EDB Open Source
Learning Day

How to run PostgreSQL on Kubernetes (DEMO)

Sergio Romera | Senior Sales Engineer

EDB™
POWER TO POSTGRES

About me

- Sergio Romera ()
- Based in France, (Île-de-France near to Paris)
- Database fanatic since 1997
- Developer, DBA, Architect, Sales Engineer
- Companies: BNPParibas, Oracle, Quest Software
- Senior Sales Engineer at EDB

EDB

Why did PostgreSQL win?

It does everything...

- Migration
- New App Development
- Replatforming to Cloud and Containers
- System of Record
- System of Analysis
- System of Engagement

It works everywhere...

- Public Cloud - IaaS
- Public Cloud - DBaaS
- Private Cloud
- Virtual Machines
- Containers

EDB

A kubernetes operator for Postgres

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

Handling PostgreSQL clusters “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

EDB

CloudNativePG/EDB Postgres for Kubernetes

- Kubernetes operator for PostgreSQL
- "Level 5", Production ready
- Day 1 & 2 operations of a PostgreSQL database
  - In traditional environments usually reserved to humans
- Open source
  - Originally created and developed by EDB
  - Vendor neutral/openly governed community
  - Apache 2.0 license
  - Submitted to the CNCF Sandbox
- Fully declarative

Some 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 compatibility
- TDE (in EDB Postgres for Kubernetes)
- … and much more

EDB

	CloudNativePG	EDB Postgres for Kubernetes		
	Community support	Professional support service plans by EDB		
		Community 360 plan	Standard plan	Enterprise plan
Community Kubernetes	☑	☑	☑	☑
PGDG PostgreSQL versions 11 through 15	☑	☑	☑	☑
Break/fix support	☐	☑	☑	☑
Amazon EKS	☐	☑	☑	☑
Microsoft AKS	☐	☑	☑	☑
Google GKE	☐	☑	☑	☑
Long Term Support version	☐	☐	☑	☑
OpenShift Container Platform (including ppc64le and s390x)	☐	☐	☑	☑
Rancher	☐	☐	☑	☑
Velero/OADP cold backup support	☐	☐	☑	☑
Generic adapter for third-party Kubernetes backup tools	☐	☐	☑	☑
EDB Postgres Extended (*)	☐	☐	☑	☑
Transparent Data Encryption (TDE)	☐	☐	☑	☑
Access to EDB Postgres Distributed (*)	☐	☐	☑	☑
Oracle compatibility through EPAS	☐	☐	☐	☑

EDB

Architectures - Multi-availability zone Kubernetes clusters

- The multi-availability zone Kubernetes architecture with three (3) or more zones is the one that we recommend for PostgreSQL usage. This scenario is typical of Kubernetes services managed by Cloud Providers.

EDB

Architectures - Single availability zone Kubernetes clusters

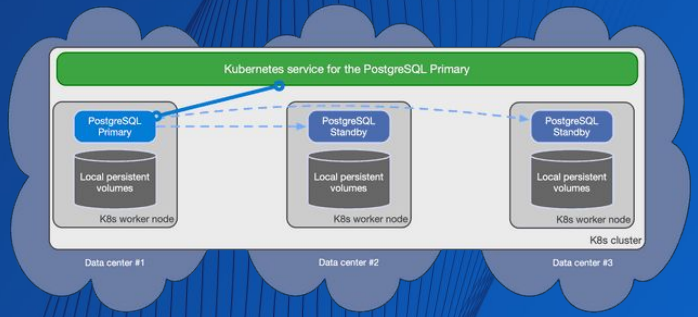
- If your Kubernetes cluster has only one availability zone, EDB Postgres for Kubernetes still provides you with a lot of features to improve HA and DR outcomes for your PostgreSQL databases, pushing the single point of failure (SPoF) to the level of the zone as much as possible - i.e. the zone must have an outage before your EDB Postgres for Kubernetes clusters suffer a failure.

- This scenario is typical of self-managed on-premise Kubernetes clusters, where only one data center is available.

EDB

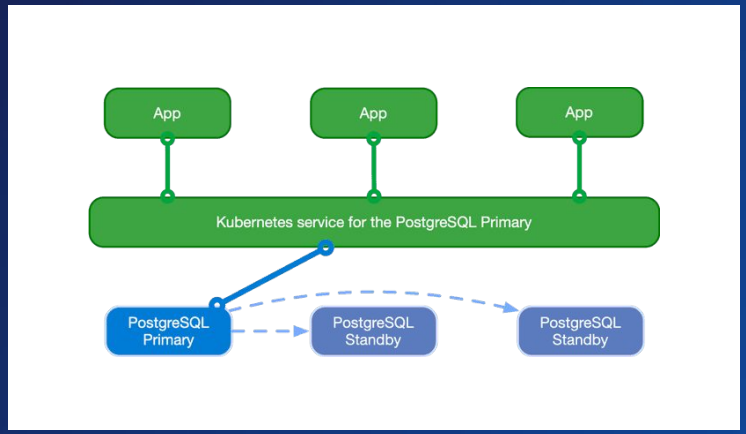
PostgreSQL architecture

- EDB Postgres for Kubernetes supports clusters based on **asynchronous and synchronous** streaming replication to manage multiple hot standby replicas within the same Kubernetes cluster, with the following specifications:
  - **One primary**, with optional multiple hot standby replicas for HA
  - **Available services** for applications:
    - -rw: applications connect only to the primary instance of the cluster
    - -ro: applications connect only to hot standby replicas for read-only-workloads
    - -r: applications connect to any of the instances for read-only workloads
  - Shared-nothing architecture recommended for better resilience of the PostgreSQL cluster:

