

# **Enterprise Scale Postgres on OpenShift for Intelligent Road Tolling**

**Deutsche Telekom / T-Systems & Crunchy Data**

**Dr Christoph Burandt | @ T-Systems**

**Karen Jex | Senior Solutions Architect @ Crunchy Data**

**Red Hat Summit Connect | Zurich | January 2025**

# Karen Jex

*Senior Solutions Architect*

25 Years in Database  
Administration

---



# Dr. Christoph Burandt



---

**T Systems** Let's power  
higher performance



# Agenda

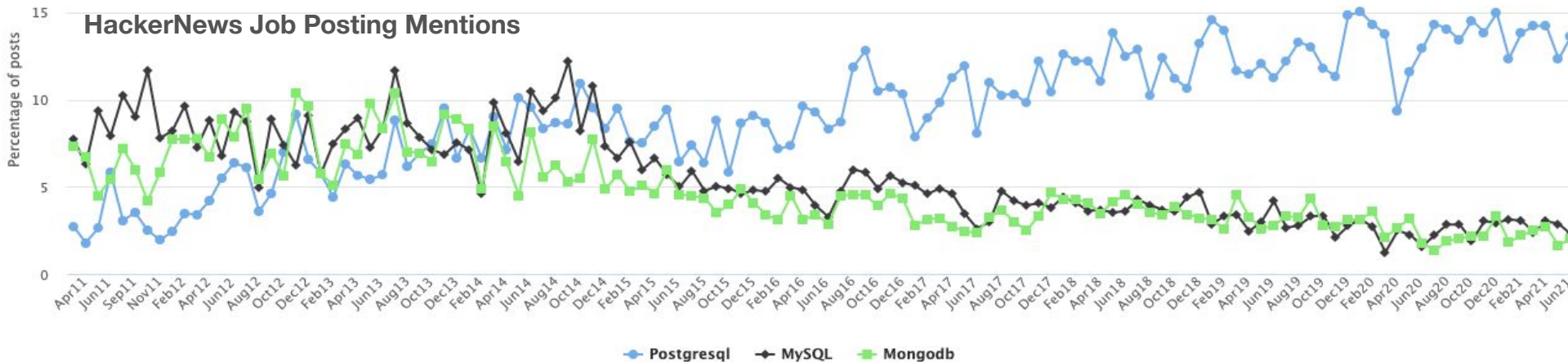
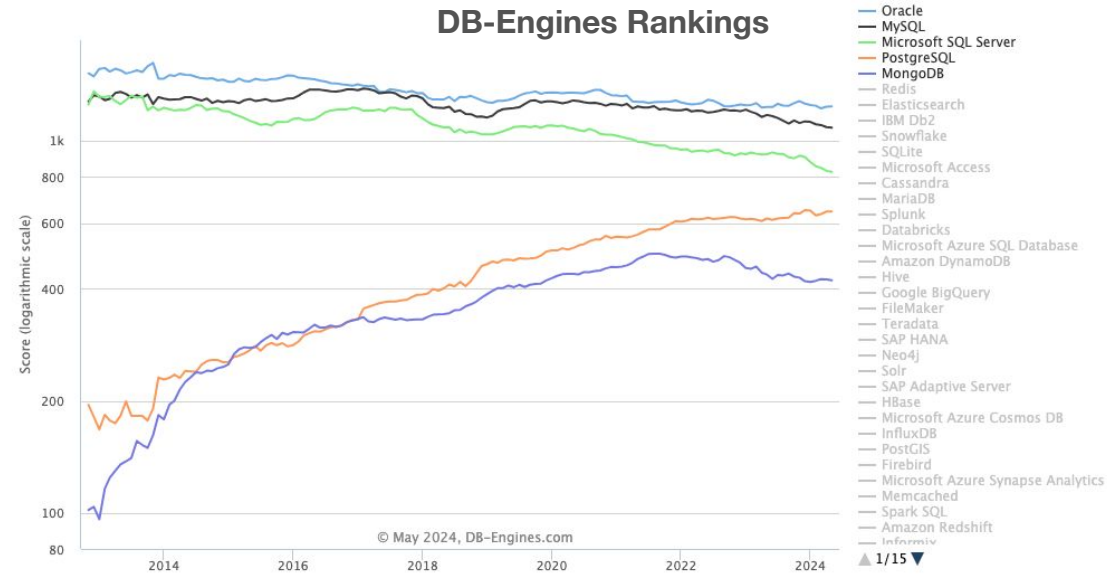
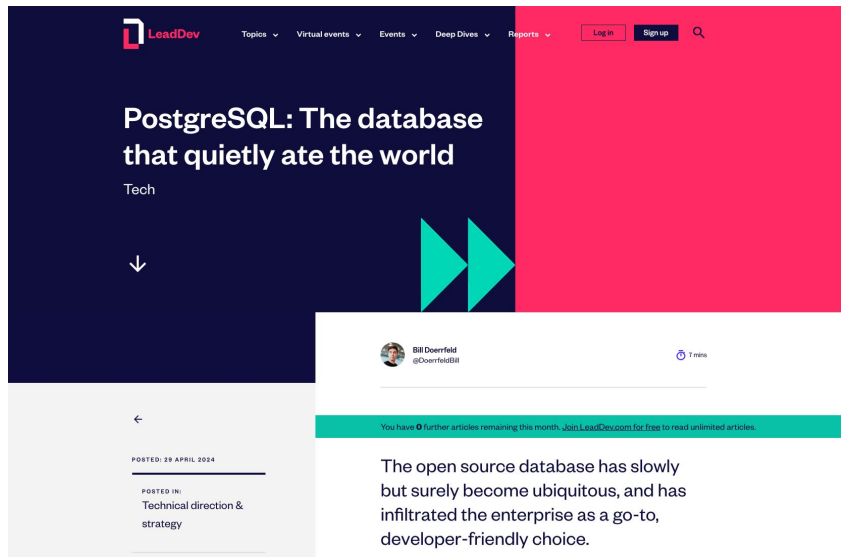
- 1 Postgres
- 2 Crunchy Data & Red Hat
- 3 CPK (Crunchy Postgres for Kubernetes)
- 4 Customer Experience
- 5 How does it Work?
- 6 Getting up and Running



