EDB
Postgres for the AI Generation

Database and AI Platform: The Cloud-Native Way

Davide Tammaro, **Senior Sales Engineer**
September 13rd 2024

EDB Postgres in Kubernetes

Why Kubernetes

One of the benefits of Kubernetes is that it makes building and running complex applications much simpler.

Services, Load Balancing and Networking	Health Checking	Storage Management
Automated Scheduling	Scalability: Scale out/Scale down	Rolling Deployments

Enabling the same PostgreSQL everywhere

From self-managed to fully managed DBaaS in the Cloud

- Same applications
- Faster innovation
- Performance and scalability
- Stability, security and control
- Seamless integration

Private

Hybrid

Multi-cloud

Public

Bare Metal

Virtual Machines

Containers

A kubernetes operator for Postgres

Kubernetes adoption is rising and it is already the de facto standard orchestration tool

PostgreSQL clusters “management the kubernetes way” enables many cloud native usage patterns, e.g. spinning up, disposable clusters during tests, one cluster per microservice and one database per cluster

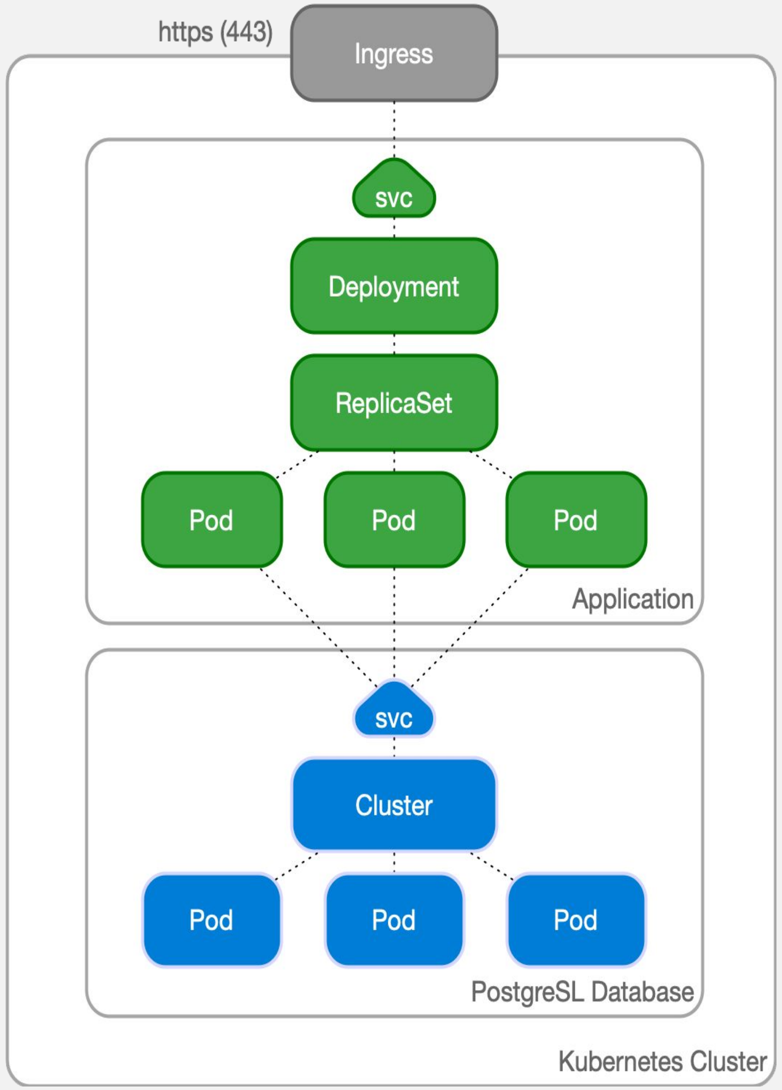
CNP tries to encode years of experience managing PostgreSQL clusters into an Operator which should automate all the known tasks a user could be willing to do

Our PostgreSQL operator must simulate the work of a DBA

Applications and Databases in K8s

- Stateless application as a deployment
  - Rolling upgrades
  - ReplicaSet for scaling and HA
  - Custom application images (Go, Django, Java, Python, C, C++, …)

- Stateful database using our operator
  - Embeds primary/standby logic
  - Service for RW and Read operations
  - Rolling upgrades, scaling, HA, …
  - “Cluster” CRD



Kubernetes timeline

- 2014, June: Google open sources Kubernetes
- 2015, July: Version 1.0 is released
- 2015, July: Google and Linux Foundation start the CNCF
- 2016, November: The operator pattern is introduced in a blog post
- 2018, August: The Community takes the lead
- 2019, April: Version 1.14 introduces **Local Persistent Volumes**
- 2019, August: EDB team starts the Kubernetes initiative
- 2020, June: we publish this blog about benchmarking local PVs on bare metal
- 2020, June: Data on Kubernetes Community founded
- 2021, February: EDB Cloud Native Postgres (CNP) 1.0 released
- 2022, May: **EDB donates CNP** and open sources it under CloudNativePG

CloudNativePG/EDB Postgres for Kubernetes

CloudNativePG

- Kubernetes operator for PostgreSQL
- "Level 5", Production ready
- Day 1 & 2 operations of a Postgres database
- Open source (May 2022)
    - Originally created by EDB
    - Apache License 2.0
    - Vendor neutral openly governed
    - 4300+ stars on GitHub
- Extends the K8s controller
    - Status of the `Cluster`
    - "no Patroni, No statefulsets"
- Immutable application containers
- Fully declarative