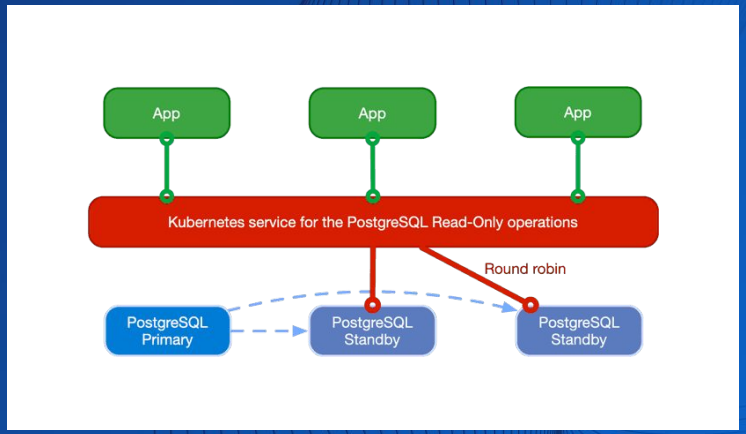


PostgreSQL workloads

Read-write work loads



Read-only work loads

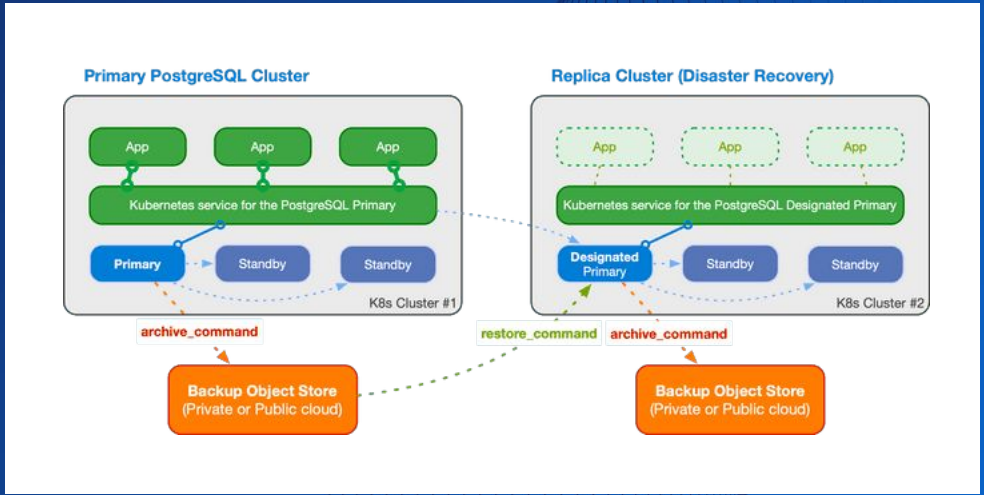


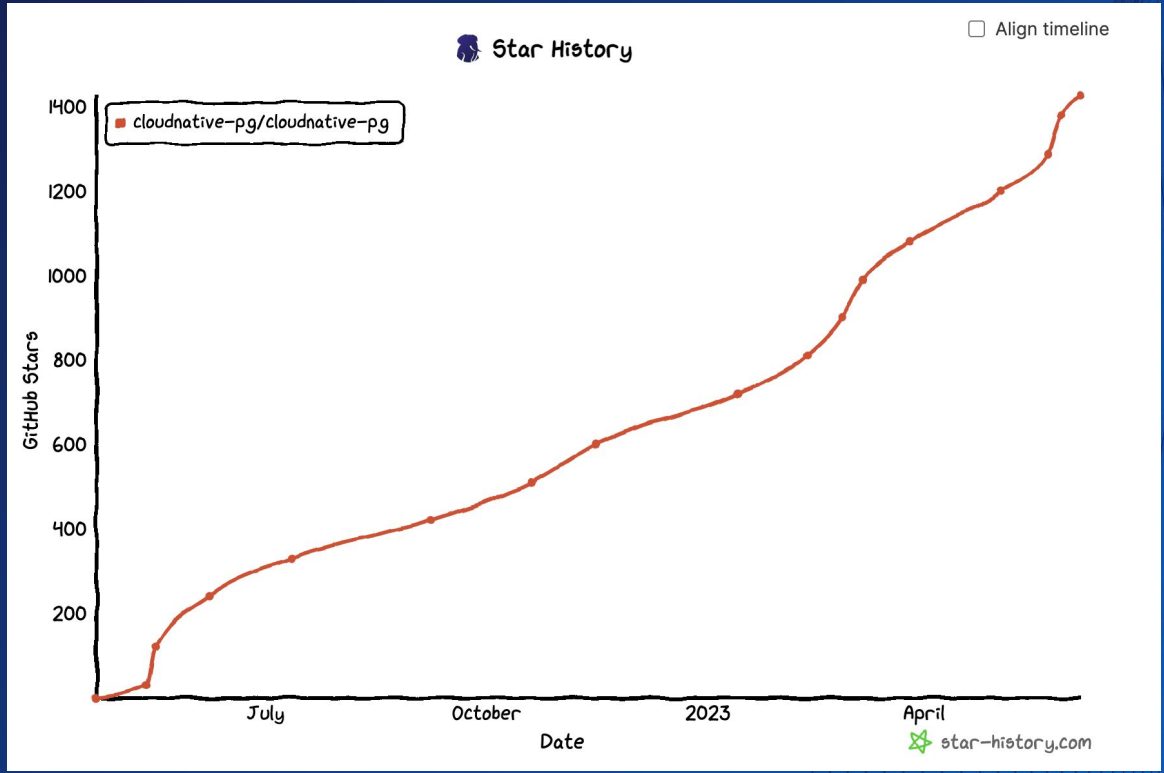
EDB

PostgreSQL Disaster Recovery

- PostgreSQL cluster spanning over two different Kubernetes clusters, where the primary cluster is in the first Kubernetes cluster and the replica cluster is in the second. The second Kubernetes cluster acts as the company's disaster recovery cluster, ready to be activated in case of disaster and unavailability of the first one.

- A replica cluster can have the same architecture of the primary cluster. In place of the primary instance, a replica cluster has a designated primary instance, which is a standby server with an arbitrary number of cascading standby servers in streaming replication (symmetric architecture).





Updated: 5/06/2023

Command line interface

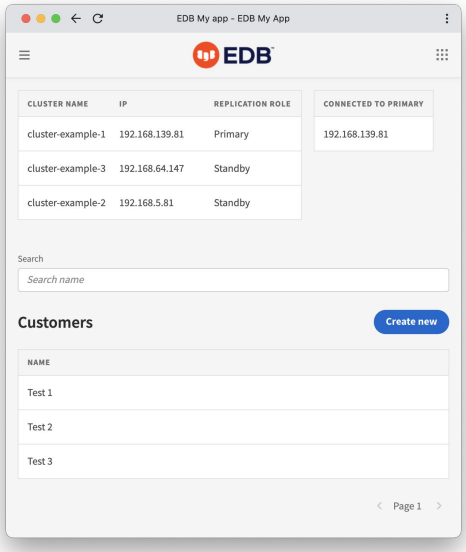
```
Cluster Summary
Name:               cluster-example
Namespace:          default
System ID:          7208481368169164826
PostgreSQL Image:   ghcr.io/cloudnative-pg/postgresql:14.2
Primary instance:   cluster-example-1
Status:             Cluster in healthy state
Instances:          3
Ready instances:    3
Current Write LSN:  0/4000060 (Timeline: 1 - WAL File: 000000010000000000000004)

Certificates Status
Certificate Name            Expiration Date               Days Left Until Expiration