# Agenda

- 1 Postgres
- 2 Crunchy Data & Red Hat
- 3 CPK (Crunchy Postgres for Kubernetes)
- 4 Customer Experience
- 5 How does it Work?
- 6 Getting up and Running

# PostgreSQL is a Growing Part of the Database Landscape



# Why Postgres

- ✓ Established, Reliable & Secure
- ✓ Feature Rich
- ✓ Extensibility
- ✓ No Central Owner
- ✓ Hiring
- ✓ 35+ year evolution

## The Technical Details

- Datatypes
- Transactional DDL
- Foreign Data Wrappers
- Concurrent Index Creation
- Conditional indexes
- JSON
- Common Table Expressions
- Fast column addition
- Listen/Notify
- Upsert
- Partitioning
- Per transaction sync replication
- Window function
- JSON/JSONB
- Continued innovation



# Agenda

- 1 Postgres
- 2 **Crunchy Data & Red Hat**
- 3 CPK (Crunchy Postgres for Kubernetes)
- 4 Customer Experience
- 5 How does it Work?
- 6 Getting up and Running



# Postgres Anywhere

## BARE METAL, VMS, CLOUD

---

### Crunchy Postgres

Crunchy Certified PostgreSQL is production ready Postgres.

#### INCLUDES:

- Backups
- Disaster recovery
- High availability
- Monitoring
- Automation
- Self managed

## KUBERNETES

---

### Crunchy Postgres for Kubernetes

Cloud Native Postgres on Kubernetes powered by Crunchy Postgres Operator.

#### INCLUDES:

- Simple provisioning
- Backups and DR included
- High availability
- Seamless upgrades
- Scale from 1 to thousands of databases
- Self managed

## FULLY MANAGED CLOUD

---

### Crunchy Bridge

The fully managed Postgres option on your choice of Cloud provider.

#### INCLUDES:

- AWS, Azure or GCP
- Continuous protection
- Backups
- Point in Time Recovery
- Pay for what you use
- The developer experience you want

# Your Partner for Postgres

## AI

---



Stored RAG Embeddings to  
integrate with you favorite LLM

## Analytics

---



Query your data lake directly  
from Postgres

## Kubernetes

---



Production Grade Postgres in  
Kubernetes

**Come visit us at our booth to learn more...**

# Crunchy Data + Red Hat

**Advanced PostgreSQL Solutions for Leading Red Hat Technology including:**



Crunchy Data & OpenShift



Level 5 Certified PostgreSQL Operator



Crunchy Data & Ansible Automation Platform



Crunchy Data with Quay & Clair





# Agenda

- 1 Postgres
- 2 Crunchy Data & Red Hat
- 3 CPK (Crunchy Postgres for Kubernetes)**
- 4 Customer Experience
- 5 How does it Work?
- 6 Getting up and Running

# Enterprise Scale Postgres



## Backups

Ensure your critical enterprise data isn't at risk



## Disaster recovery

The ability to quickly and reliably recover from data corruption or loss.