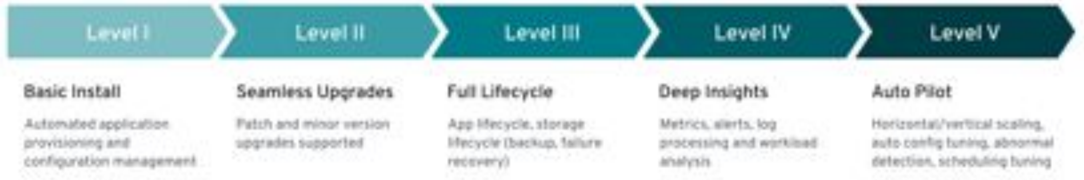
EDB Postgres for Kubernetes

- Fork of CloudNativePG

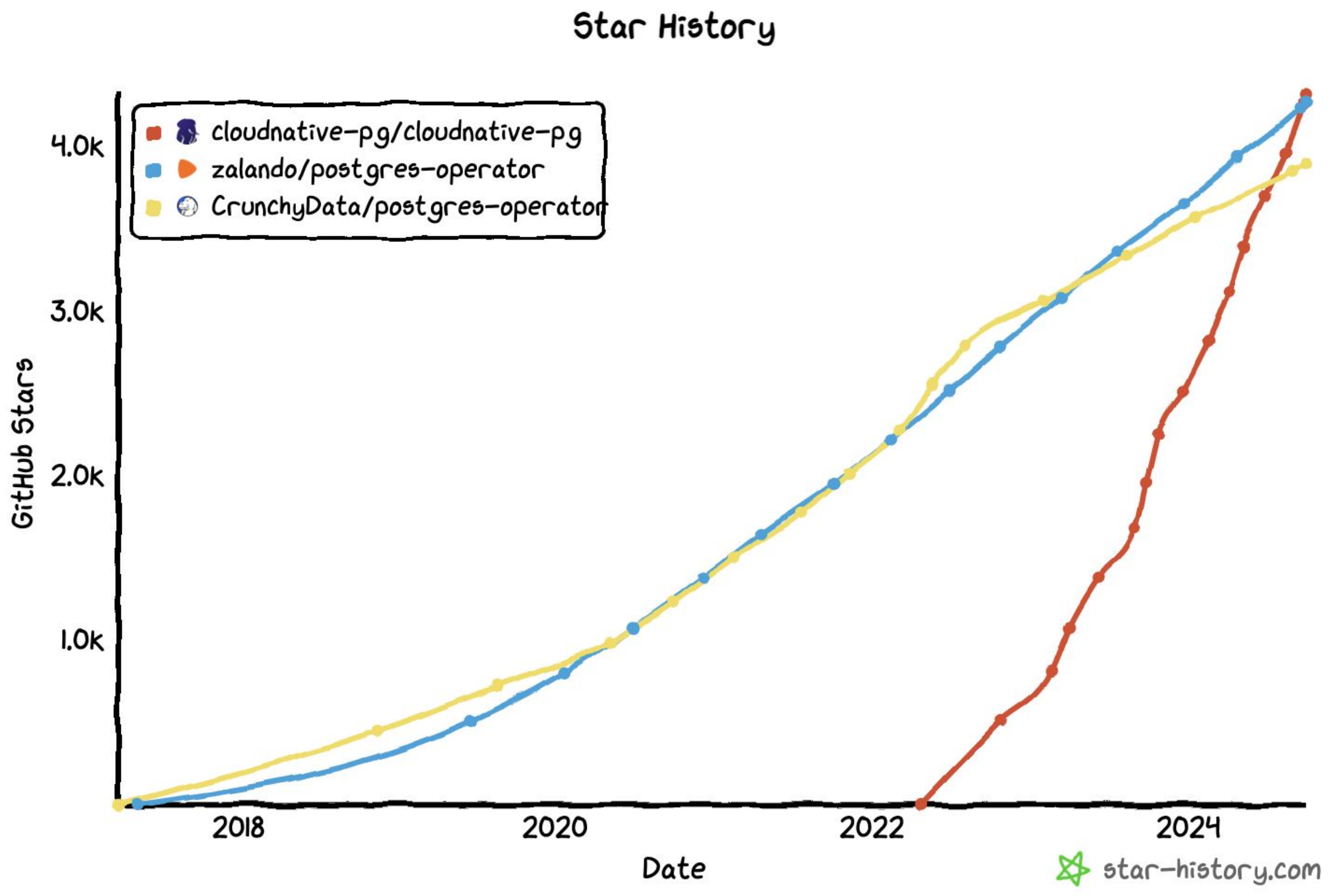
+

- Provides Long Term Support
- Access to EDB Postgres Extended (TDE)
- Access to EDB Postgres Advanced (TDE + Oracle Compatibility layer)
- Red Hat OpenShift compatibility
- Kubernetes level backup integration
    - Generic external backup interface