----------------            ---------------               --------------------------
cluster-example-ca          2023-06-07 09:43:47 +0000 UTC  90.00
cluster-example-replication 2023-06-07 09:43:47 +0000 UTC  90.00
cluster-example-server      2023-06-07 09:43:47 +0000 UTC  90.00

Continuous Backup status
Not configured

Streaming Replication status
Name              Sent LSN   Write LSN  Flush LSN  Replay LSN  Write Lag  Flush Lag  Replay Lag  State      Sync State  Sync Priority
----              --------   ---------  ---------  ----------  ---------  ---------  ----------  -----      ----------  -------------
cluster-example-2 0/4000060  0/4000060  0/4000060  0/4000060   00:00:00   00:00:00   00:00:00    streaming  quorum      1
cluster-example-3 0/4000060  0/4000060  0/4000060  0/4000060   00:00:00   00:00:00   00:00:00    streaming  quorum      1

Unmanaged Replication Slot Status
No unmanaged replication slots found

Instances status
Name              Database Size  Current LSN  Replication role  Status  QoS        Manager Version  Node
----              -------------  -----------  ----------------  ------  ---        ---------------  ----
cluster-example-1 33 MB          0/4000060    Primary           OK      Burstable  1.19.0           docker-desktop
cluster-example-2 33 MB          0/4000060    Standby (sync)    OK      Burstable  1.19.0           docker-desktop
cluster-example-3 33 MB          0/4000060    Standby (sync)    OK      Burstable  1.19.0           docker-desktop
```