## High availability

With high availability you can trust your database is online.



## Connection scaling

With built in connection scaling you can easily scale to tens of thousands of connections for your database.



## Monitoring

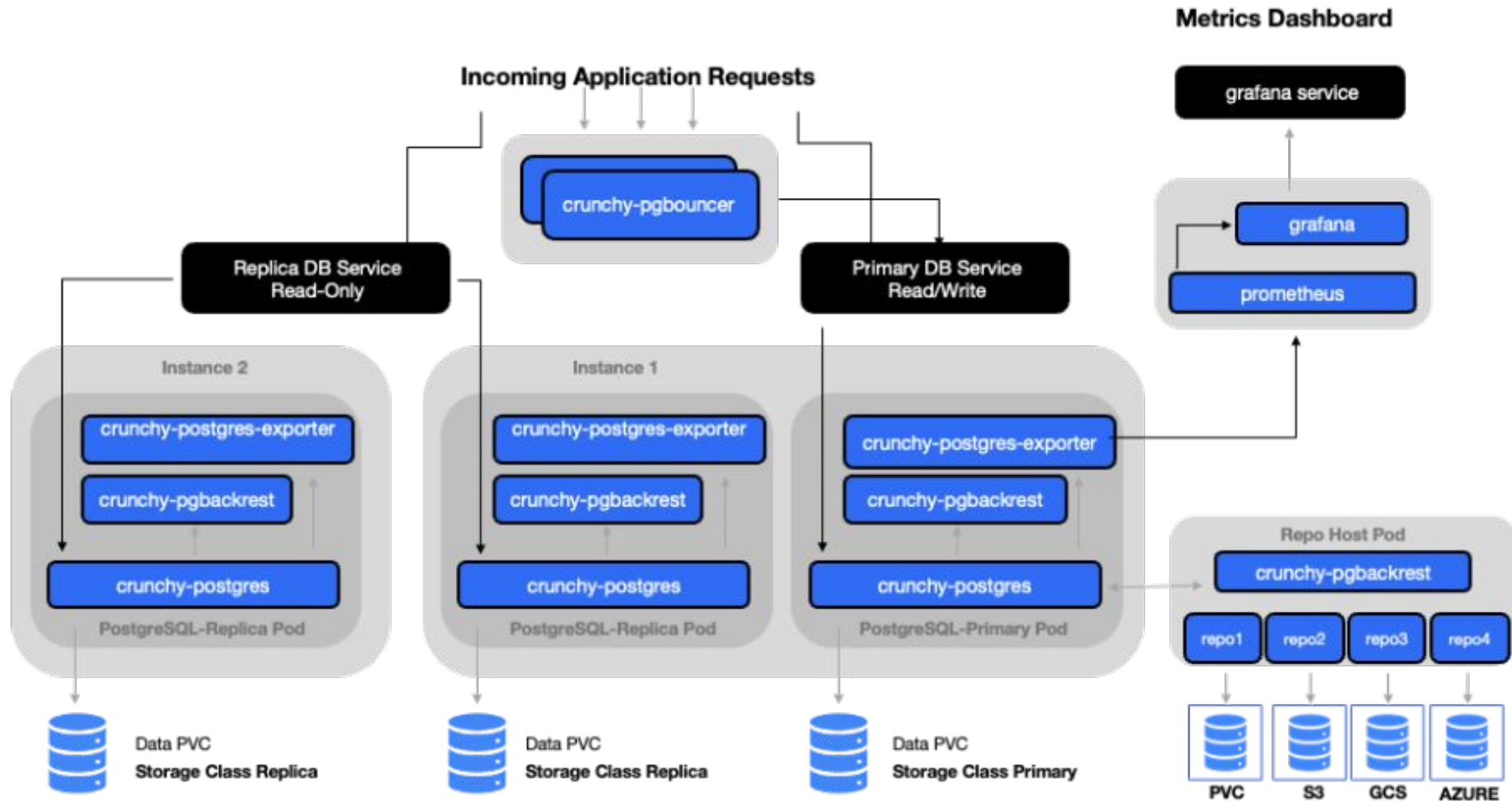
Ensure your database is properly resourced so your application performs well



## Easy Upgrades

Keeping pace with security fixes and version updates is critical.