Operator Capabilities Levels


Level I
Basic Install
Automated application provisioning and configuration management
Level II
Seamless Upgrades
Patch and minor version upgrades supported
Level III
Full Lifecycle
App lifecycle, storage lifecycle (backup, failure recovery)
Level IV
Deep Insights
Metrics, alerts, log processing and workload analysis
Level V
Auto Pilot
Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

Github stars

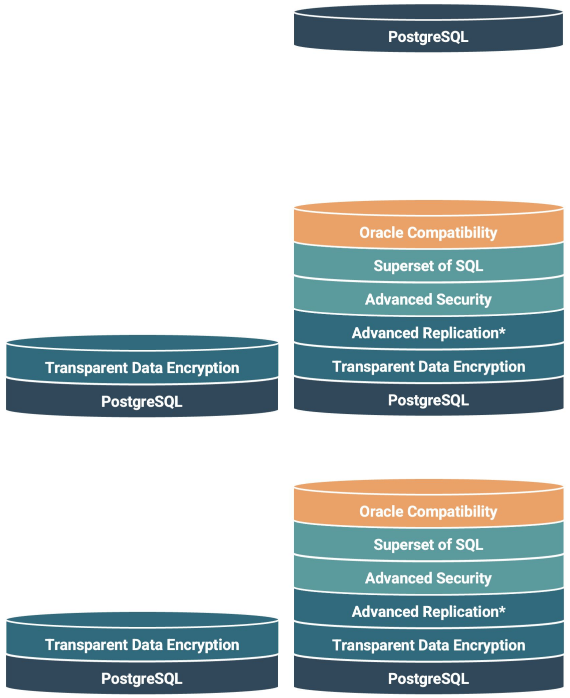


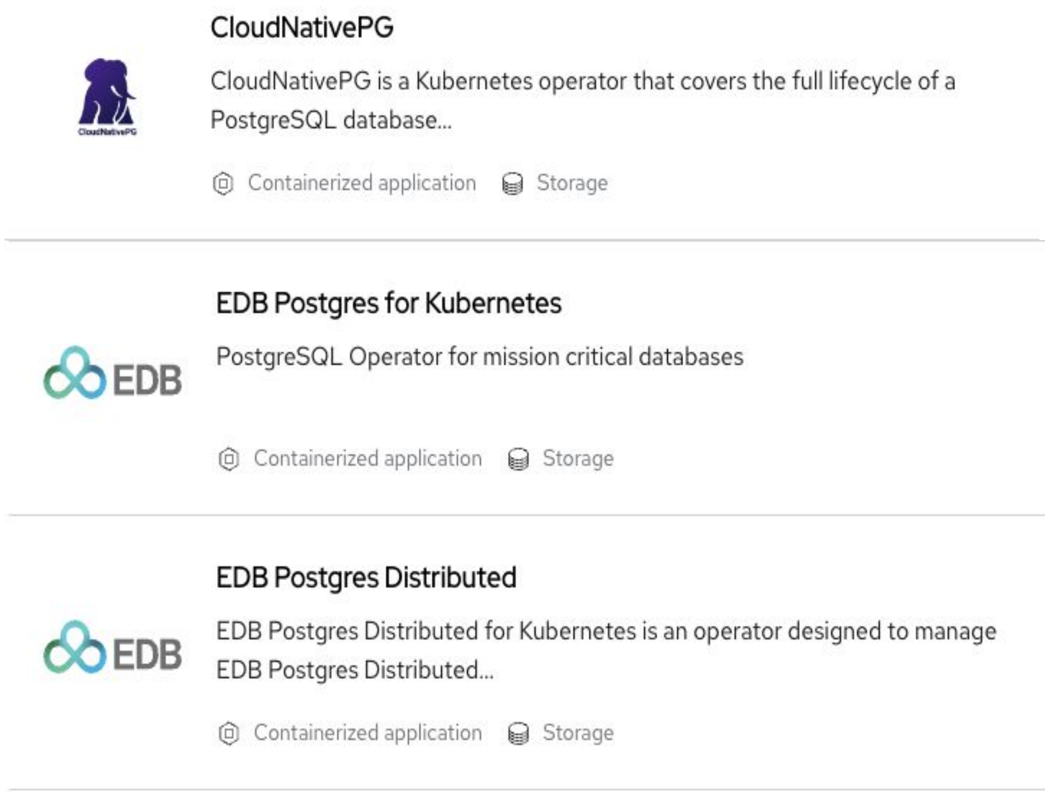
https://star-history.com/#cloudnative-pg/cloudnative-pg&zalando/postgres-operator&CrunchyData/postgres-operator&Date

https://github.com/cloudnative-pg/cloudnative-pg

Features

- Automated failover
- Services for RW and RO workloads
- Affinity control
- Backup and Recovery
- Rolling updates
- Scale up/down of read replicas
- Fencing and hibernation
- Native Prometheus exporters
- Log in JSON format to stdout
- OpenShift (and other K8S platforms) compatibility
- TDE (in EDB Postgres for Kubernetes)
- … and much more




CloudNativePG

CloudNativePG is a Kubernetes operator that covers the full lifecycle of a PostgreSQL database...

