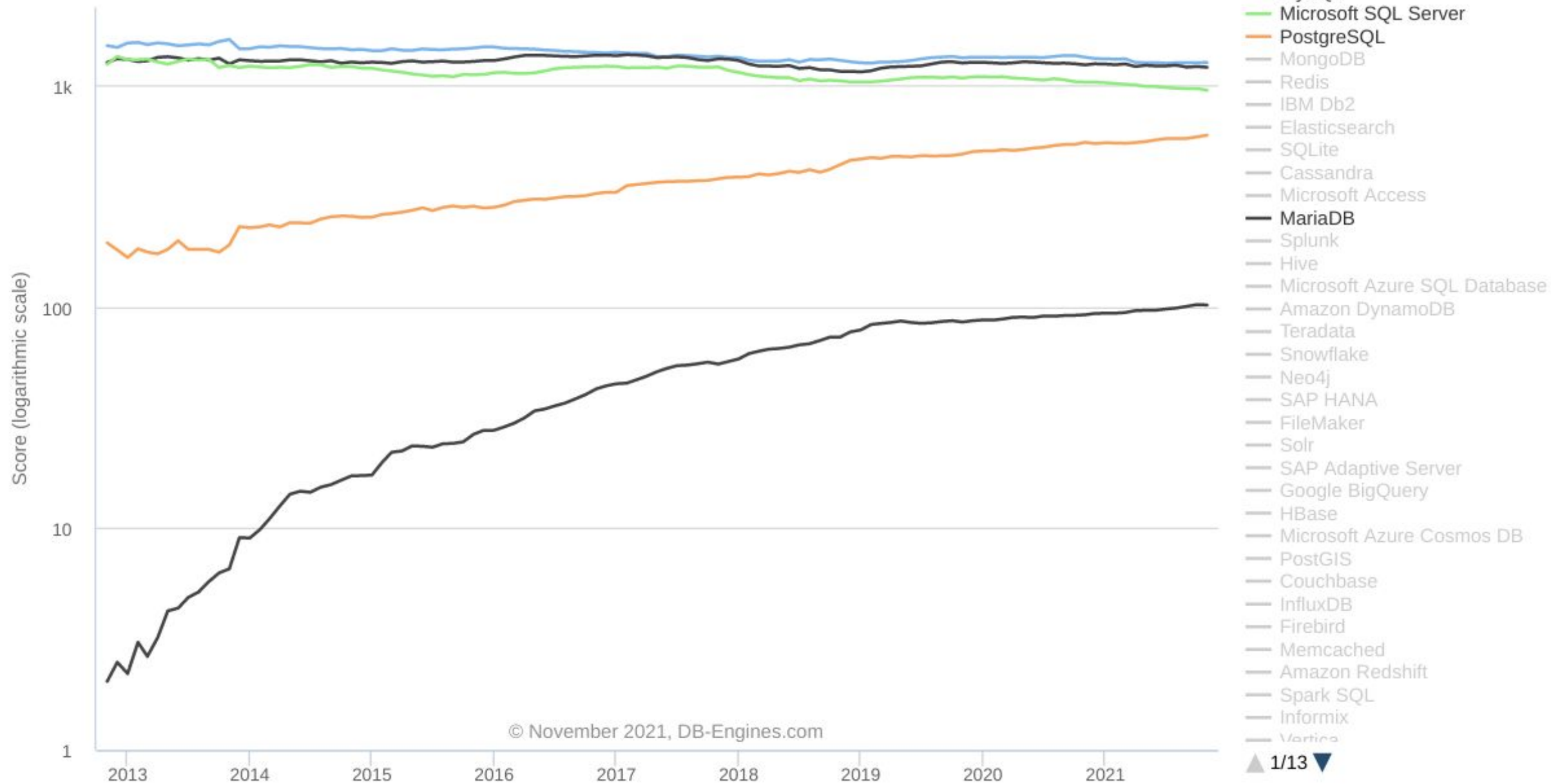


TimescaleDB

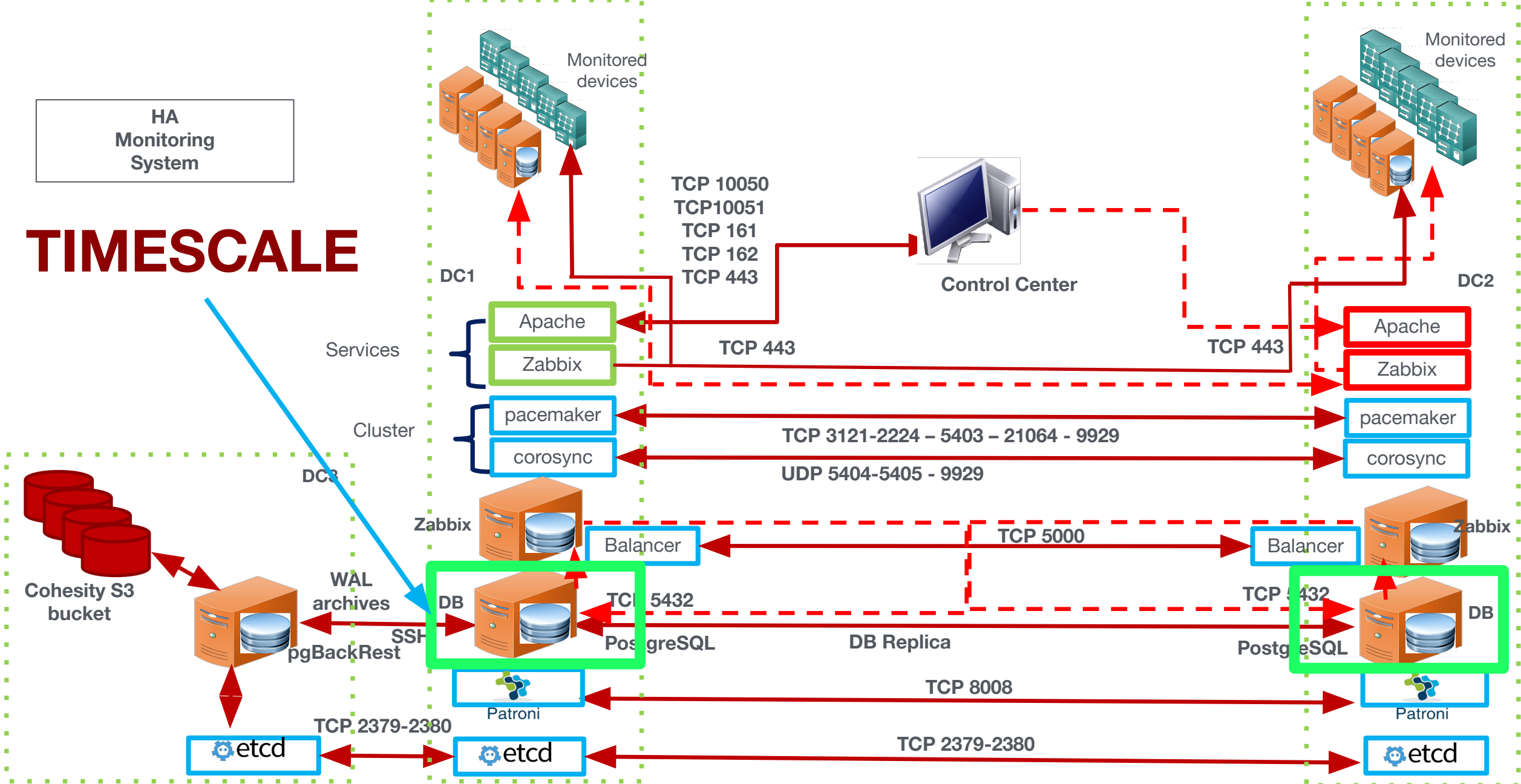


pgBackRest

# DB-Engines Ranking



# TIMESCALE



## Timescale & compression

**TimescaleDB** is a category-defining relational database for time-series data. Packaged as a PostgreSQL extension, TimescaleDB is designed to be easy to use, easy to get started, and easy to maintain.

<https://docs.timescale.com/latest/getting-started/installation/rhel-centos/installation-yum>

**Pro:** as Zabbix history and trends tables are now partitioned, the housekeeping will be a simple drop of a partition, so very fast