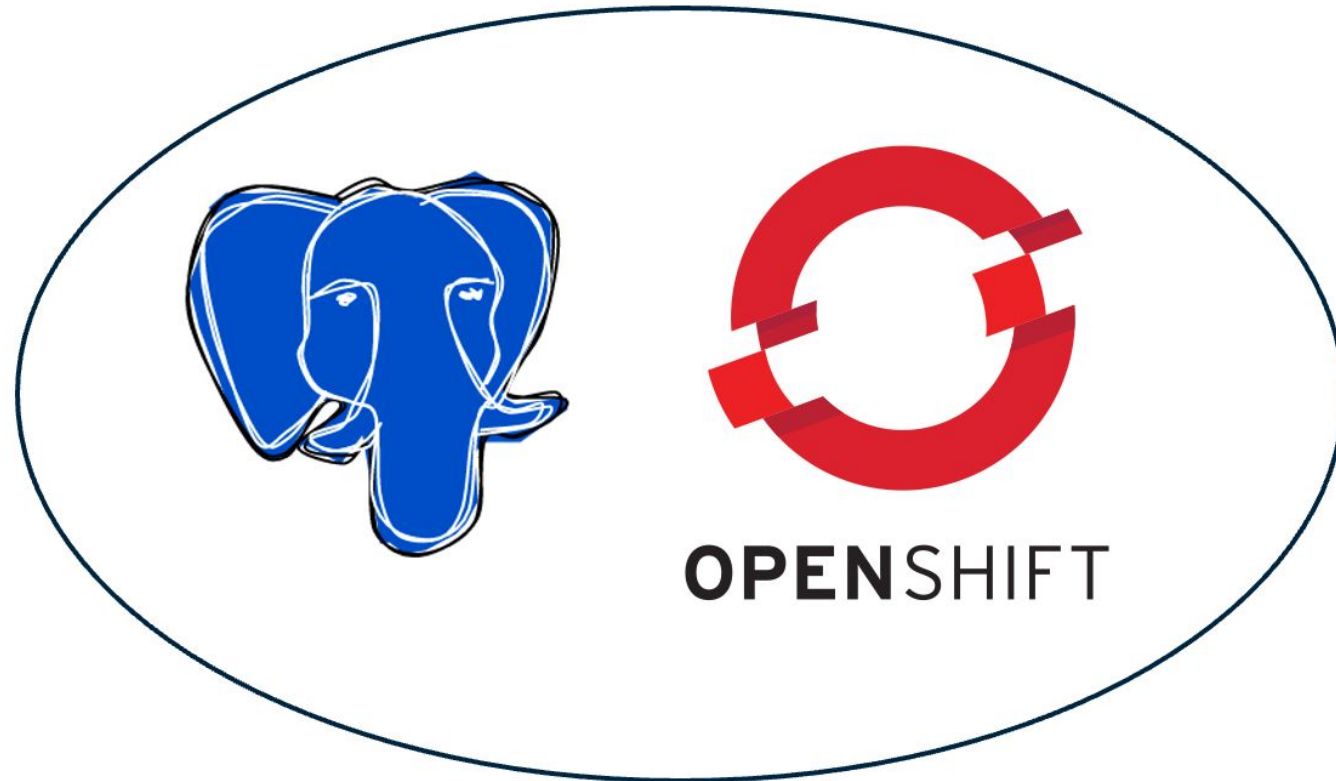
# Enterprise Scale Postgres



# Why Postgres on OpenShift?

- Automation
- Deploying at scale
- Multi-tenancy
- Microservices
- OpenShift already in use

# Crunchy Postgres for Kubernetes





# Crunchy Postgres for Kubernetes

- Enterprise scale PostgreSQL on OpenShift
- Combined strength of OpenShift and Postgres
- Robust, secure, scalable architecture
- Virtual database administrator



# Agenda

- 1 Postgres
- 2 Crunchy Data & Red Hat
- 3 CPK (Crunchy Postgres for Kubernetes)
- 4 Customer Experience**
- 5 How does it Work?
- 6 Getting up and Running

