What role did **CloudNativePG** (and EDB) play in this transformation?

And what does the **future** hold?

EDB
Postgres® for the AI Generation

The Past, Present, and Future of Postgres in Kubernetes

Gabriele Bartolini
VP, Chief Architect, Kubernetes at EDB
Dutch Cloud Native Day 2025, Utrecht, 3 July

Gabriele Bartolini

VP, Chief Architect of Kubernetes at EDB

PostgreSQL user since ~2000

Ex 2ndQuadrant (co-founder)

PostgreSQL Contributor

DoK Ambassador

DevOps evangelist

Open source contributor

- Barman (2011)
- CloudNativePG (2022)

Blog: gabrielebartolini.it @_GBartolini_

#1 Contributors to 

Creators of  CloudNativePG

enterprisedb.com

cloudnative-pg.io

postgresql.org

EDB
Postgres® for the AI Generation

Agenda

- Introduction
- The Past
- The Present
- The Future
- Conclusions

EDB
Postgres® for the AI Generation

Introduction

What is PostgreSQL?

The world's most advanced database. Also known as **Postgres**. URL: postgresql.org

- **100% Open Source**
  - Widely used, extremely robust, and feature-rich.
- **Extensible and Customizable**
  - Support for custom data types, functions, and procedural languages.
- **Advanced features**
  - Includes replication, partitioning, full-text search, and JSON support.
- **ACID-Compliant**
  - Ensures Atomicity, Consistency, Isolation, and Durability for transactions.
- **Strong Community Support**
  - Backed by a large, active global community.

CloudNativePG leverages PostgreSQL's native physical replication to synchronise states across different locations, including cascading and synchronous replication at the transaction level, as well as Hot Standby.

File storage replication in PostgreSQL has gradually become obsolete, starting in 2005 with the introduction of Warm Standby.

Although CloudNativePG is storage-agnostic, **relying on storage replication for PostgreSQL in Kubernetes is considered bad practice.** It also leads to “write-amplification” when used with Postgres replicas.

Allow read only connections during recovery, known as Hot Standby.

Browse files

Enabled by recovery_connections = on (default) and forcing archive recovery using a recovery.conf. Recovery processing now emulates the original transactions as they are replayed, providing full locking and MVCC behaviour for read only queries. Recovery must enter consistent state before connections are allowed, so there is a delay, typically short, before connections succeed. Replay of recovering transactions can conflict and in some cases deadlock with queries during recovery; these result in query cancellation after max_standby_delay seconds have expired. Infrastructure changes have minor effects on normal running, though introduce four new types of WAL record.

New test mode "make standbycheck" allows regression tests of static command behaviour on a standby server while in recovery. Typical and extreme dynamic behaviours have been checked via code inspection and manual testing. Few port specific behaviours have been utilised, though primary testing has been on Linux only so far.

This commit is the basic patch. Additional changes will follow in this release to enhance some aspects of behaviour, notably improved handling of conflicts, deadlock detection and query cancellation. Changes to VACUUM FULL are also required.

Simon Riggs, with significant and lengthy review by Heikki Linnakangas, including streamlined redesign of snapshot creation and two-phase commit.

Important contributions from Florian Pflug, Mark Kirkwood, Merlin Moncure, Greg Stark, Gianni Ciolli, Gabriele Bartolini, Hannu Krosing, Robert Haas, Tatsuo Ishii, Hiroyuki Yamada plus support and feedback from many other community members.

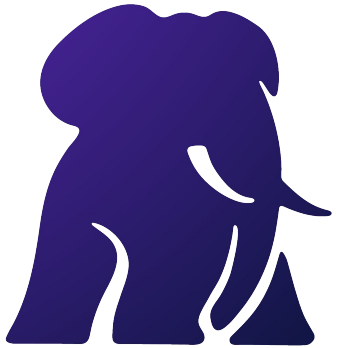
master

REL_17_BETA2 ... REL8_5_ALPHA3

simonat2ndQuadrant committed on Dec 19, 2009

1 parent 78a0914 commit efc16ea

EDB



CloudNativePG

is now a CNCF Sandbox Project

the first relational database to enter since 2018

the first ever for PostgreSQL

Key adopters

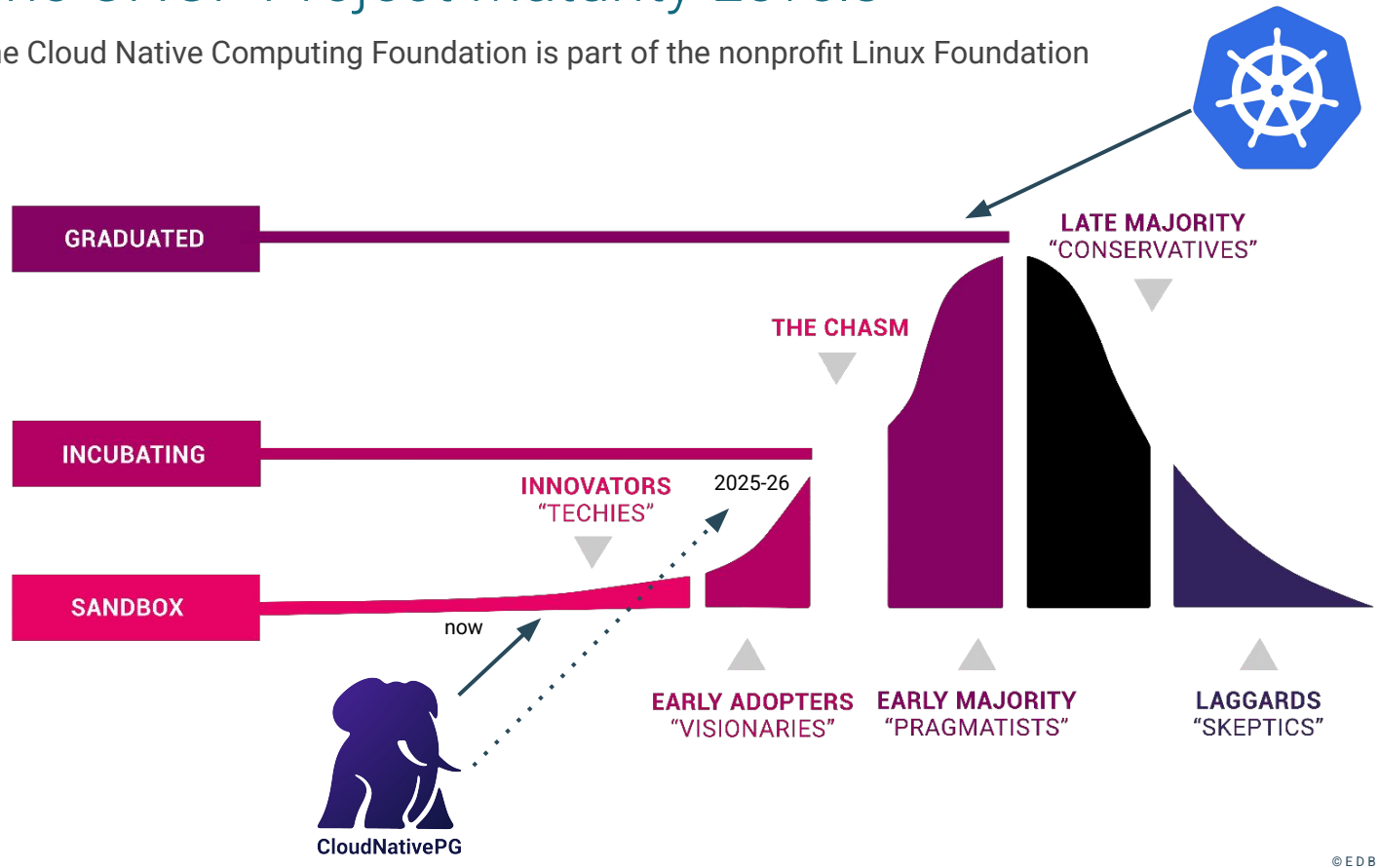
IBM Cloud Pak, Google Cloud, Azure, Bitnami, Akamai, Novo Nordisk, Hitachi