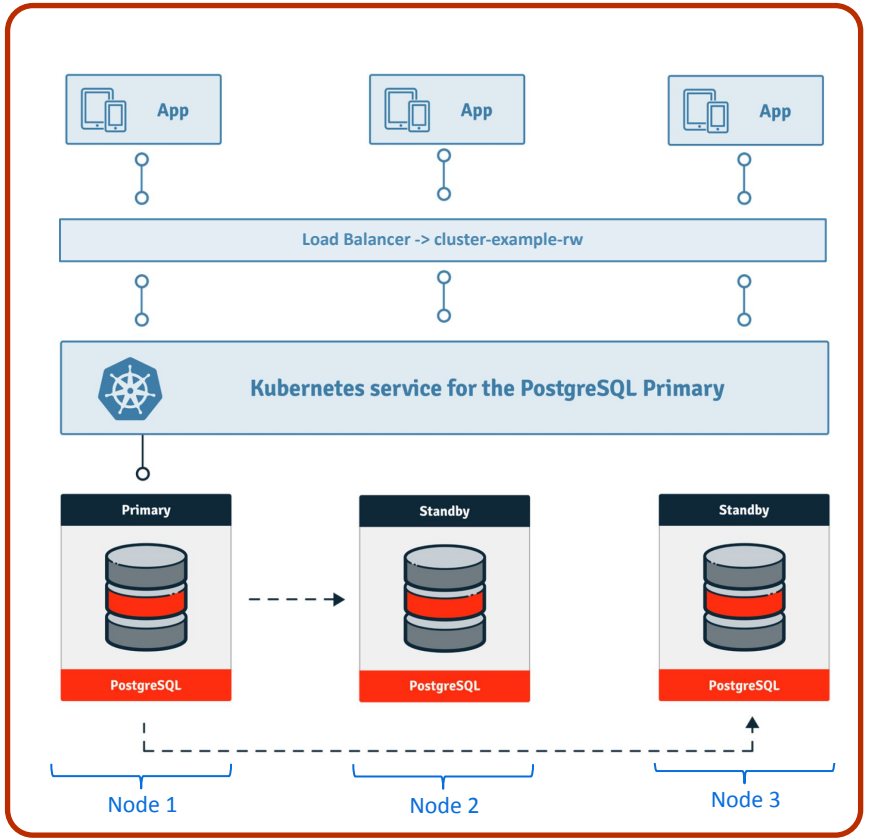
Demo

EDB

Demo Architecture

kubernetes





EDB

Last CloudNativePG tested version is 1.20.0

Features demo

- Kubernetes plugin install
- CloudNativePG operator install
- Postgres cluster install
- Insert data in the cluster
- Switchover (promote)
- Failover
- Backup
- Recovery
- Scale out/down
- Point In Time Recovery (PITR)
- Fencing
- Monitoring
- Rolling updates (minor and major)

EDB

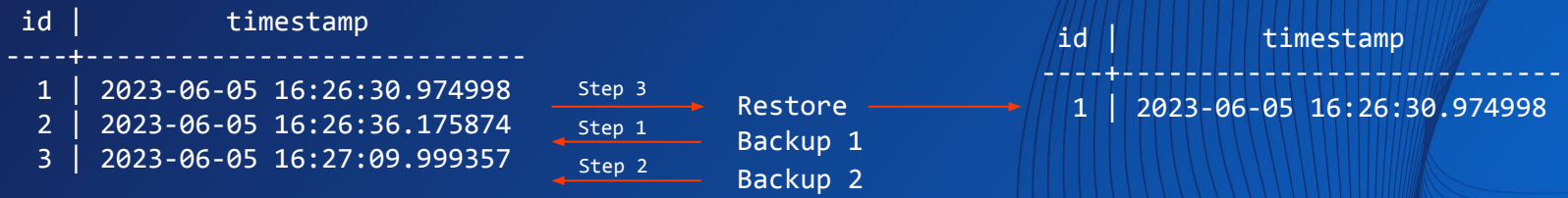
Replication Slots for High Availability

- Replication slots are a native PostgreSQL feature introduced in 9.4 that provides an automated way to ensure that the primary does not remove WAL segments until all the attached streaming replication clients have received them, and that the primary does not remove rows which could cause a recovery conflict even when the standby is (temporarily) disconnected.
- A replication slot exists solely on the instance that created it, and PostgreSQL does not replicate it on the standby servers.
- As a result, after a failover or a switchover, the new primary does not contain the replication slot from the old primary.
- This can create problems for the streaming replication clients that were connected to the old primary and have lost their slot.

EDB

Point in Time Recovery (PITR) from a backup

- The operator enables you to create a new PostgreSQL cluster by recovering an existing backup to a specific point-in-time, defined with a timestamp, a label or a transaction ID.
- This capability is built on top of the full restore one and supports all the options available in PostgreSQL for PITR.



Fencing

- Fencing in EDB Postgres for Kubernetes is the ultimate process of protecting the data in one, more, or even all instances of a PostgreSQL cluster when they appear to be malfunctioning. When an instance is fenced, the PostgreSQL server process (postmaster) is guaranteed to be shut down, while the pod is kept running.
- This makes sure that, until the fence is lifted, data on the pod is not modified by PostgreSQL and that the file system can be investigated for debugging and troubleshooting purposes.

EDB

Hibernation

- EDB Postgres for Kubernetes is designed to keep PostgreSQL clusters up, running and available anytime.
- There are some kinds of workloads that require the database to be up only when the workload is active. Batch-driven solutions are one such case.
- In batch-driven solutions, the database needs to be up only when the batch process is running.
- The declarative hibernation feature enables saving CPU power by removing the database Pods, while keeping the database PVCs.

EDB

cluster-example.yaml

- Cluster name: cluster-example
- 3 Instances using replication slots
  - 1 Primary
  - 2 Standby's
- PostgreSQL 14.2
- Min 1 sync replica
- Activate pg_stat_statement extension
- 1GB disk
- Activate monitoring metrics
- CPU
  - Request: 1
  - Limit: 2