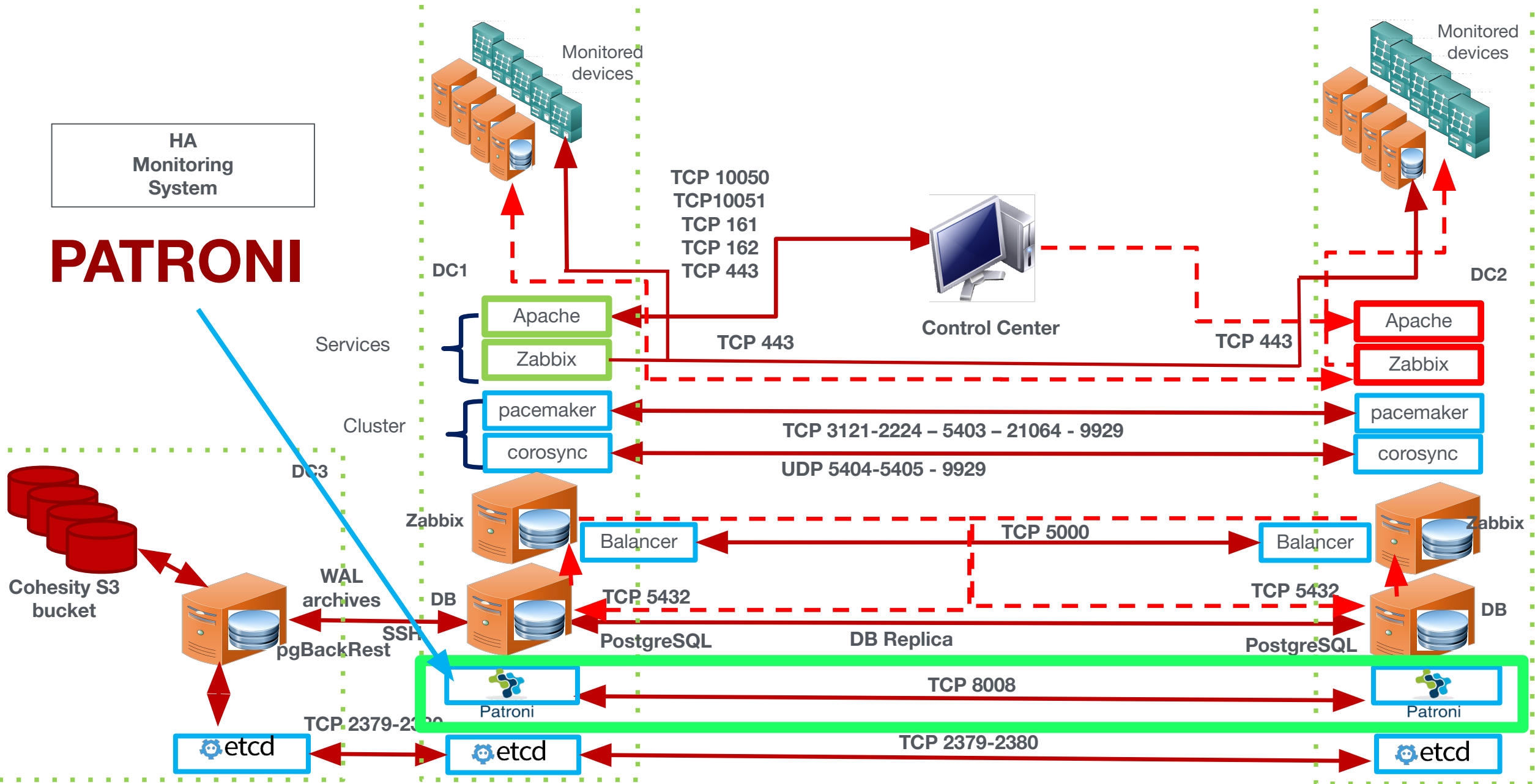
**Cons:** we need one and only one retention time for history and trends

**Compression:** Since Zabbix 5.0 we can enable data compression for History/Trends tables, we can save 50-70% of DB space. Using a well-sized CPU there is no performance degradation.



HA Monitoring System

# PATRONI



# PATRONI

**Patroni** is a tool to create, manage, maintain and monitor highly available PostgreSQL cluster setups using streaming replication.

**Patroni** creates the cluster, initiates streaming replication, handles synchronicity requirements, monitors liveness of primary and replica, can change the configuration of all cluster members, issues reload commands and restarts selected cluster members, handles planned switchovers and unplanned failovers, rewinds a failed primary to bring it back in line and reinitiates all replication connections to point to the newly promoted primary.