Containerized application Storage

EDB Postgres for Kubernetes

PostgreSQL Operator for mission critical databases

Containerized application Storage

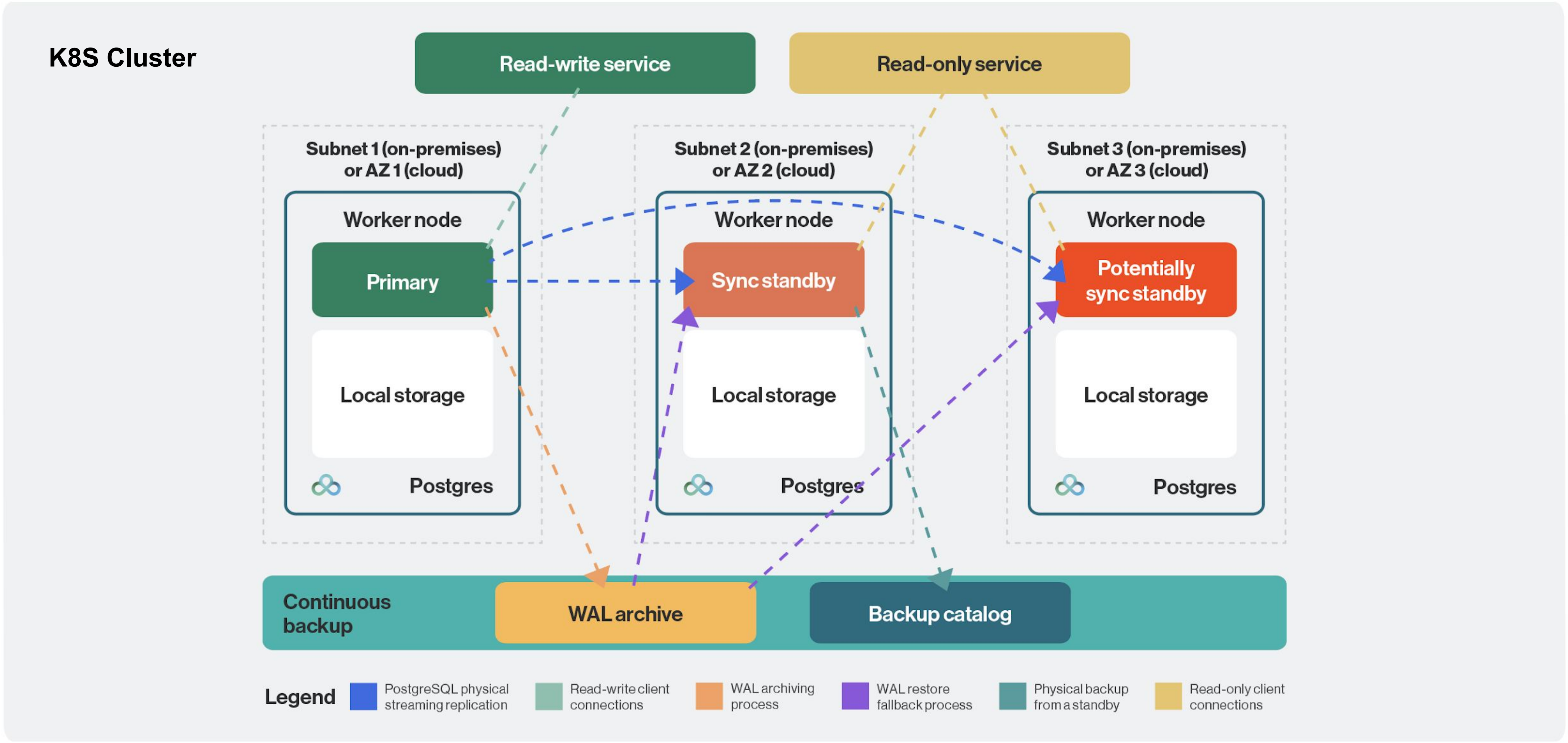
EDB Postgres Distributed

EDB Postgres Distributed for Kubernetes is an operator designed to manage EDB Postgres Distributed...

Containerized application Storage

Architectures

Production Cluster with 3+ Availability Zones


K8S Cluster
Read-write service
Read-only service
Subnet 1 (on-premises) or AZ 1 (cloud)
Subnet 2 (on-premises) or AZ 2 (cloud)
Subnet 3 (on-premises) or AZ 3 (cloud)
Worker node
Primary
Local storage
Postgres
Worker node
Sync standby
Local storage
Postgres
Worker node
Potentially sync standby
Local storage
Postgres
Continuous backup
WAL archive
Backup catalog
Legend
PostgreSQL physical streaming replication
Read-write client connections
WAL archiving process
WAL restore fallback process
Physical backup from a standby
Read-only client connections

