CloudNativePG

An open source operator for PostgreSQL

Gabriele Bartolini, VP/CTO Cloud Native at EDB

May 11, 2022 - Presentation to TAG storage

About myself

- PostgreSQL user (since ~2000)
  - Community member since 2006
  - Co-founder of PostgreSQL Europe
- 2ndQuadrant (2008 to 2020)
  - Co-founder
  - Head of Global Support
  - Cloud Native Initiative Lead
- VP of Cloud Native at EDB (2020 to date)
- Co-founder of Barman
- Member of DoK
- DevOps evangelist

Setting the context

About EDB and our long history with PostgreSQL

- EDB is the largest contributor to the open source PostgreSQL database
- In 2020 EDB acquired 2ndQuadrant, major contributors with EDB for:
  - Crash recovery
  - Continuous backup and Point-In-Time-Recovery
  - Physical replication, including synchronous, including Hot Standby (read replicas)
  - Logical replication
  - Partitioning
- History of 15+ years of professional support on PostgreSQL
- Kubernetes Certified Service Provider (2019)
- Data on Kubernetes Community Founding Sponsor (2020)

History of CloudNativePG

- **Aug 2019:** the Cloud Native journey at 2ndQuadrant starts
- **Sep 2019:** we started writing our own operator from scratch
  - No existing open source operator was using native Kubernetes technologies
    - Primarily imperative (we wanted declarative)
    - Required an external failover management tool (we wanted the Kubernetes API server)
- **Dec 2019:** 2ndQuadrant becomes KCSP
- **Sep 2020:** EDB acquires 2ndQuadrant
- **Feb 2021:** EDB launches Cloud Native PostgreSQL
- **Feb 2022:** EDB decides to open source Cloud Native PostgreSQL
- **Apr 2022: CloudNativePG** is **open sourced** and **submitted for CNCF Sandbox**
- **May 2022:** CloudNativePG is first presented to **TAG Storage**

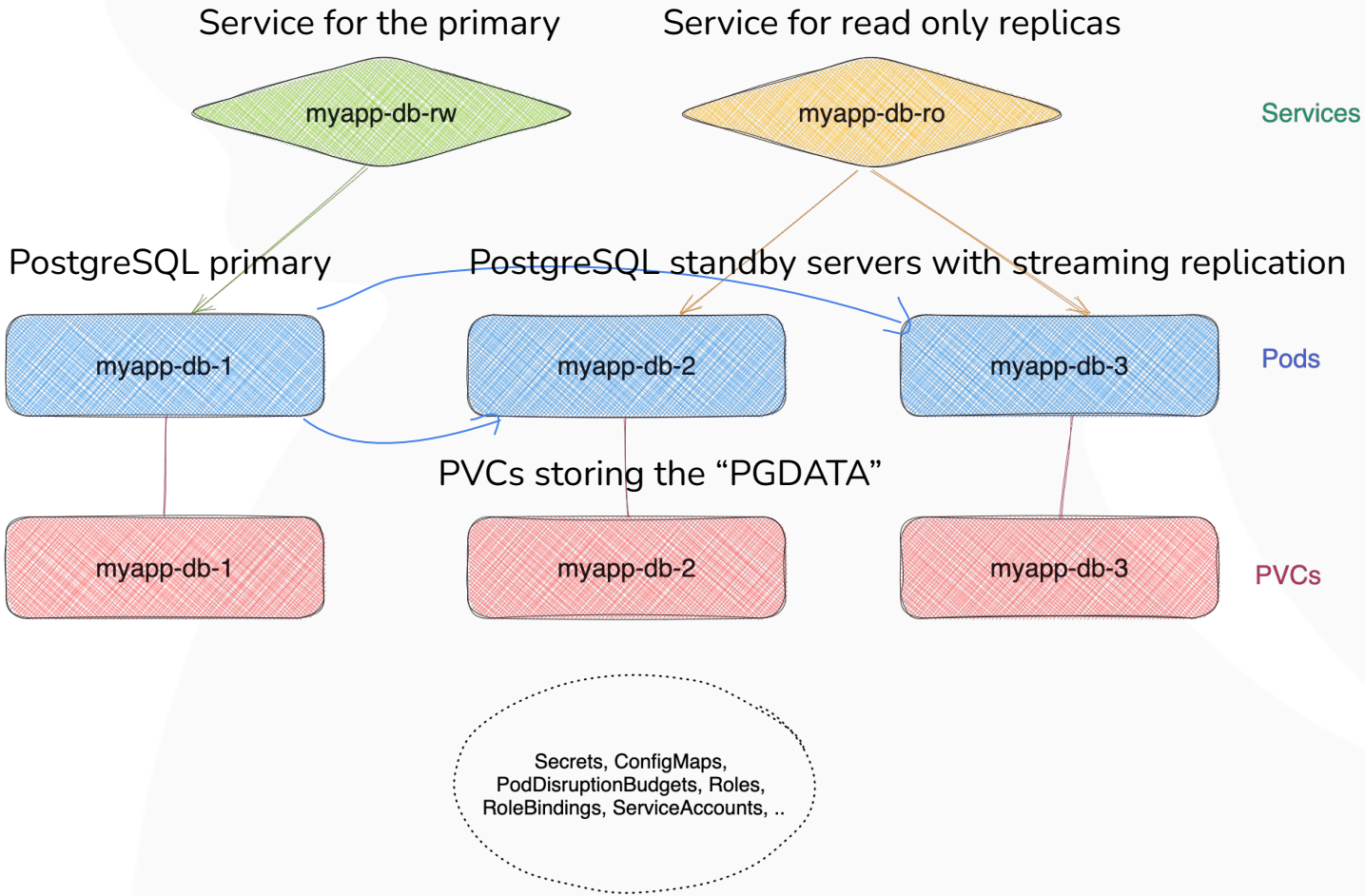
Why contributing CloudNativePG to CNCF?