# 4 years of experience running Crunchy Postgres Cluster on Open Shift

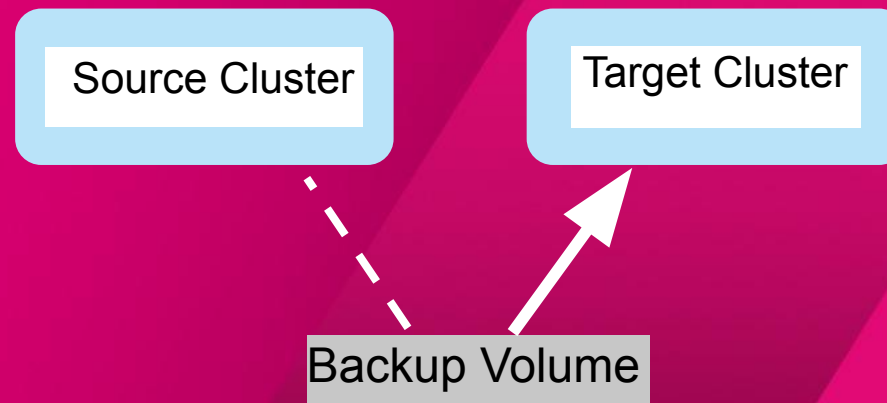
Dr. Christoph Burandt 15.01.2025

# Agenda

- 1 In 2021 first Crunchy Cluster running on OpenShift
- 2 Operating Postgres/Crunchy Cluster in Version 4
- 3 In 2023 start Migration from pgo 4 to 5
- 4 Operating Postgres/Crunchy Cluster in Version 5
- 5 Updates and Upgrades

# 1 In 2021 first Crunchy-Cluster running on OpenShift

Takeover Data from existing Customer Crunchy Cluster using Export / Import and Reassignment of Backup Volume in newly created Cluster to another Namespace



Due to intensive LOB Usage – PVC growth -> Solution with implementation of a function calling vacuumlo managed by policy schedule -> deleting orphan Lobs (Crunchy Data Support)

## 2 Operating Postgres/Crunchy Cluster in Version 4

- In general each Crunchy Postgres Cluster is located in a separated Namespace
- Reducing from 3 Replica per Cluster to 1 Replica per Cluster
- Extension of PVC size using graphical Interface and pg config : `oc edit pgcluster`
- Configure Monitoring : PrometheusRules -> Incidents by self-developed Open Shift Ticket Creator
- Configure Parameters `oc edit configmap` or `pgo update cluster .....`
- Creating users : `pgo create user...`
- Configure Backups : `pgo create schedule ...`

..... Most actions had to be done using **PGO** – Operator or **OC** - commands

### 3 In 2023 start Migration from pgo 4 to 5

#### Method:

Migrate from Backup -> Delete Cluster using Option : --keep-backups  
Create new cluster using data restore from old Cluster Backup

**Downtime : applications offline to ensure all the data is migrated.**

Most important change from PGO Version 4 to 5 : **Manifest :**

```
apiVersion: postgres-operator.crunchydata.com/v1beta1
kind: PostgresCluster
metadata:
  name: dbulatest-db
spec:
  openshift: true
  image:
    image-registry.openshift-image-registry.svc:5000/dt-test-pg-latest/crunchy-postgres:ubi8-15.8-5.6.1-0
  postgresVersion: 15
  instances:
    - name: instance1
      replicas: 3
```

## 4 Operating Postgres/Crunchy Cluster in Version 5

- Many old Skripts using **pgo / oc** - commands
  - > replaced by Manifest Changes and simple oc apply ...
  - > But for Data Security some manual Tasks are still necessary
    - > ex. : drop user / drop database ...
- Changing Values in Manifest -> easier – relax 😊 , maintain old Manifest-Versions !



## 5 Updates and Upgrades

### Updates (Operator and Cluster):

Our individual Patchcycle: 2 x times a year Minor Crunchy Update and PostgreSQL Update

Steps:

**1. Operator Update :**

create new Source files (YAML)

tag new Operator Image

oc apply... and wait until Operator Pod is running

**2. Cluster Update (Rolling):**

tag of all new Cluster related Images

change Manifest : postgres.yaml

oc apply .....