Configuration example

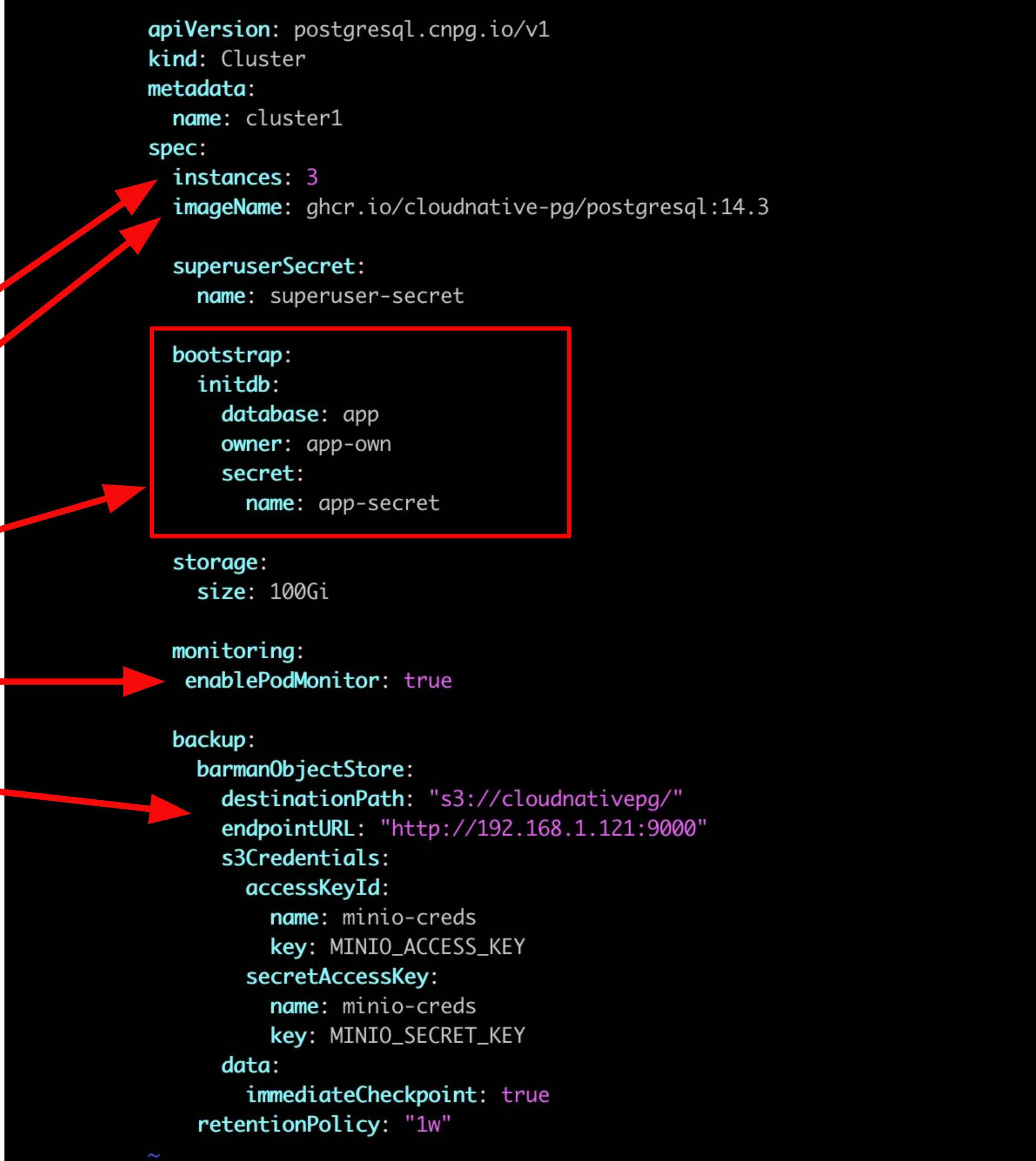
Number of instances in streaming replica

Postgres version

Initialize database (new)

Monitoring to prometheus

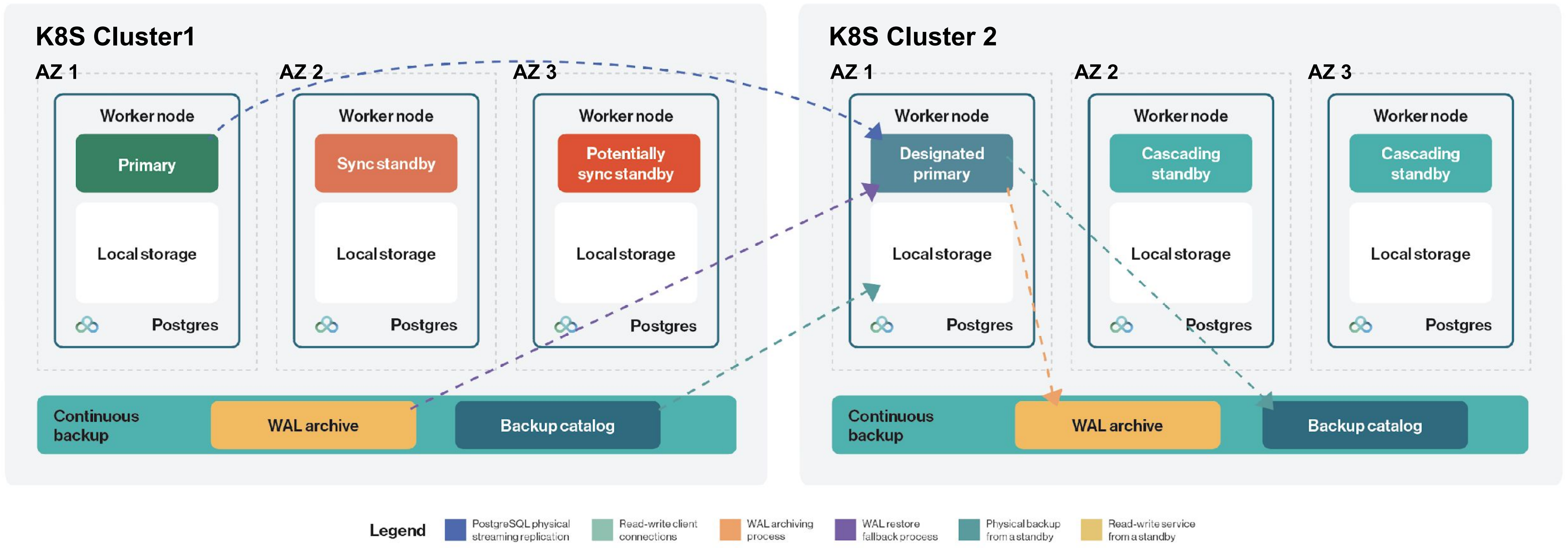
Barman backup repository



```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: cluster1
spec:
  instances: 3
  imageName: ghcr.io/cloudnative-pg/postgresql:14.3

  superuserSecret:
    name: superuser-secret

  bootstrap:
    initdb:
      database: app
      owner: app-own
      secret:
        name: app-secret

  storage:
    size: 100Gi

  monitoring:
   enablePodMonitor: true

  backup:
    barmanObjectStore:
      destinationPath: "s3://cloudnativepg/"
      endpointURL: "http://192.168.1.121:9000"
      s3Credentials:
        accessKeyId:
          name: minio-creds
          key: MINIO_ACCESS_KEY
        secretAccessKey:
          name: minio-creds
          key: MINIO_SECRET_KEY
      data:
        immediateCheckpoint: true
    retentionPolicy: "1w"
```