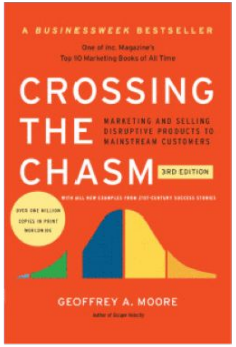
Target: CNCF Incubation in 2025-2026

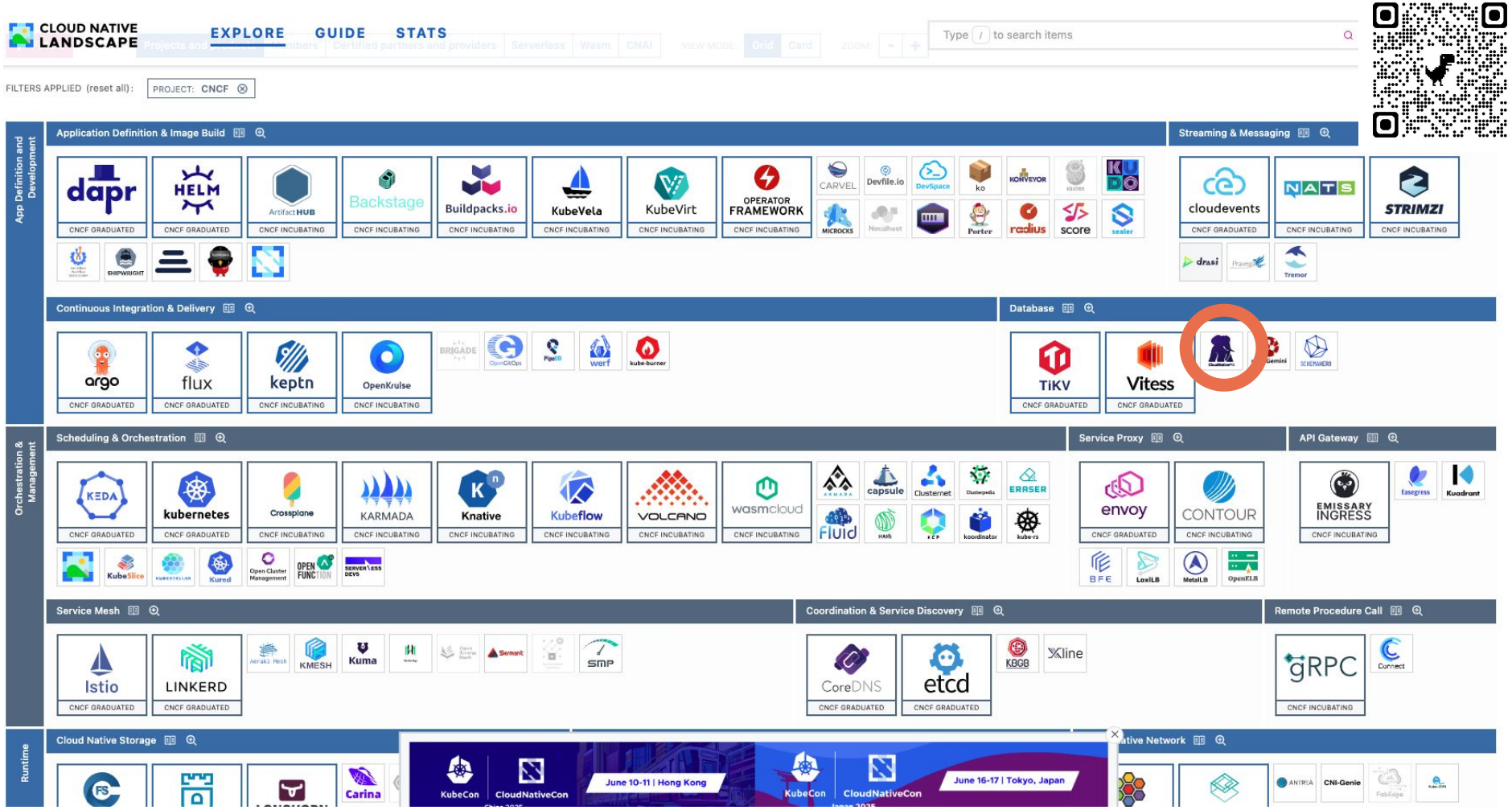
github.com/cloudnative-pg

The CNCF Project Maturity Levels

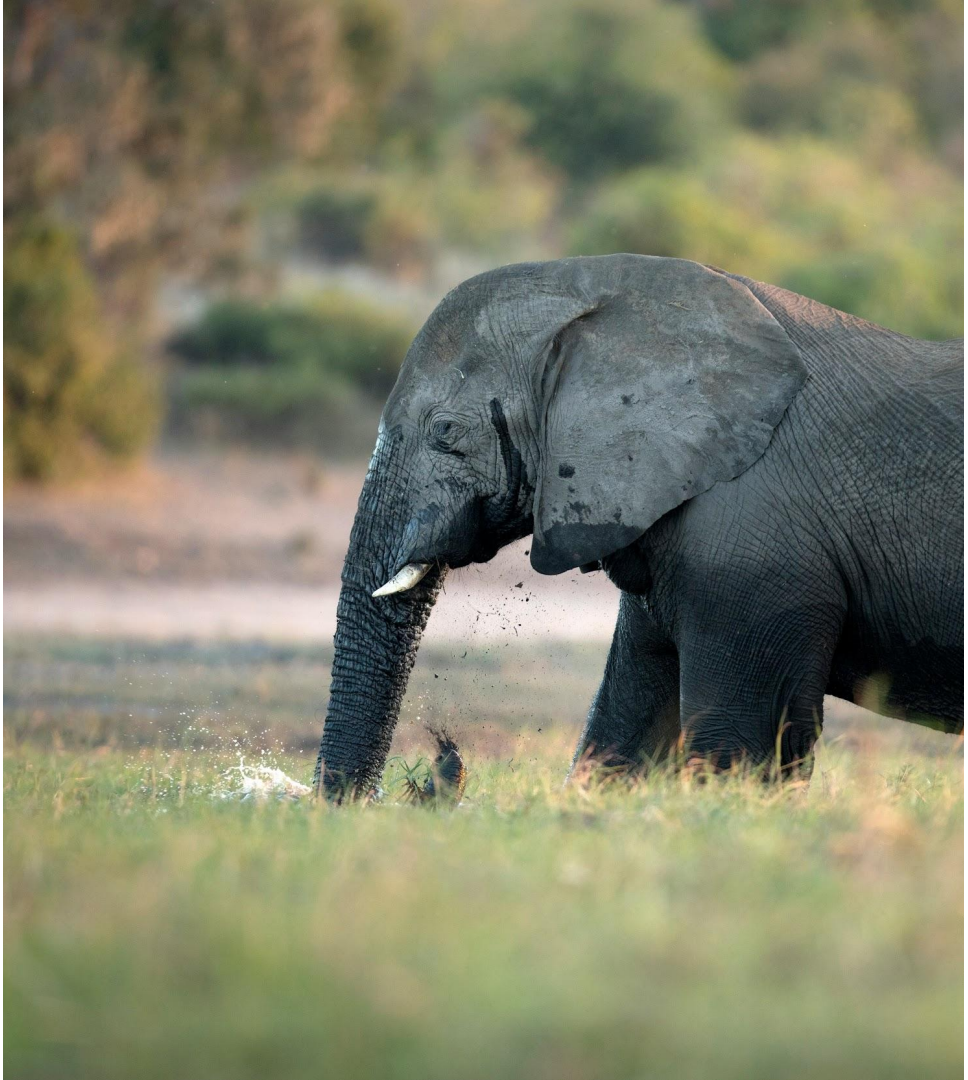
The Cloud Native Computing Foundation is part of the nonprofit Linux Foundation