**3. Check** if all pods are running and ....

Tag - Example: `oc tag <source_namespace>/crunchy-postgres:ubi8-15.8-5.6.1-0  
<target_namespace>/crunchy-postgres:ubi8-15.8-5.6.1-0`

Take care : References must match to Operator's Image List

## 5 Updates and Upgrades

Upgrade :

Example: Separated Major Postgres Upgrade (Version 12->15 ) -> Downtime needed

Steps:

1. Backup
2. Shutdown Cluster
3. Upgrade
4. Restart Cluster with new version
5. Post-Upgrade Tasks

Images: crunchy-upgrade-Image + new crunchy-postgres-Image needed:

`crunchy-upgrade:ubi8-5.6.1-0`

`crunchy-postgres:ubi8-15.8-5.6.1-0`

Link to CrunchyData Image for Info of PGO Version 5.6:

[https://access.crunchydata.com/downloads/browse/containers/postgres-operator/releases/redhat/pgo\\_5.6/pgo\\_5.6.1.txt](https://access.crunchydata.com/downloads/browse/containers/postgres-operator/releases/redhat/pgo_5.6/pgo_5.6.1.txt)



# Agenda

- 1 Postgres
- 2 Crunchy Data & Red Hat
- 3 CPK
- 4 Customer Experience
- 5 How does it Work?**
- 6 Getting up and Running

```
apiVersion: postgres-operator.crunchydata.com/v1beta1
kind: PostgresCluster
metadata:
  name: zurich
spec:
  postgresVersion: 16
  instances:
    - name: instance1
      replicas: 3
      dataVolumeClaimSpec:
        accessModes:
          - "ReadWriteOnce"
      resources:
        requests:
          storage: 1Gi
```