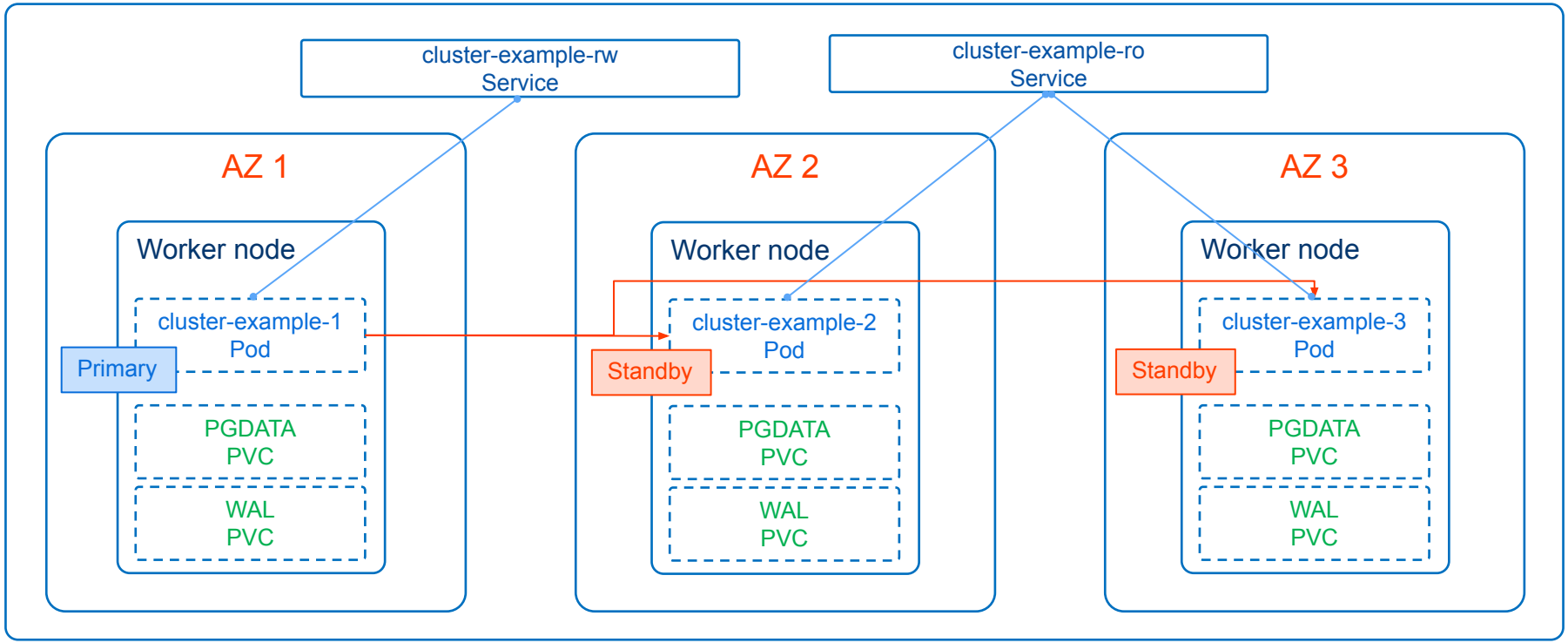
EDB

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

  # Reduce the frequency of standby HA slots updates to once every 5 minutes
  replicationSlots:
    highAvailability:
      enabled: true

  minSyncReplicas: 1
  maxSyncReplicas: 1

  postgresql:
    parameters:
      pg_stat_statements.max: "10000"
      pg_stat_statements.track: all