Symmetric Architecture on 2 different K8S clusters

Primary + DR, RPO=0, RTO=manual, 100% declarative configuration



Configuration example

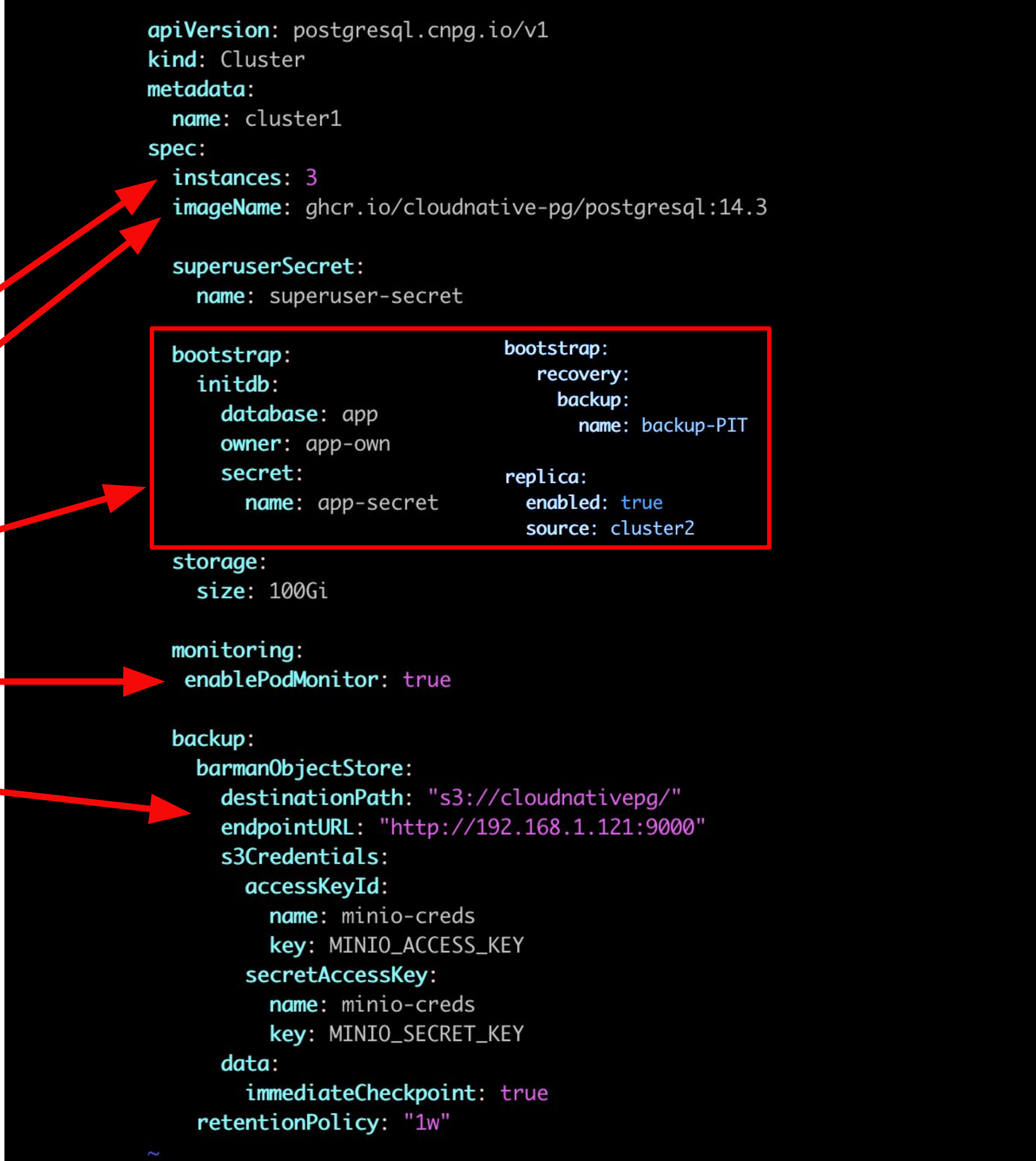
Number of instances in streaming replica