High Availability

Backup & Recovery

Security

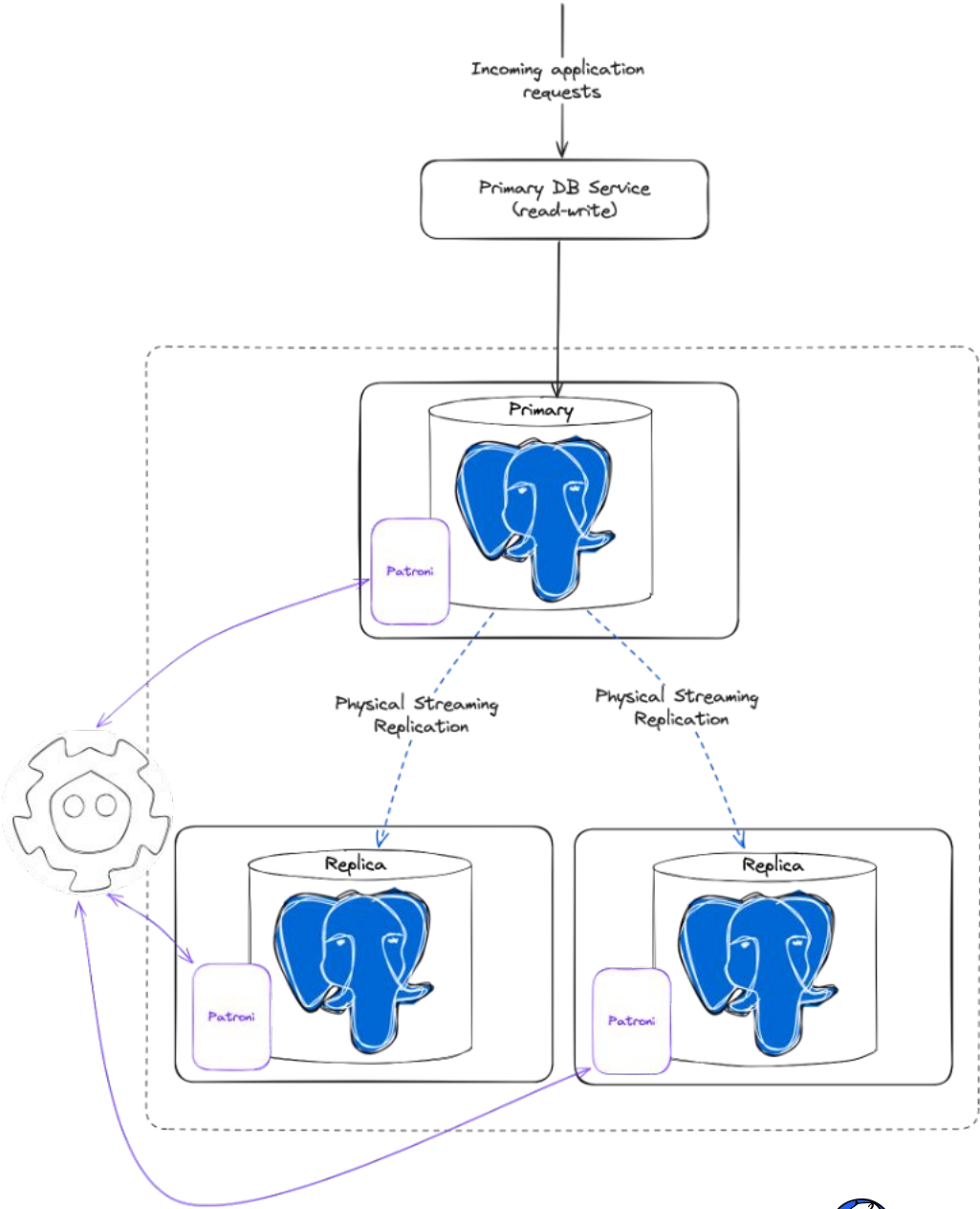
Monitoring

Install & Upgrade

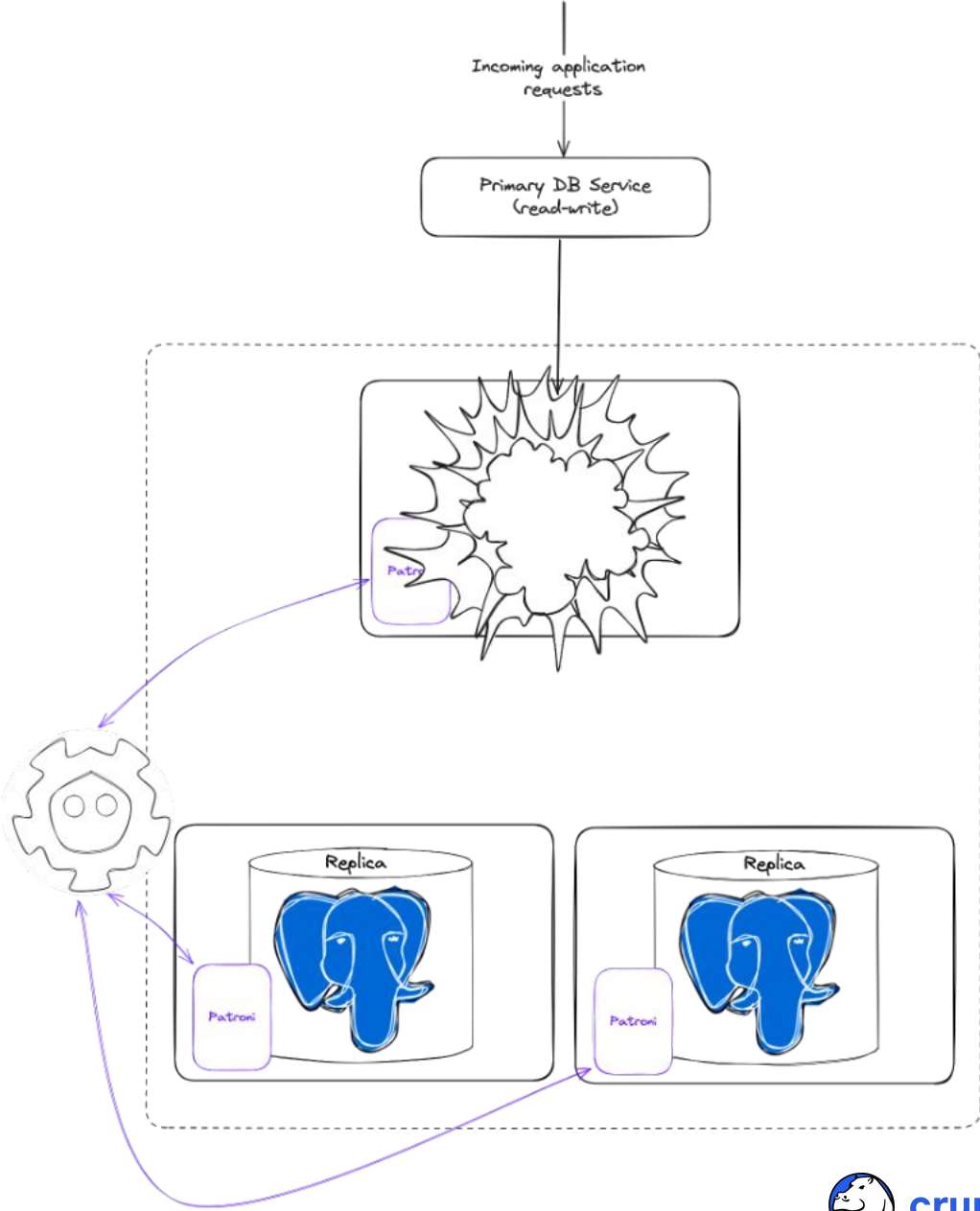