https://landscape.cncf.io



EDB
Postgres® for the AI Generation

The past

First impressions last

Kubernetes was initially focused on stateless workloads

Kelsey Hightower
@kelseyhightower

Kubernetes has made huge improvements in the ability to run stateful workloads including databases and message queues, but I still prefer not to run them on Kubernetes.

Traduci il Tweet

3:04 PM · 13 feb 2018

Kelsey Hightower
@kelseyhightower

Kubernetes supports stateful workloads; I don't.

3:26 PM · 13 feb 2018

First impressions last

It takes time to change the general perception

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. 📕

Traduci il Tweet

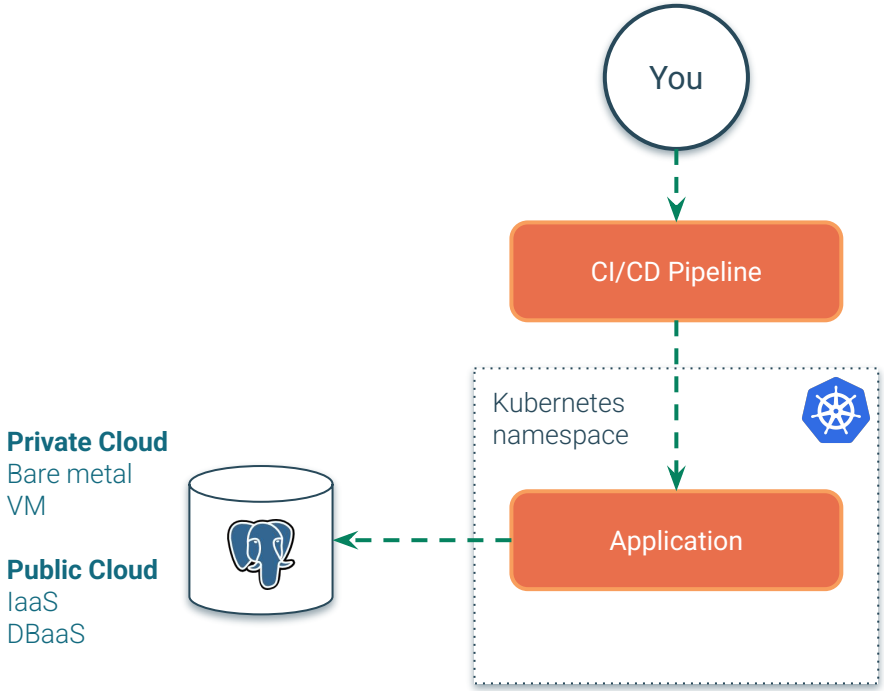
Soham Dasgupta @thesobercoder · 10 feb

@kelseyhightower Bust a myth for us please - running any sort of database on a Kubernetes instance is bad idea. I've heard this enough times to actually start believing it. #kubernetes #mythbuster

Mostra questa discussione

5:21 PM · 10 feb 2023 · **318.944** visualizzazioni

Database outside your Kubernetes cluster
(Bare metal, VM, IaaS, and DBaaS approaches)



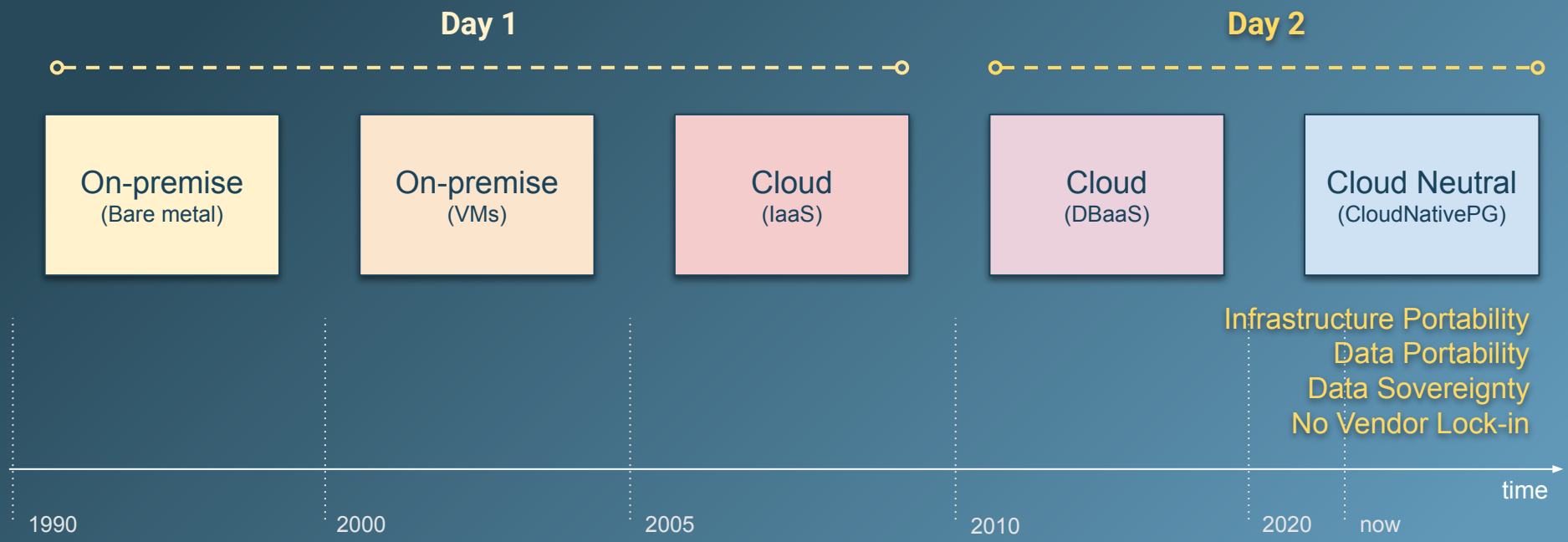
Evolution of PostgreSQL in containers

From Docker system containers to Kubernetes native databases with CloudNativePG

- **2013/3**: Docker is released. Postgres runs mainly for testing in system containers
- **2015/7**: Kubernetes 1.0 is released. Stateless applications only.
- **2016/11**: Operator pattern by CoreOS
- **2017/3**: Crunchy Data releases the first Postgres operator based on Patroni
- **2017/12**: Statefulsets are introduced in Kubernetes 1.9 (*1 year after beta in 1.5*)
- **2018/8**: Zalando releases their operator
- **2019/4**: Local persistent volumes are introduced in Kubernetes 1.14
- **2019/8**: The Cloud Native initiative at EDB (2ndQuadrant at that time) begins
- **2021/2:** EDB launches Cloud Native Postgres
- **2022/5: EDB open sources CloudNativePG**
- **2024/10**: CloudNativePG reaches 4500 stars on GitHub (#1 Postgres operator)
- **2025/1**: **CloudNativePG** becomes a **CNCF** project entering the **Sandbox**