Postgres version

Initialize database (as replica from)

Monitoring to prometheus

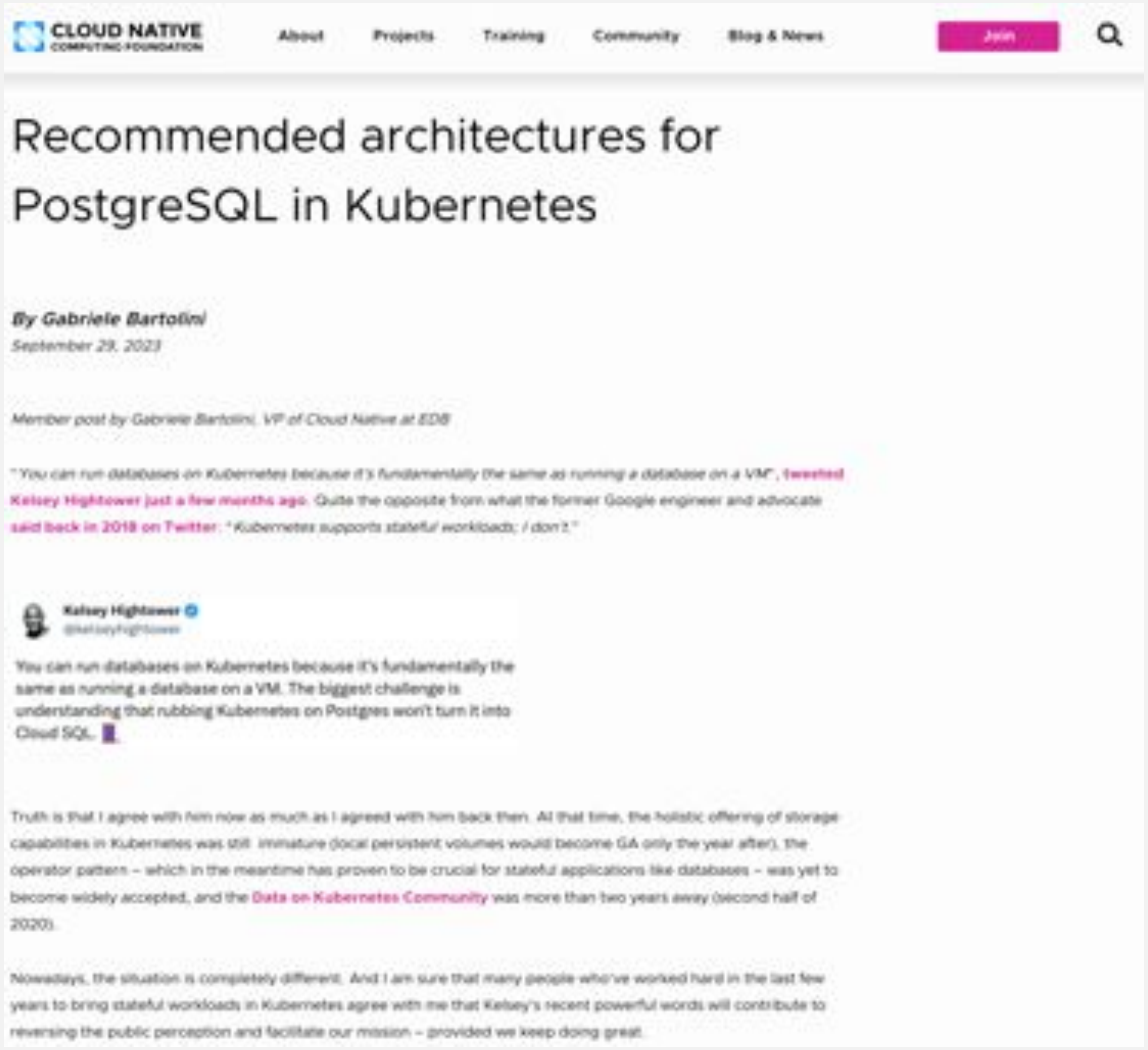
Barman repository



```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: cluster1
spec:
  instances: 3
  imageName: ghcr.io/cloudnative-pg/postgresql:14.3

  superuserSecret:
    name: superuser-secret

                                 bootstrap:
  bootstrap:                        recovery:
    initdb:                           backup:
      database: app                     name: backup-PIT
      owner: app-own
      secret:                        replica:
        name: app-secret               enabled: true
                                       source: cluster2
  storage:
    size: 100Gi

  monitoring:
   enablePodMonitor: true

  backup:
    barmanObjectStore:
      destinationPath: "s3://cloudnativepg/"
      endpointURL: "http://192.168.1.121:9000"
      s3Credentials:
        accessKeyId:
          name: minio-creds
          key: MINIO_ACCESS_KEY
        secretAccessKey:
          name: minio-creds
          key: MINIO_SECRET_KEY
      data:
        immediateCheckpoint: true
    retentionPolicy: "1w"
~
```