  storage:
    size: 1Gi

  monitoring:
    enablePodMonitor: true

  resources:
    requests:
      memory: "512Mi"
      cpu: "1"
    limits:
      memory: "1Gi"
      cpu: "2"
```

Monitoring

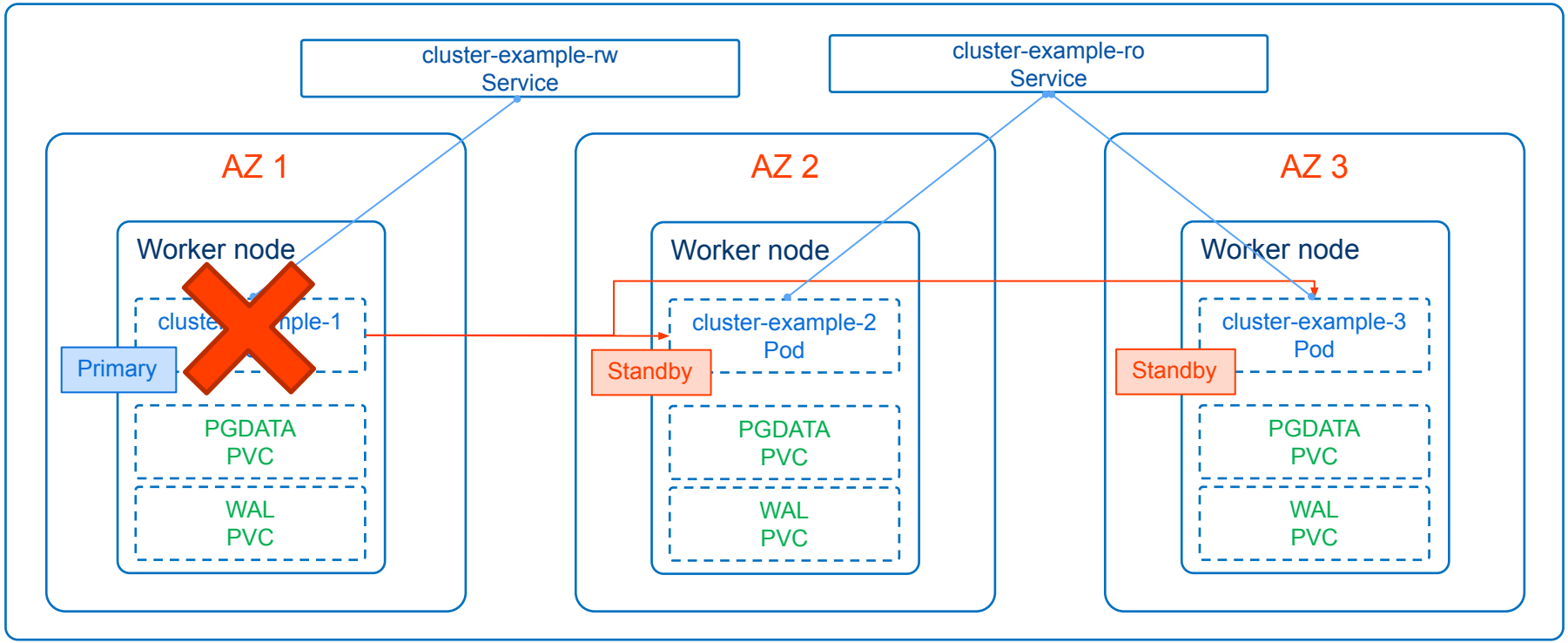
- For each PostgreSQL instance, the operator provides an exporter of metrics for Prometheus via HTTP, on port 9187, named metrics.
- The operator comes with a predefined set of metrics, as well as a highly configurable and customizable system to define additional queries via one or more ConfigMap or Secret resources

This is what happens under the hood

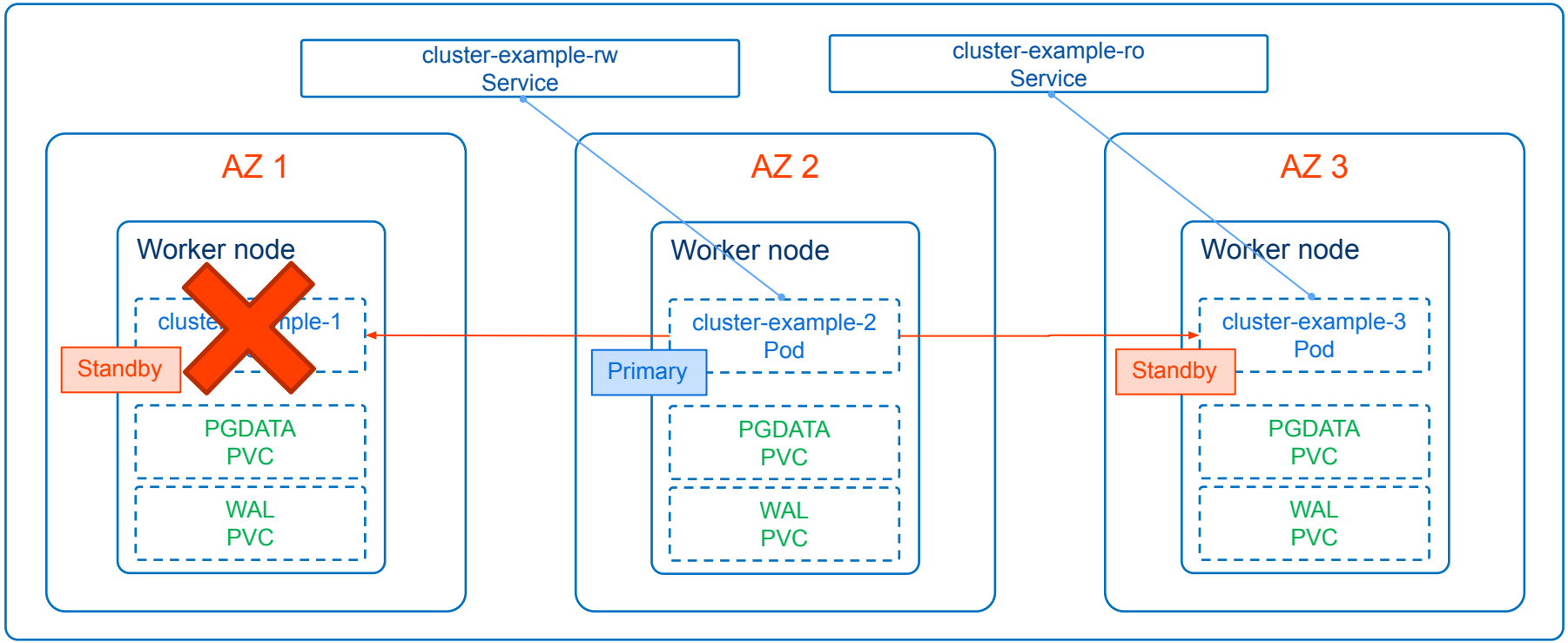


EDB

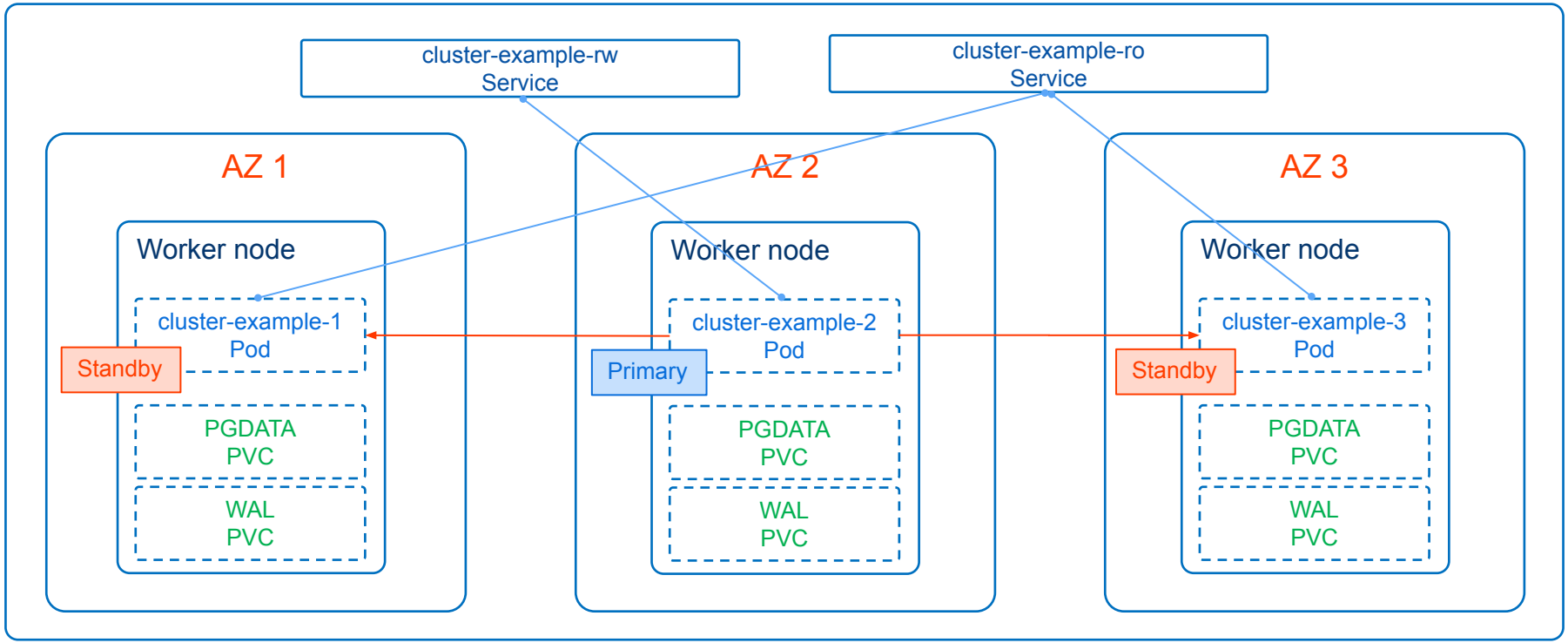
This is what happens under the hood



This is what happens under the hood



This is what happens under the hood



This demo is in

https://github.com/sergioenterprisedb/kubecon2022-demo

EDB

DEMO In yellow

Key capabilities

- Direct integration with Kubernetes API server for High Availability, without requiring an external tool
- Failover of the primary instance by promoting the most aligned replica
- Automated recreation of a replica
- Planned switchover of the primary instance by promoting a selected replica
- Scale up/down capabilities
- Definition of the *read-write* service, to connect your applications to the only primary server of the cluster
- Definition of the *read-only* service, to connect your applications to any of the instances for reading workloads