- **Wicked problem:** database adoption in Kubernetes is lagging, and vice versa
  - A single vendor cannot solve this problem unilaterally
  - A holistic approach is required as it also involves cultural aspects
    - Promote usage of data workloads in Kubernetes (e.g. DoK Community)
- Lay the foundations of a vendor neutral and openly governed community
  - Centered around an open source project
  - Built on solid principles and values inspired by the CNCF
  - Attracting multiple vendors and users of PostgreSQL in the Kubernetes ecosystem
  - The home for Postgres in Kubernetes within CNCF
- Enhance the quality of PostgreSQL through feedback from this community
- Foster integration of PostgreSQL in the CNCF ecosystem

Bird’s eye view of CloudNativePG

Example

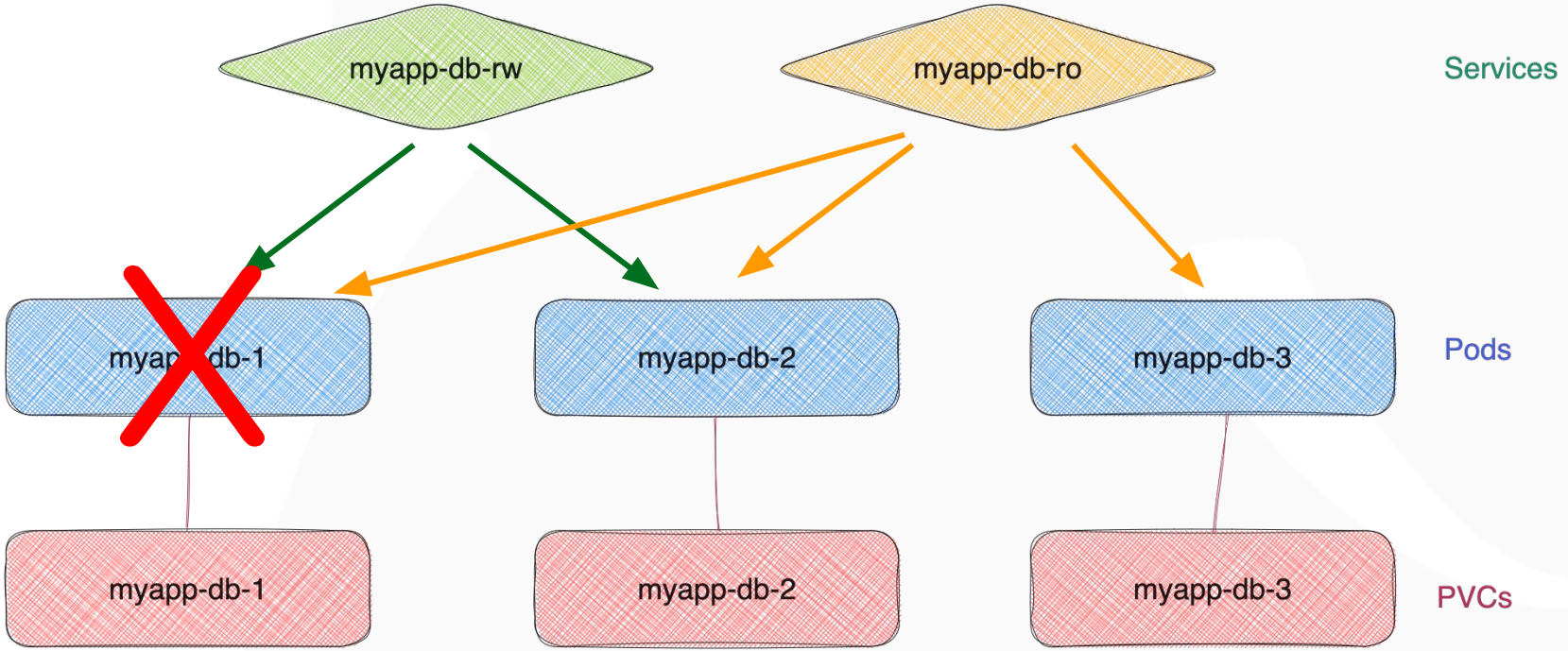
```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: myapp-db
spec:
  instances: 3
  imageName: ghcr.io/cloudnative-pg/postgresql:14.2

  storage:
    size: 10Gi
```