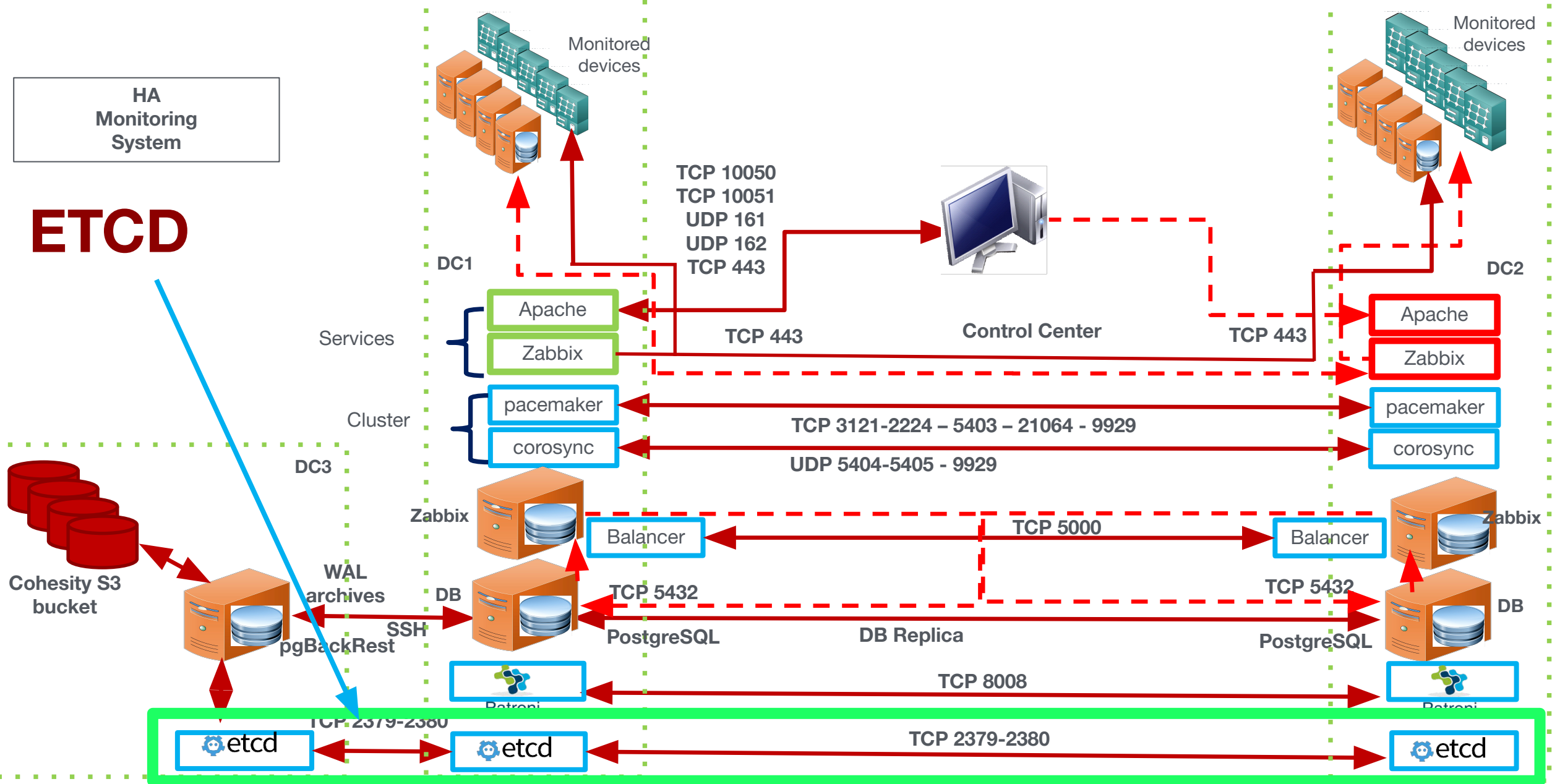
All of the data that Patroni collects is mirrored in a distributed key-value store, and based on the information present in the store, all Patroni instances agree on decisions, such as which replica to promote if the primary has failed.

**The Patroni configuration for a specific PostgreSQL instance is a single YAML file**



HA  
Monitoring  
System

ETCD



## ETCD

A DCS (distributed consensus store) based on RAFT protocol that runs across all computers in a cluster and provides a dynamic configuration registry, allowing various configuration data to be easily and reliably shared between the cluster members. The ETCD distributed key-value store enables atomic manipulation of keys and values. This forwards the difficult problem of cluster consensus (which is critical to avoid the split-brain scenario) to battle tested components, proven to work correctly even under the worst circumstances.