The evolution of Postgres use cases

From Handcrafted PostgreSQL to Cloud-Neutral Automation with GitOps & K8s



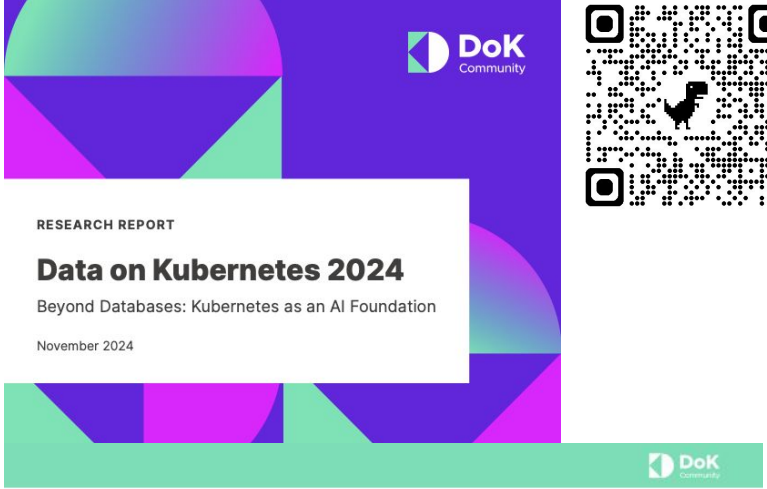
EDB
Postgres® for the AI Generation

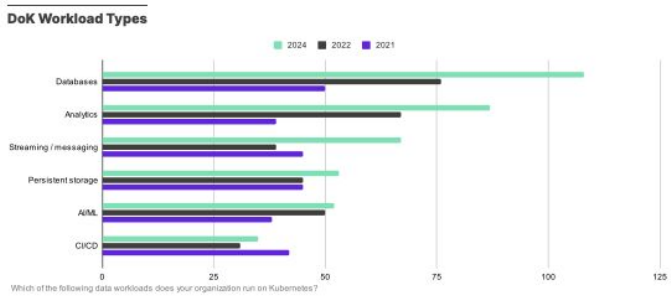
The present

Data on Kubernetes Community

Databases are #1 workload in Kubernetes

The Future of DBaaS on Kubernetes - M. Logan, S. Pronin, D. Sigireddi, G. Bartolini
Kubecon NA 2024 - Panel





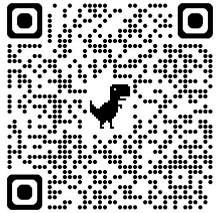
Database Workloads: The Steady Foundation

Databases continue to be the cornerstone of DoK deployments. For the third consecutive year, databases remain the most common DoK workload, demonstrating the platform's reliability for critical data services. The consistency in database workload adoption demonstrates:

1. **Platform Reliability:** Organizations trust Kubernetes for critical data services.
2. **Operational Standardization:** Growing comfort with running databases on Kubernetes.
3. **Deployment Confidence:** Increased willingness to run production database workloads.

Cloud Neutral PostgreSQL

Achieve cloud neutrality with the CloudNativePG stack (K8s, Postgres and CNPG)



- **Leverage infrastructure portability of Kubernetes**
  - Private, public, multi-cloud, hybrid cloud
  - The rise of data sovereignty is revitalising on-premises data centres
- **Provide internal/external DBaaS services with CloudNativePG**
  - Extend Day-2 operations to PostgreSQL with CloudNativePG
  - Implement IaC and GitOps for your PostgreSQL databases
- **Fully exploit PostgreSQL for data portability**
  - Think about the European Data Act
  - Streaming replication, both physical and logical
- **Cattle vs Pets? no … Elephants!**
  - Bare metal with local storage for shared nothing architectures and reduced VM costs!
  - Isolate PostgreSQL workloads with node taints (physical) and affinity rules (logical)
- **Free from any form of vendor lock-in**

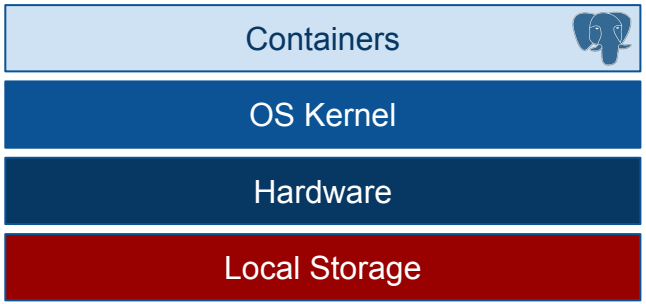
Suggested reading from the CNCF blog

Cloud Neutral Postgres Databases with Kubernetes and CloudNativePG

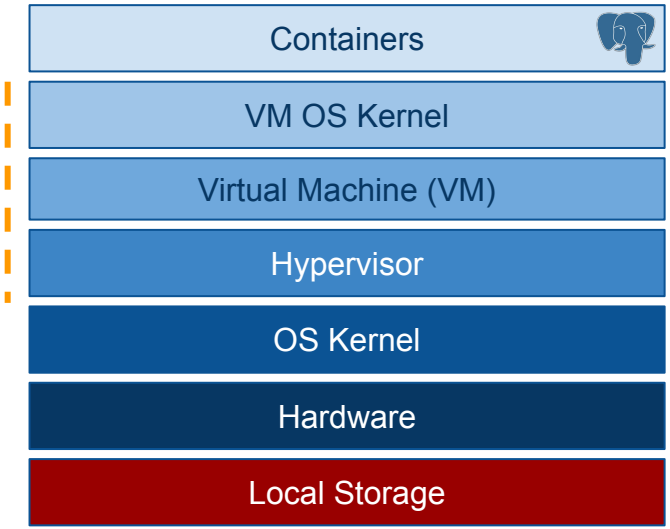
You can run Kubernetes on bare metal nodes

With locally attached and dedicated storage. Migrating “Postgres on VMs” to CloudNativePG on bare metal Kubernetes nodes.