Service for the primary
Service for read only replicas
myapp-db-rw
myapp-db-ro
Services
PostgreSQL primary
PostgreSQL standby servers with streaming replication
myapp-db-1
myapp-db-2
myapp-db-3
Pods
PVCs storing the “PGDATA”
myapp-db-1
myapp-db-2
myapp-db-3
PVCs
Secrets, ConfigMaps,
PodDisruptionBudgets, Roles,
RoleBindings, ServiceAccounts, ..

Management of the PostgreSQL cluster

- Ensure there's always one (and only one) primary in the cluster
- Self-healing capabilities
  - Probes for liveness and readiness
  - Failover of the primary
  - Automated creation of a replica
  - Resync of a former primary
- Planned switchover of the primary
  - Promotion of a selected replica
- Rolling updates
- Scale up/down capabilities

Automated failover

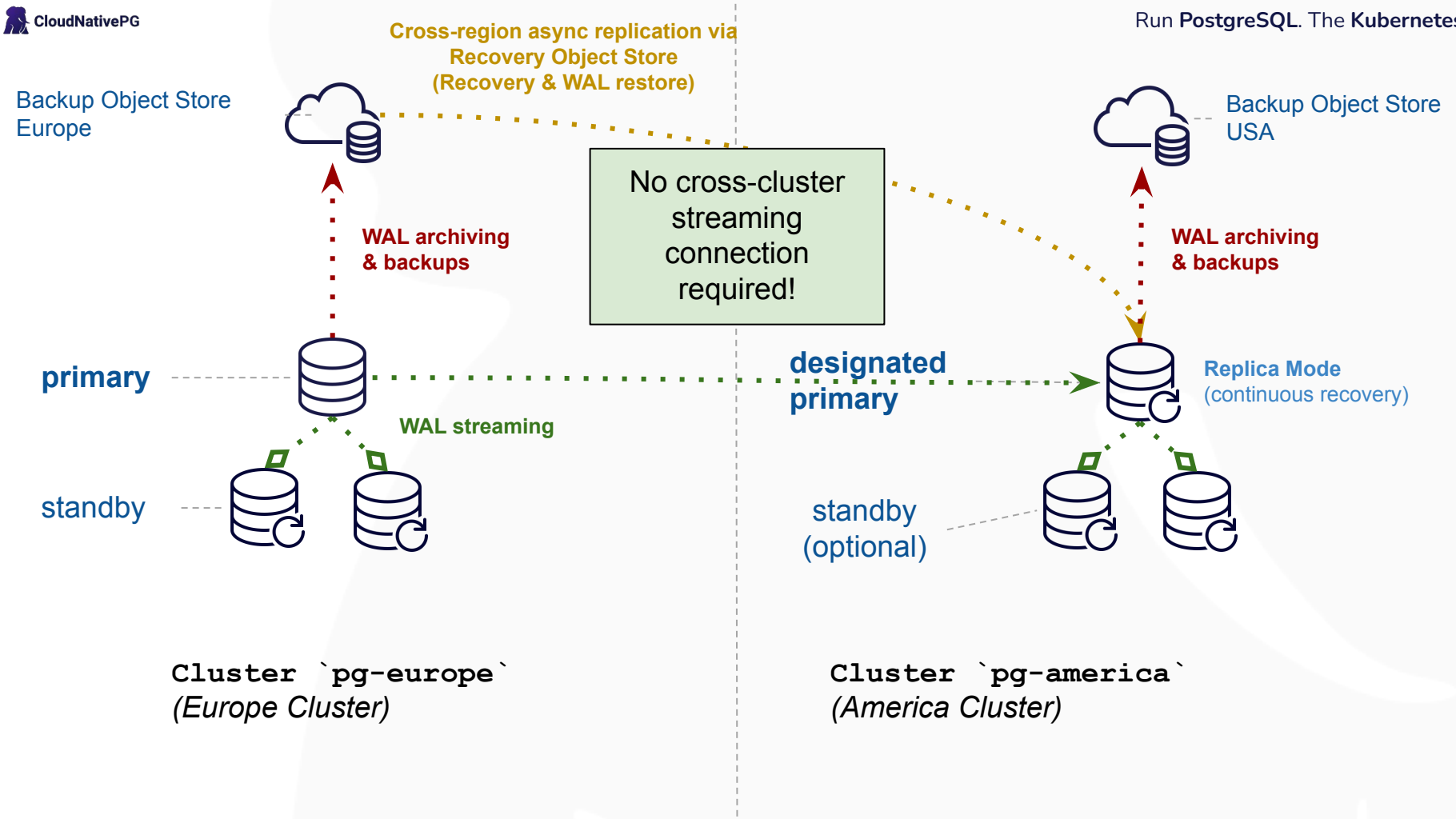


Storage management

- Storage is the most critical component for a database
- PGDATA is the most important asset in PostgreSQL
  - Mindset we adopted in CloudNativePG
  - The PVC storing PGDATA is central to the CloudNativePG operator
    - We have a saying: “The PGDATA PVC is worth a 1000 pods”
- Support for Persistent Volume Claims (PVC)
  - Automatic generation of PVC
  - Support for PVC templates
  - Reuse of storage for Pods in the same cluster
- Storage classes
- Freedom of choice
  - Local storage
  - Network storage

There's much more ...

- Rolling updates
- Security by default with PSP and TLS connections+authentication
- Continuous backup and Point-In-Time-Recovery (RPO < 5m)
- Compliance with Kubernetes resource management and pod scheduling