<https://computingforgeeks.com/how-to-install-etcd-on-rhel-centos-8/>

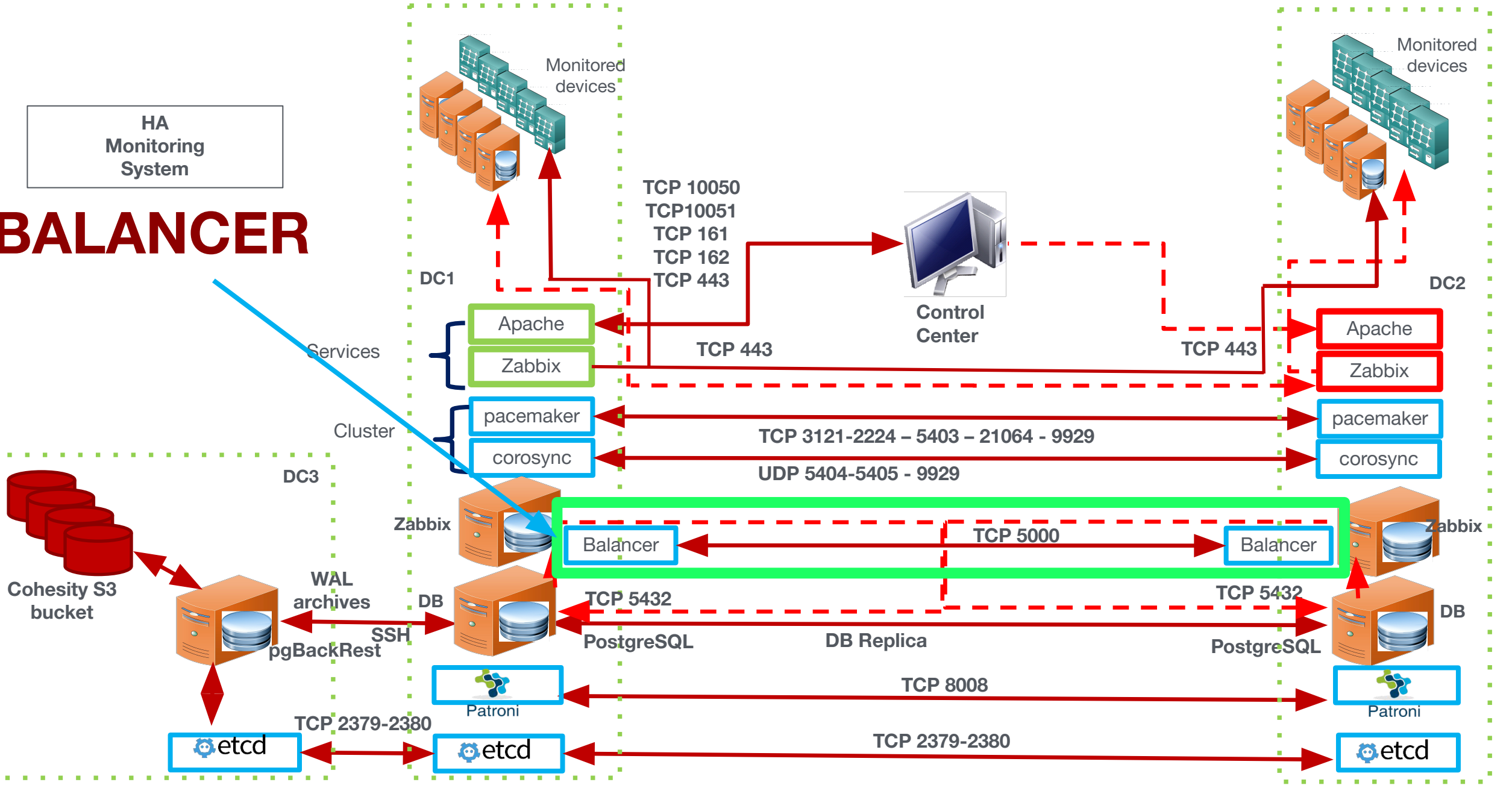
**Pro:** easy to setup

**Cons:** very sensitive to IO load...



HA  
Monitoring  
System

# BALANCER

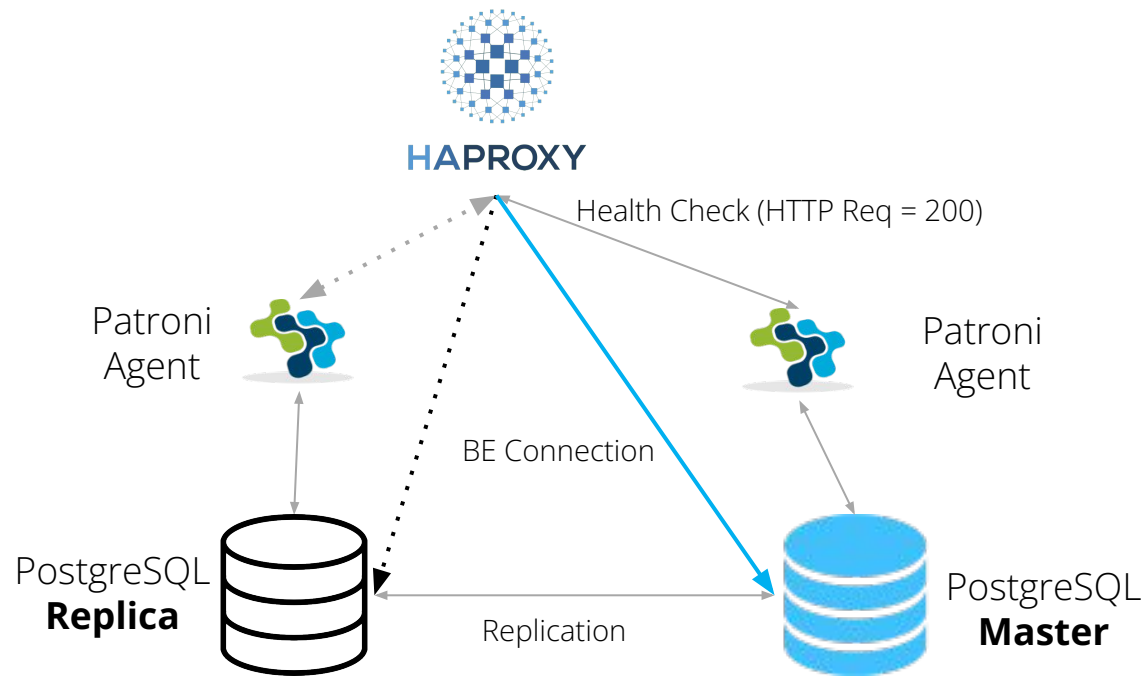