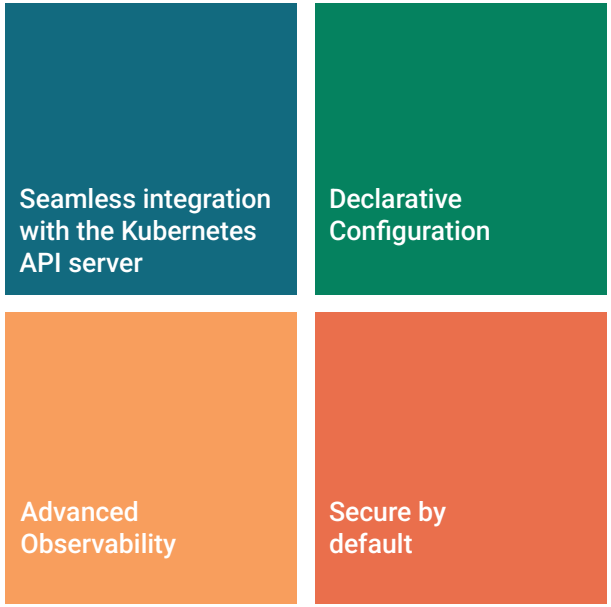
Bare metal



VMs



The **4 pillars** of CloudNativePG that make it a **Kubernetes native database**



CloudNativePG

github.com/cloudnative-pg

The main features of CloudNativePG

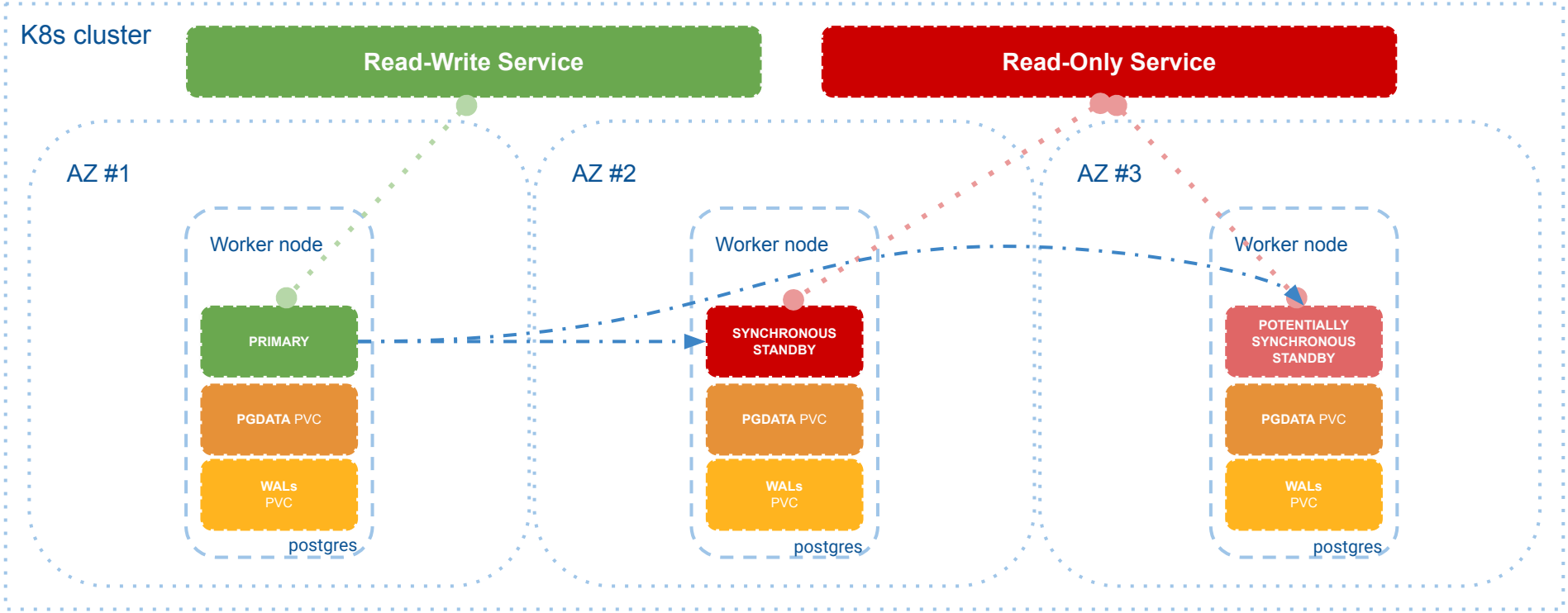
Overview of CloudNativePG's main features covering through Day 2 operations

- High Availability and Self-Healing
- Support for local PVCs
- Managed services for rw and ro workloads
- Continuous backup (including snapshots)
- Point In Time Recovery (incl. snapshots)
- Scale up/down of read-only replicas
- "Security by default", including mTLS
- Native Prometheus exporter
- Logging to stdout in JSON format
- Rolling updates, incl. minor Postgres releases
- Synchronous replication
- Major upgrades of Postgres
- Online import of Postgres databases
- Separate volume for WALs
- Postgres tablespaces, including temporary
- Replica clusters and distributed topologies
- Declarative roles and database
- Declarative hibernation and fencing
- Connection pooling
- Postgres extensions (pgvector, PostGIS, …)