- Native Prometheus exporter
- JSON logging in standard output, including audit
- Automated pipelines in GitHub with E2E tests (currently > 100 specs)
- Multiple bootstrapping methods (empty or from another cluster/backup)
- Replica cluster


Cross-region async replication via
Recovery Object Store
(Recovery & WAL restore)
Backup Object Store
Europe
Backup Object Store
USA
No cross-cluster
streaming
connection
required!
WAL archiving
& backups
WAL archiving
& backups
primary
designated
primary
Replica Mode
(continuous recovery)
WAL streaming
standby
standby
(optional)
Cluster `pg-europe`
(Europe Cluster)
Cluster `pg-america`
(America Cluster)

How to contribute

The CloudNativePG Project

- CloudNativePG organization in GitHub: github.com/cloudnative-pg
  - CloudNativePG operator (main project): github.com/cloudnative-pg/cloudnative-pg
  - PostgreSQL operand images
  - Helm charts
- License: Apache 2.0
- IP: “The CloudNativePG Contributors”

Join us!

- We adopt the CNCF code of conduct
- Simple governance model based on maintainers for the initial phase
- Public roadmap using GitHub Projects beta
- Start from the CONTRIBUTING.md file
  - GitHub issues and discussions primarily
  - Slack channel
  - Participate to the biweekly developer meetings
- Special instructions for source code contributions
  - Work in progress
  - Setup of the dev environment
  - Setup of the test environment to run E2E tests with kind and k3d
  - DCO required

Originally created and sponsored by

Thank you!

github.com/cloudnative-pg

@CloudNativePg