## HA BALANCER (Database TCP connections)

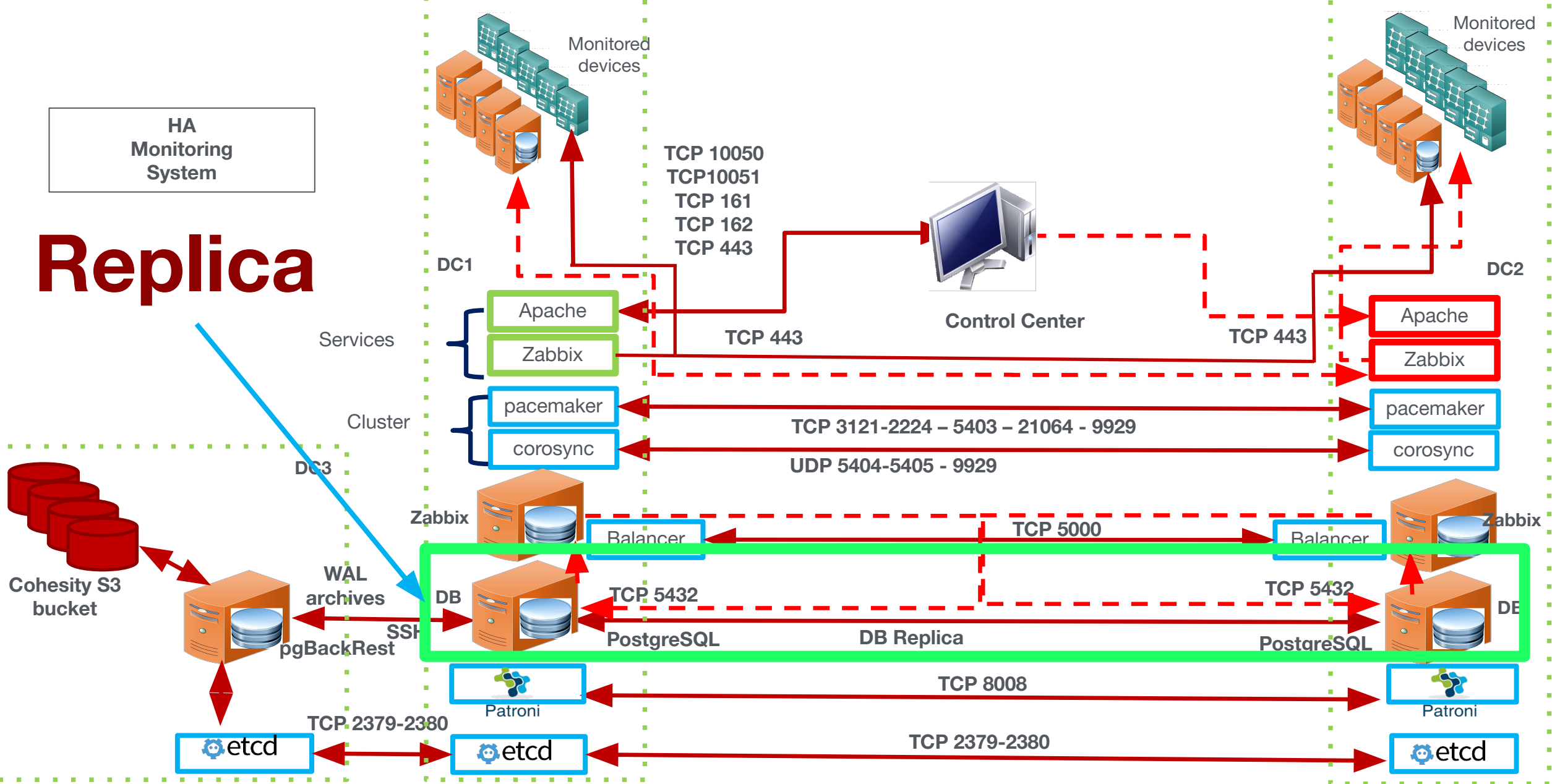
The balancer needs to know which PostgreSQL instance is active, the check is implemented using a RestAPI call to the Patroni Endpoint (default port 8008/TCP )

The Balancer could be software (i.e. HAProxy) or hardware (i.e. F5)