Recommended architectures

https://www.cncf.io/blog/2023/09/29/recommended-architectures-for-postgresql-in-kubernetes/



Recommended architectures for PostgreSQL in Kubernetes

By Gabriele Bartolini

September 29, 2023

Member post by Gabriele Bartolini, VP of Cloud Native at EDB

"*You can run databases on Kubernetes because it's fundamentally the same as running a database on a VM*", tweeted Kelsey Hightower just a few months ago. Quite the opposite from what the former Google engineer and advocate said back in 2018 on Twitter: "*Kubernetes supports stateful workloads; I don't.*"

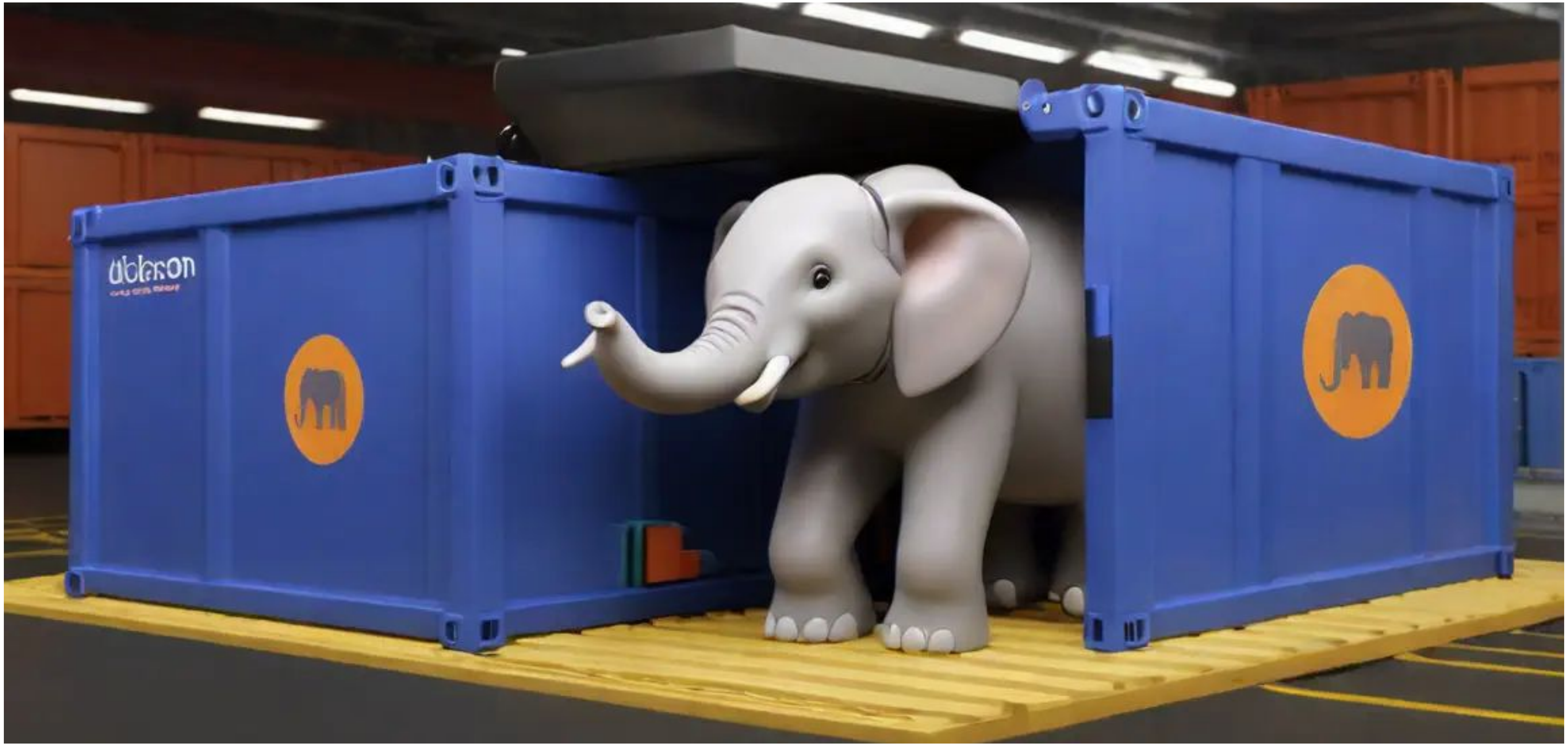
Kelsey Hightower
@kelseyhightower

You can run databases on Kubernetes because it's fundamentally the same as running a database on a VM. The biggest challenge is understanding that rubbing Kubernetes on Postgres won't turn it into Cloud SQL.

Truth is that I agree with him now as much as I agreed with him back then. At that time, the holistic offering of storage capabilities in Kubernetes was still immature (local persistent volumes would become GA only the year after), the operator pattern – which in the meantime has proven to be crucial for stateful applications like databases – was yet to become widely accepted, and the Data on Kubernetes Community was more than two years away (second half of 2020).

Nowadays, the situation is completely different. And I am sure that many people who've worked hard in the last few years to bring stateful workloads in Kubernetes agree with me that Kelsey's recent powerful words will contribute to reversing the public perception and facilitate our mission – provided we keep doing great.

Demo Time



Questions?

Thank you