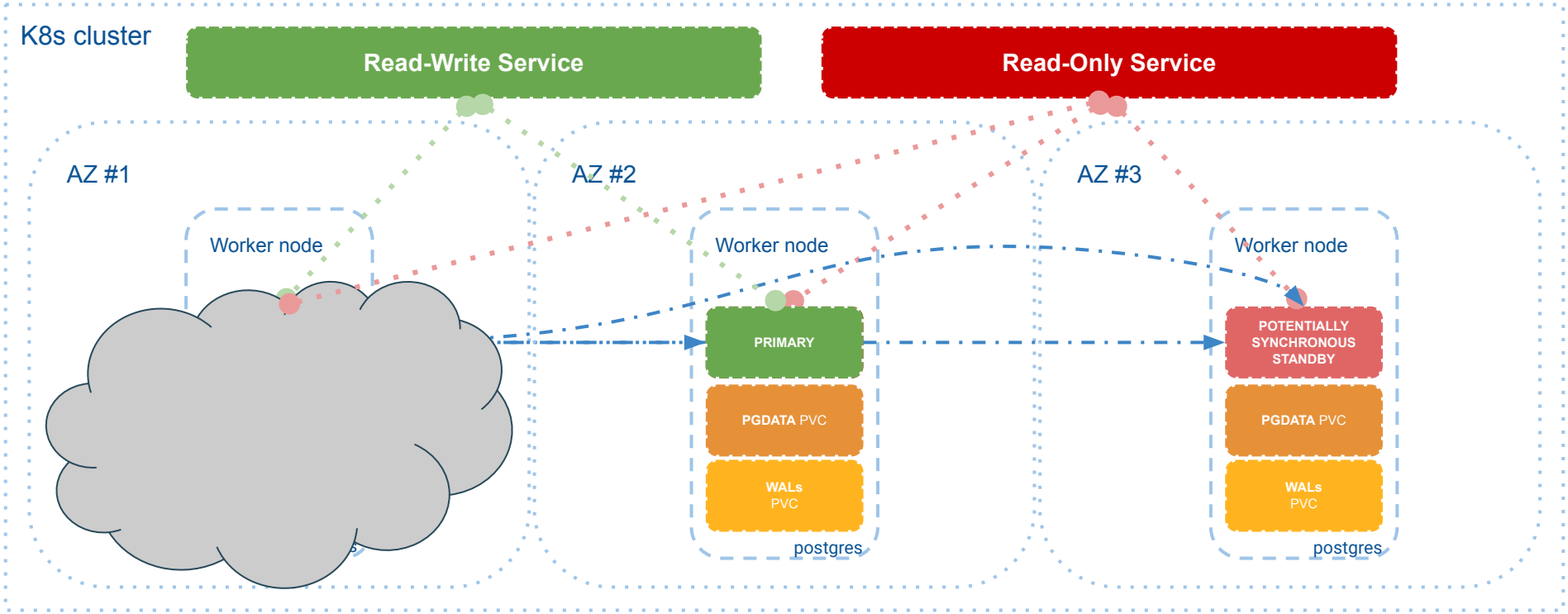
The PostgreSQL `Cluster` resource

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: clapton
spec:
  instances: 3
  affinity:
    nodeSelector:
      node-role.kubernetes.io/postgres: ""
  postgresql:
    synchronous:
      method: any
      number: 1
  storage:
    size: 40Gi
  walStorage:
    size: 10Gi
```

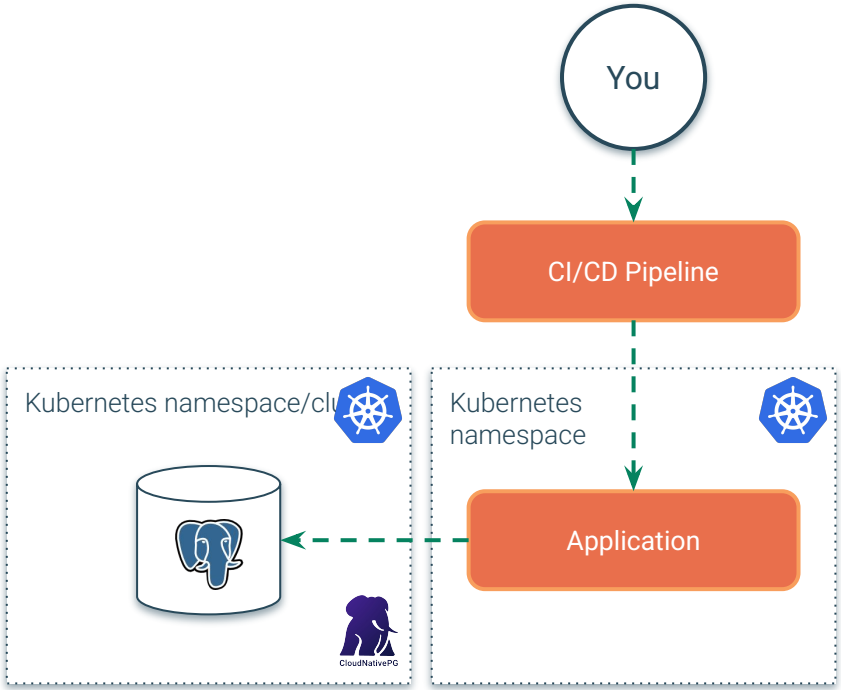
Highly Available PostgreSQL Cluster



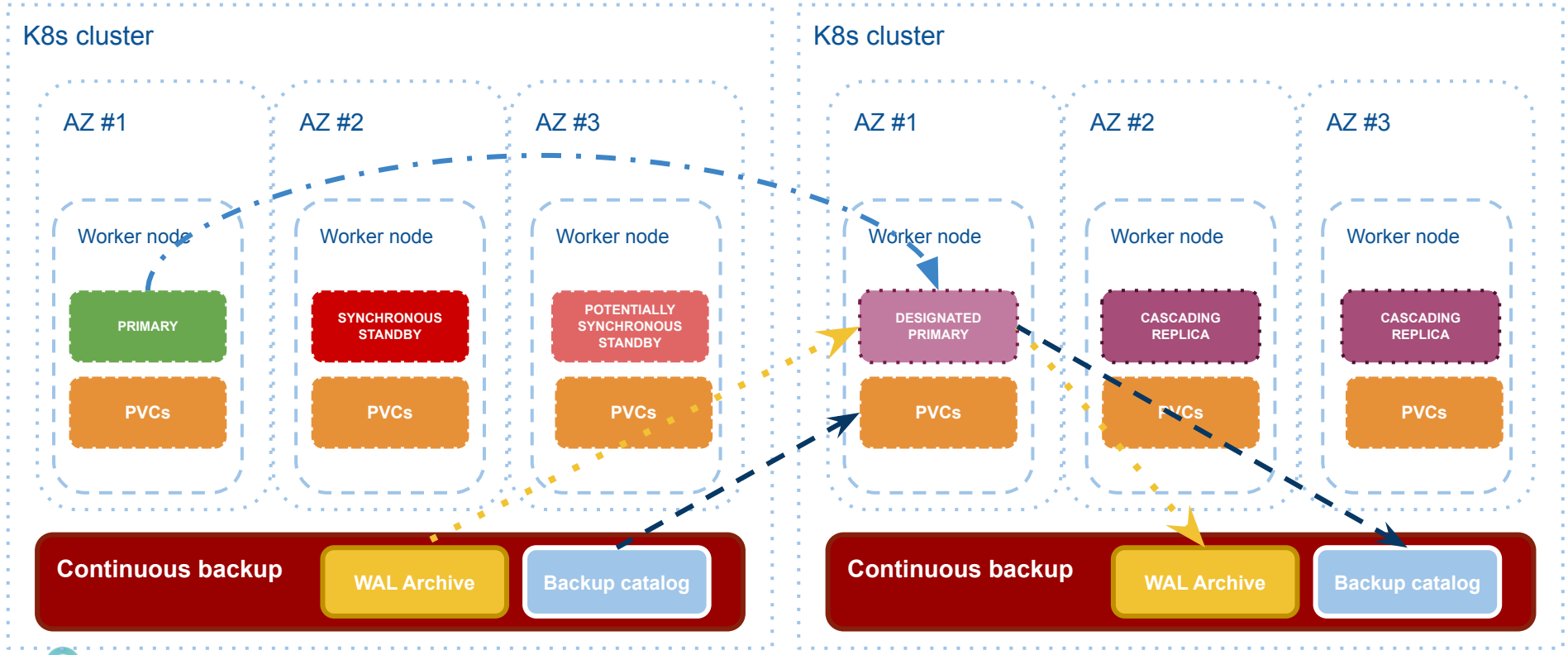
Automated failover (HA with very low RTO)



Database outside your Kubernetes namespace
(CloudNativePG internal DBaaS approach)



Symmetric architecture on 2 Kubernetes clusters

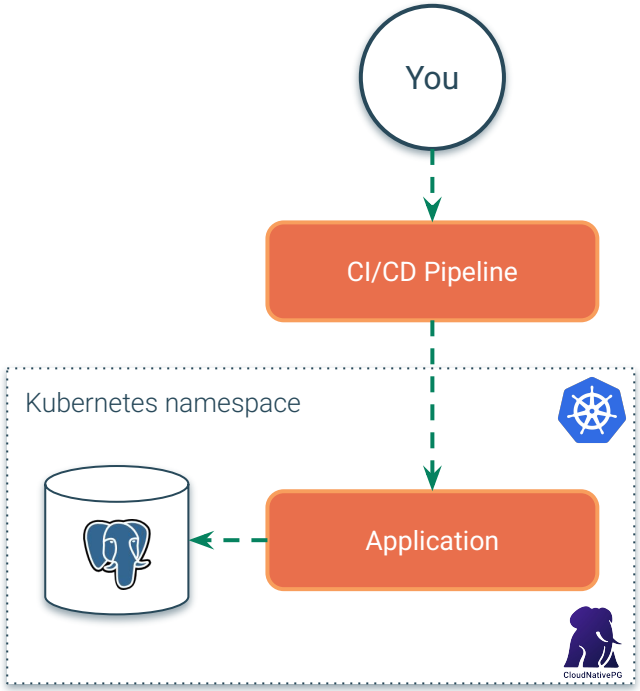


Suggested reading from the CNCF blog

EDB
Postgres® for the AI Generation

The future

Database inside your Kubernetes namespace
(CloudNativePG microservice database)

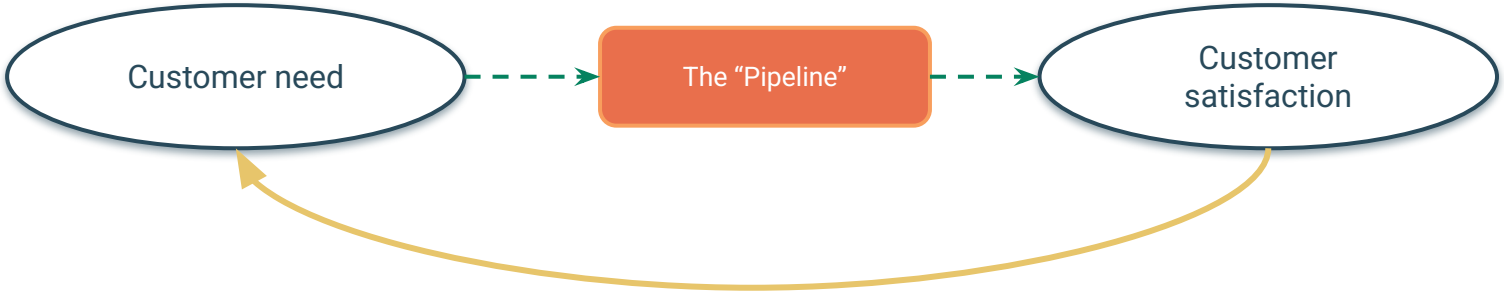


Stream aligned team

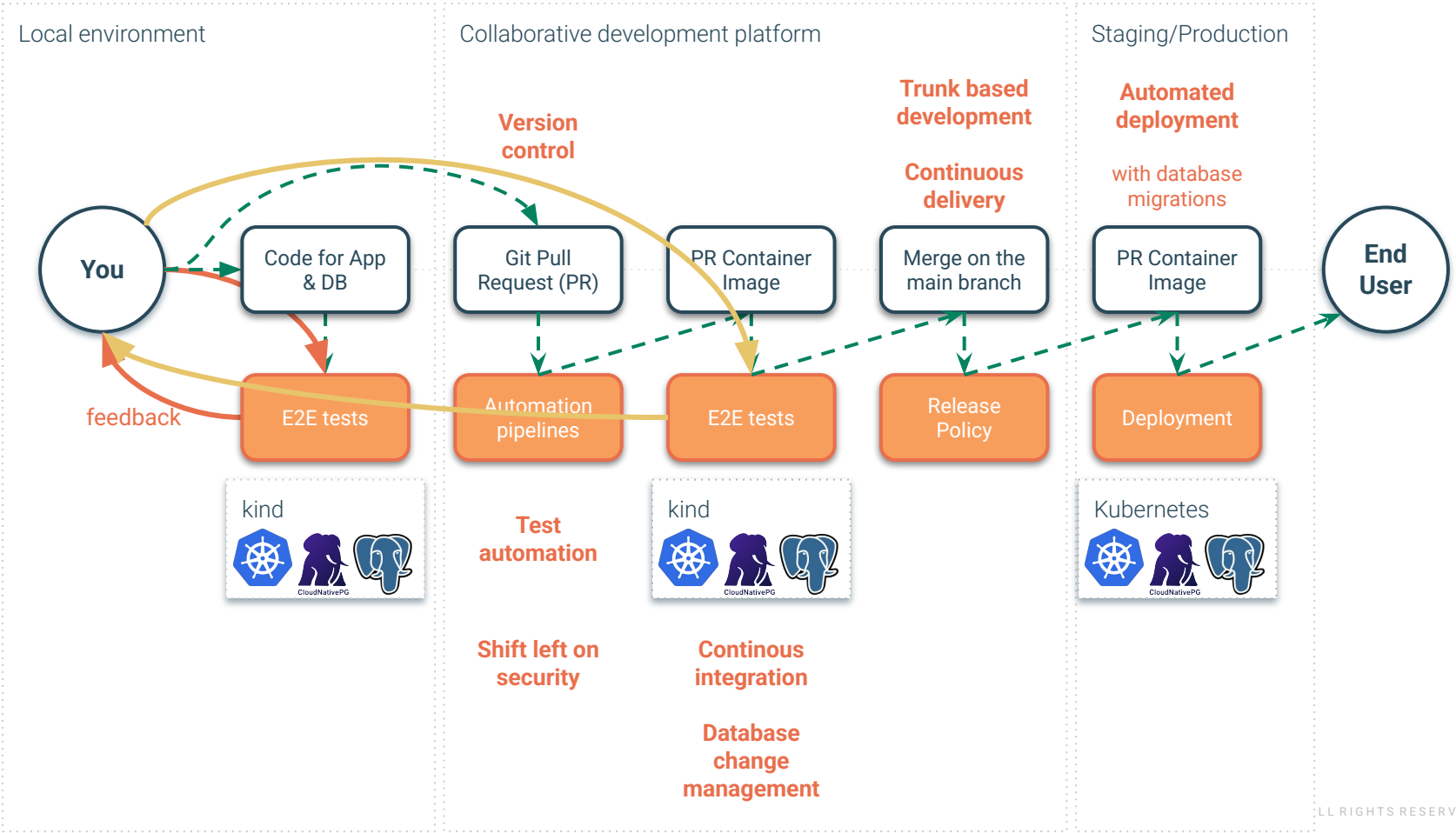
Suggested reading from my blog

Maximizing Microservice Databases with Kubernetes, Postgres, and CloudNativePG

How any customer-facing software should be ideally developed



DevOps capabilities, CloudNativePG databases and automated pipelines



Roadmap

What lies ahead

- **CNCF Incubation**
- **Features**
  - **PostgreSQL 18 support**
  - **Dynamic Loading of Extensions**
  - **Isolation checker for the liveness probe of a primary**
  - Quorum Failover Protection
  - Make backup and recovery available via plugins only
    - including volume snapshots and volume group snapshots
  - `Role` resource (declarative role management)
  - Foreign Data Wrappers

Suggested reading

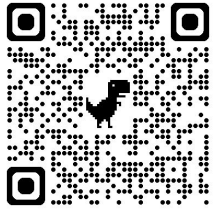
The Immutable Future of PostgreSQL Extensions in Kubernetes with CloudNativePG

3 March 2025 · 8 mins

kubernetes k8s cloudnativepg cnpg postgresql postgres dok data on kubernetes extensions Container images sbom pgvector imagevolume extension_control_path

Commit `4f7f7b0`

 **petere** and **mattheusv** committed ✓ 7 / 10