HA  
Monitoring  
System

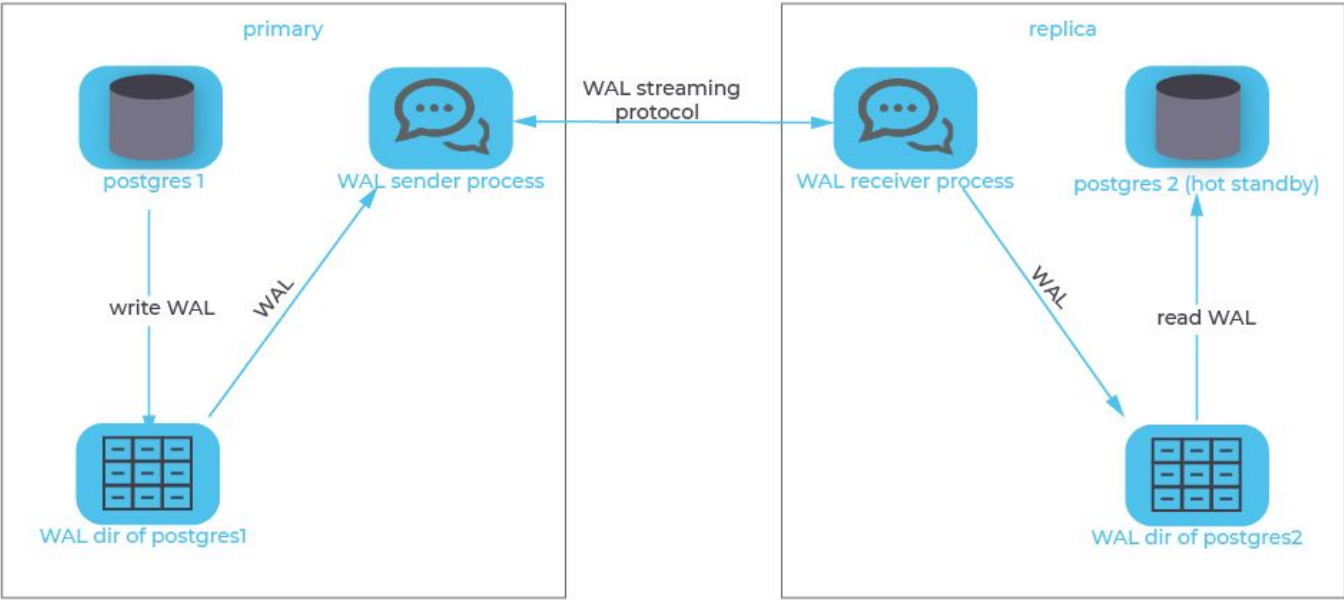
# Replica



# Postgres Replica

A replica is a secondary copy of your data (the primary copy), i.e. of your databases. In the event of a failure, it is possible to continue business operations on the replica. During normal operations, both the primary and the secondary copy can be read from to boost performance.

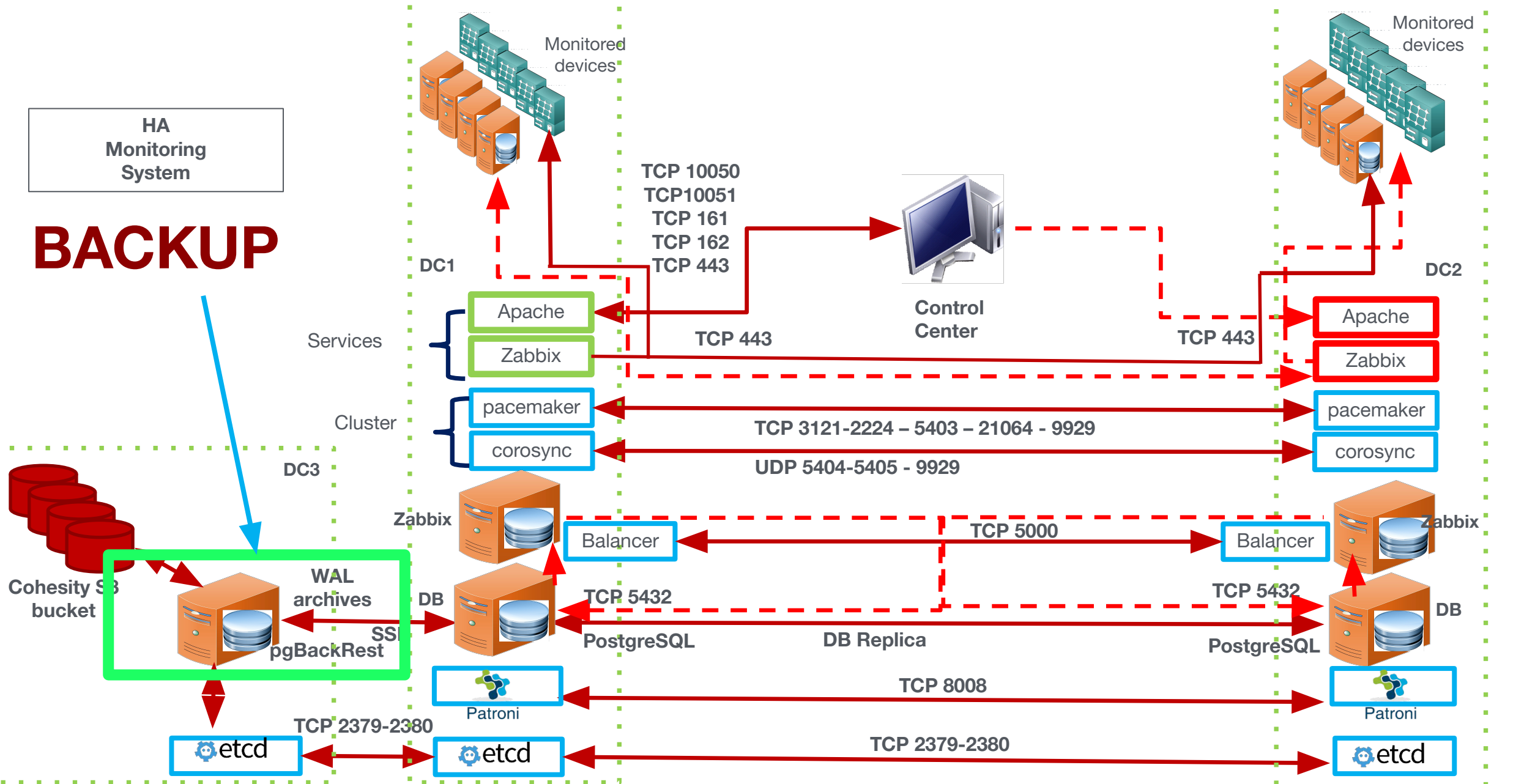
In order to achieve very low RPO (recovery point objective), PostgreSQL uses streaming replication (Write Ahead Log), where single transaction log messages are reproduced to the replica in asynchronous mode (hot-standby)