- Declarative management of PostgreSQL configuration, including certain popular Postgres extensions through the cluster spec: pg_audit, auto_explain, and pg_stat_statements
- Support for Local Persistent Volumes with PVC templates
- Reuse of Persistent Volumes storage in Pods
- Separate volume for WAL files
- Rolling updates for PostgreSQL minor versions
- In-place or rolling updates for operator upgrades
- TLS connections and client certificate authentication
- Support for custom TLS certificates (including integration with cert-manager)
- Continuous backup to an object store (AWS S3 and S3-compatible, Azure Blob Storage, and Google Cloud Storage)
- Backup retention policies (based on recovery window)
- Full recovery and Point-In-Time recovery from an existing backup in an object store
- Offline import of existing PostgreSQL databases, including major upgrades of PostgreSQL
- Parallel WAL archiving and restore to allow the database to keep up with WAL generation on high write systems
- Support tagging backup files uploaded to an object store to enable optional retention management at the object store layer Replica clusters for
- PostgreSQL deployments across multiple Kubernetes clusters, enabling private, public, hybrid, and multi-cloud architectures
- Support for Synchronous Replicas
- Support for HA physical replication slots at cluster level
- Connection pooling with PgBouncer
- Support for node affinity via nodeSelector
- Native customizable exporter of user defined metrics for Prometheus through the metrics port (9187)
- Standard output logging of PostgreSQL error messages in JSON format
- Automatically set readOnlyRootFilesystem security context for pods
- cnpg plugin for kubectl
- Fencing of an entire PostgreSQL cluster, or a subset of the instances
- Simple bind and search+bind LDAP client authentication
- Multi-arch format container images
- Postgres cluster hibernation

Sergio Romera

EDB – Senior Sales Engineer

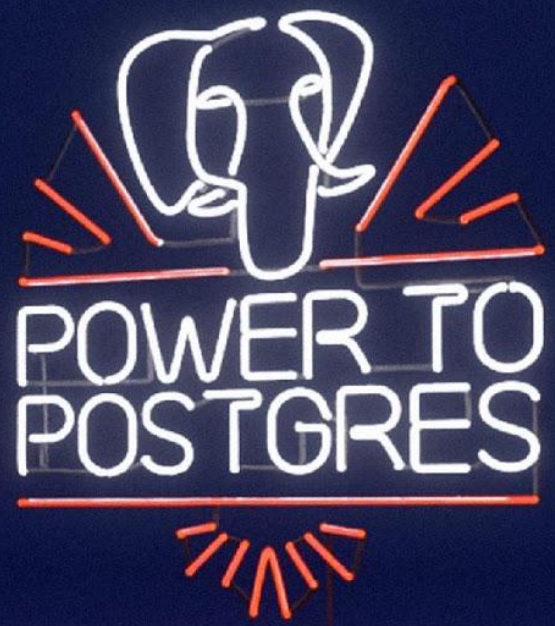
Oracle Cloud Infrastructure Architect
Oracle Autonomous Database Cloud Specialist
Database Administrator

AWS Cloud Practitioner

PostgreSQL 14 Essentials

Microsoft Azure Fundamentals
Microsoft Data Azure Fundamentals

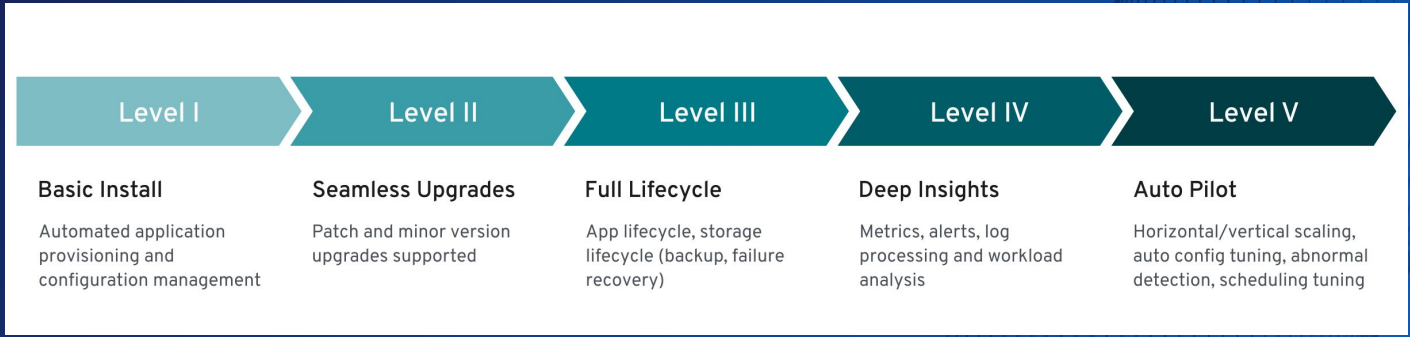
EDB Certified Associate
Postgres Advanced Server 12
EDB Certified Professional
Postgres Advanced Server 13



Thank you

EDB

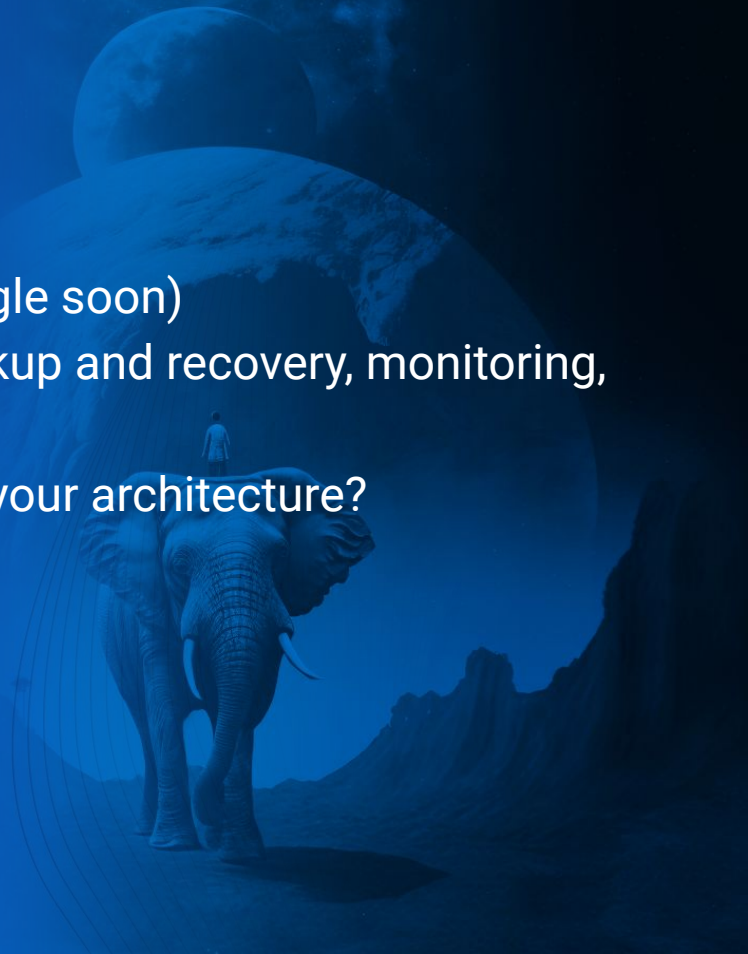
Operator Capabilities Levels



Source: https://sdk.operatorframework.io/docs/overview/operator-capabilities/

Contact EDB if you need:

- Support for PostgreSQL Opensource
- Oracle migrations to PostgreSQL
- Managed Postgres on Azure or AWS (and Google soon)
- Enterprise tools for Postgres (HA, failover, backup and recovery, monitoring, trainings, …)
- Do you need a workshop to better understand your architecture?



EDB