```
extension_control_path

The new GUC extension_control_path specifies a path to look for
extension control files.  The default value is $system, which looks in
the compiled-in location, as before.

The path search uses the same code and works in the same way as
dynamic_library_path.

Some use cases of this are: (1) testing extensions during package
builds, (2) installing extensions outside security-restricted
containers like Python.app (on macOS), (3) adding extensions to
PostgreSQL running in a Kubernetes environment using operators such as
CloudNativePG without having to rebuild the base image for each new
extension.

There is also a tweak in Makefile.global so that it is possible to
install extensions using PGXS into an different directory than the
default, using 'make install prefix=/else/where'.  This previously
only worked when specifying the subdirectories, like 'make install
datadir=/else/where/share pkglibdir=/else/where/lib', for purely
implementation reasons.  (Of course, without the path feature,
installing elsewhere was rarely useful.)

Author: Peter Eisentraut <peter@eisentraut.org>
Co-authored-by: Matheus Alcantara <matheusssilv97@gmail.com>
Reviewed-by: David E. Wheeler <david@justatheory.com>
Reviewed-by: Gabriele Bartolini <gabriele.bartolini@enterprisedb.com>
Reviewed-by: Marco Nenciarini <marco.nenciarini@enterprisedb.com>
Reviewed-by: Niccolò Fei <niccolo.fei@enterprisedb.com>
Discussion: https://www.postgresql.org/message-id/flat/E7C7BFFB-8857-48D4-A71F-88B359FADCFD@justatheory.com
```

CloudNativePG

View Repository

View Site

Go programming (operator development) | Kubernetes and CRDs (Custom Resource Definitions)

CNCF - CloudNativePG: Declarative Management of PostgreSQL FDWs (2025 Term 2)

openssf best practices | in progress 15%

Terms

Term 2: Jun - Aug

This project aims to extend the CloudNativePG operator to support declarative configuration of foreign data wrappers through its Database custom resource. PostgreSQL supports the SQL/MED (Management of External Data) specification, enabling access to external data sources through standard SQL queries. These sources—known as foreign data—are accessed via foreign data wrappers (FDWs), which are libraries that handle the connection and data exchange with the...

View More

Code of Conduct

Applications Closed

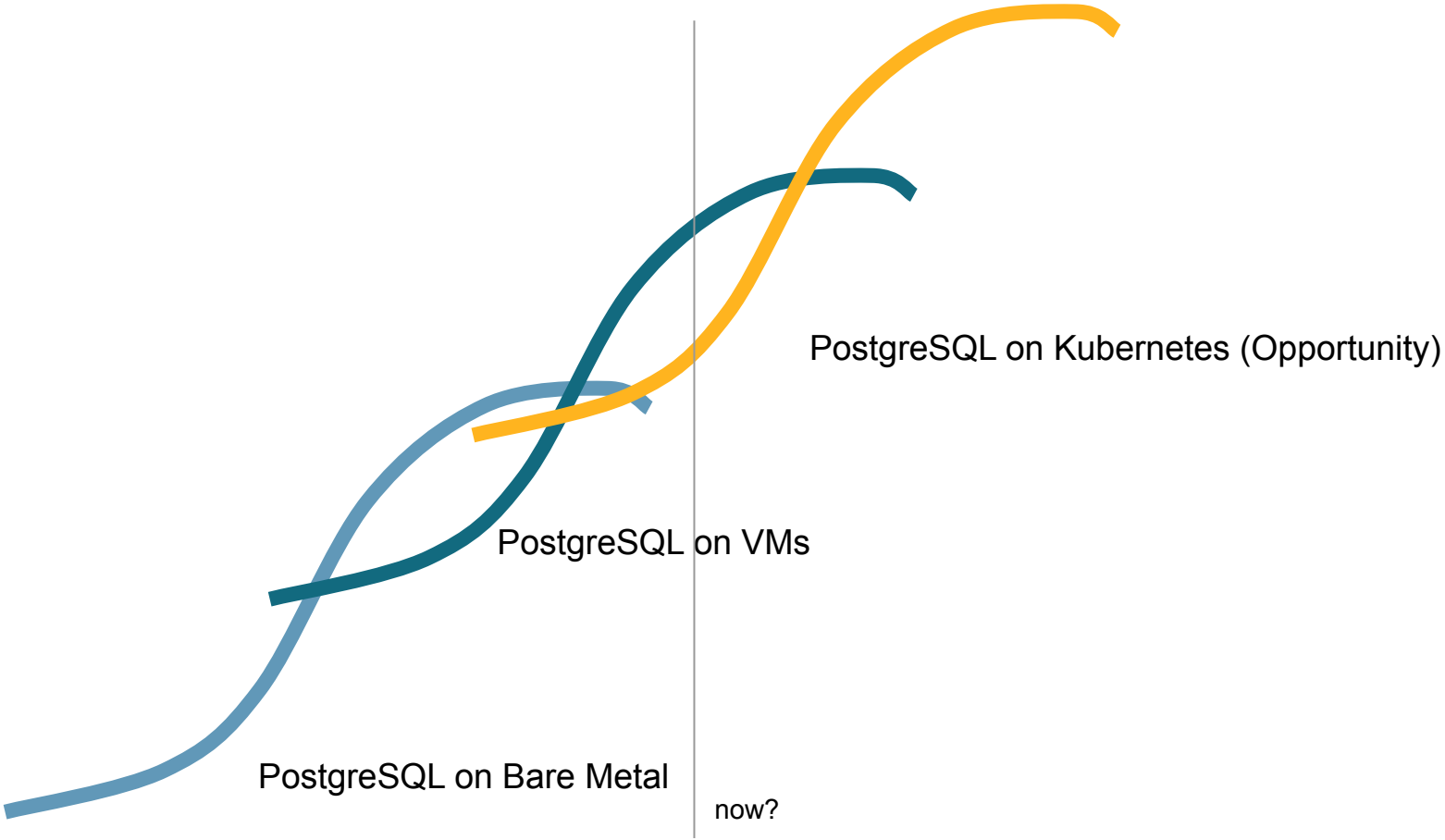
Mentees

Mentors

Sponsor Organizations

EDB
Postgres® for the AI Generation

Conclusions



EDB

Cattle or **pets**? Better … **herds**

“Pinnawala orphanage provides a lifeline to the orphaned baby elephants and orphaned elephants lost in the wilderness” by Puviraj Diluckshan

Key take aways

Your opportunity to run Postgres databases in Kubernetes starts today

- PostgreSQL is the most popular database
- From “Kubernetes is stateless” to “Databases are #1 workloads”
- CloudNativePG is the most popular operator for Postgres
- CloudNativePG and EDB are pushing the boundaries of Data on Kubernetes
- Internal/External DBaaS is the most popular use case for CNPG
  - Migration of VMs to bare metal K8s for license cost savings
  - ~0 cutover migrations/upgrades to CNPG via logical replication
- Next step: microservice databases
- Cloud Neutrality for PostgreSQL databases

Thank you

Maintainers

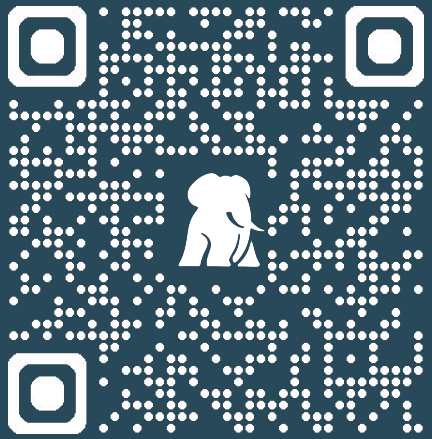
- Gabriele Bartolini
- Francesco Canovai
- Leonardo Cecchi
- Jonathan Gonzalez
- Marco Nenciarini
- Armando Ruocco
- Philippe Scorsolini

All the contributors, users and community members <3

EDB
For generously donating the intellectual property for this project and for their decades of invaluable contributions to the broader PostgreSQL ecosystem.

Join our community and help us shape the future of Postgres in Kubernetes!

Questions?



gabrielebartolini.it

@_GBartolini_

Creators of  **CloudNativePG**

enterprisedb.com

cloudnative-pg.io

postgresql.org

dok.community

cncf.io

The virtuous cycle of open-source software sustainability

Help us innovate through open-source software!