HA  
Monitoring  
System

# BACKUP



# Postgres backup using pgBackRest tool (<https://pgbackrest.org/> )

**pgBackRest** aims to be a reliable, easy-to-use backup and restore solution that can seamlessly scale up to the largest databases and workloads by utilizing algorithms that are optimized for database-specific requirements.

A custom protocol allows pgBackRest to backup, restore, and archive locally or remotely via SSH with minimal configuration. We strongly advise on a external S3 repository for more rock solid implementation.

## Pro:

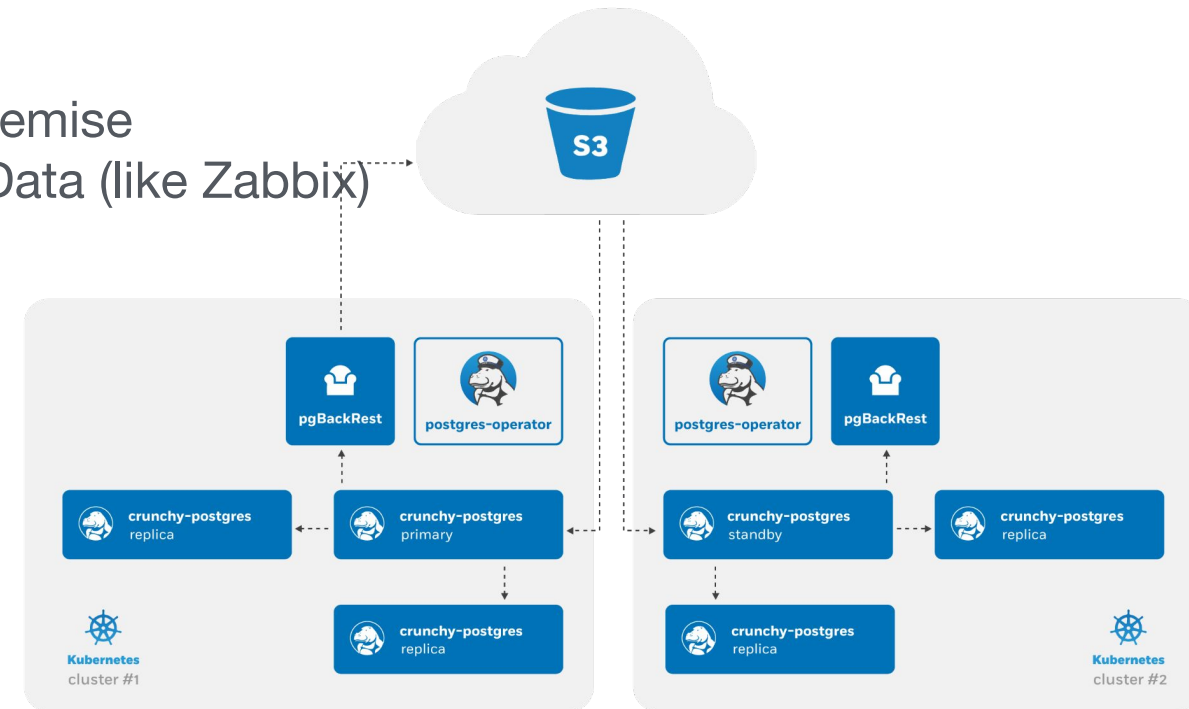
- Highly customizable to suite many user scenarios like DR or classic Backup
- Many storage options (filesystem,CIFS,NFS,S3,Azure Blob,GCS)

## Cons:

- None

# KUBERNETES - PostgreSQL Operators

- **Patroni PostgreSQL Operator** (<https://github.com/zalando/postgres-operator>)
  - Poor documentation
  - Most of the work is based on Amazon AWS
- **CrunchyData PostgreSQL Operator**  
(<https://access.crunchydata.com/documentation/postgres-operator/v5/>)
  - Very well documented
  - Declarative Postgres solution
  - Many Cloud solution are supported but also Onpremise
  - Opensource but Officially Supported by CrunchyData (like Zabbix)
  - Backup integration with pgBackRest







SUMMIT  
ONLINE / 2021

# Thanks!!!



**QUADRATA**  
SERVICE GROUP s.r.l.