Provisioning & Scaling

# “DBA” Tasks

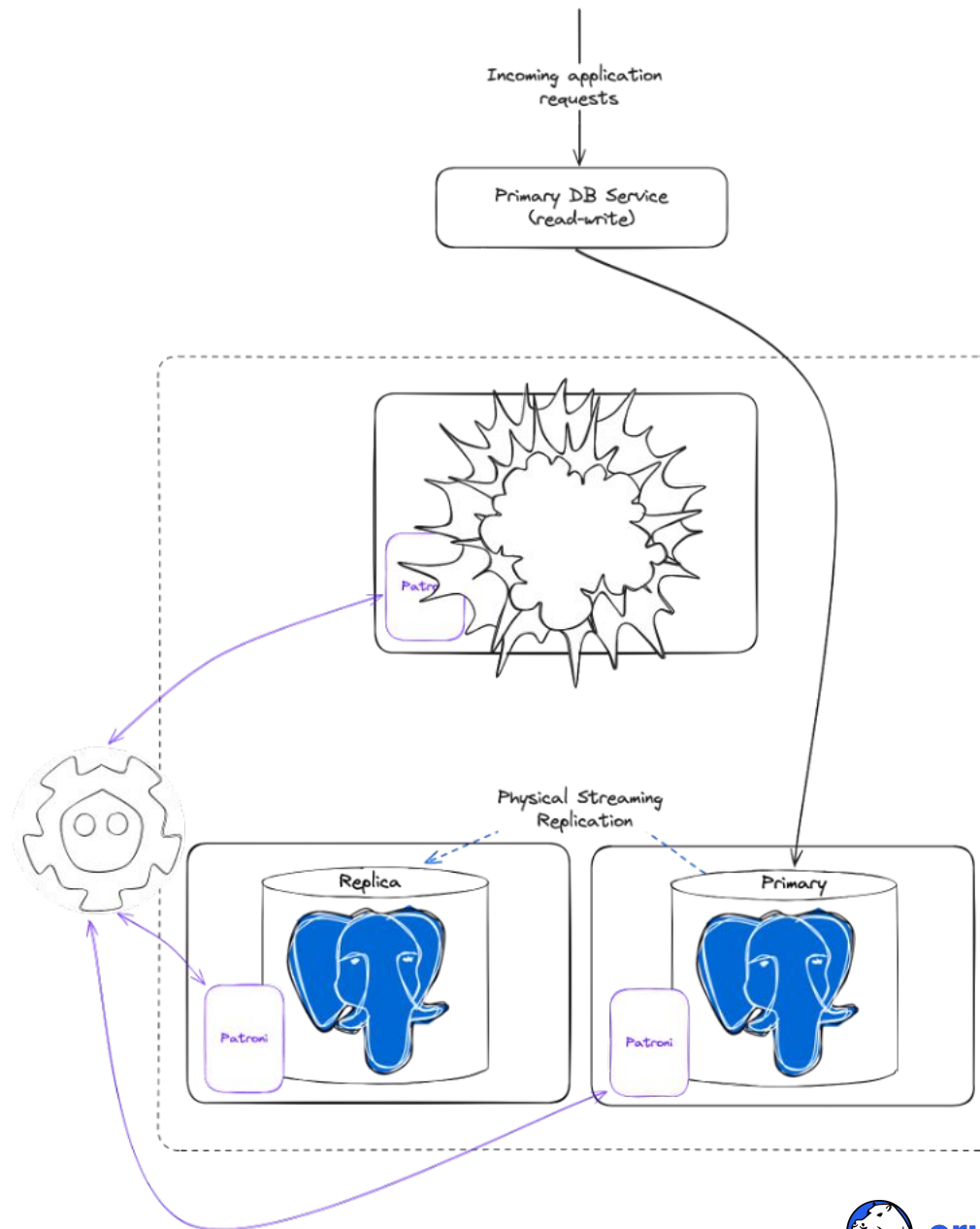
# CPK: High Availability



# CPK: High Availability

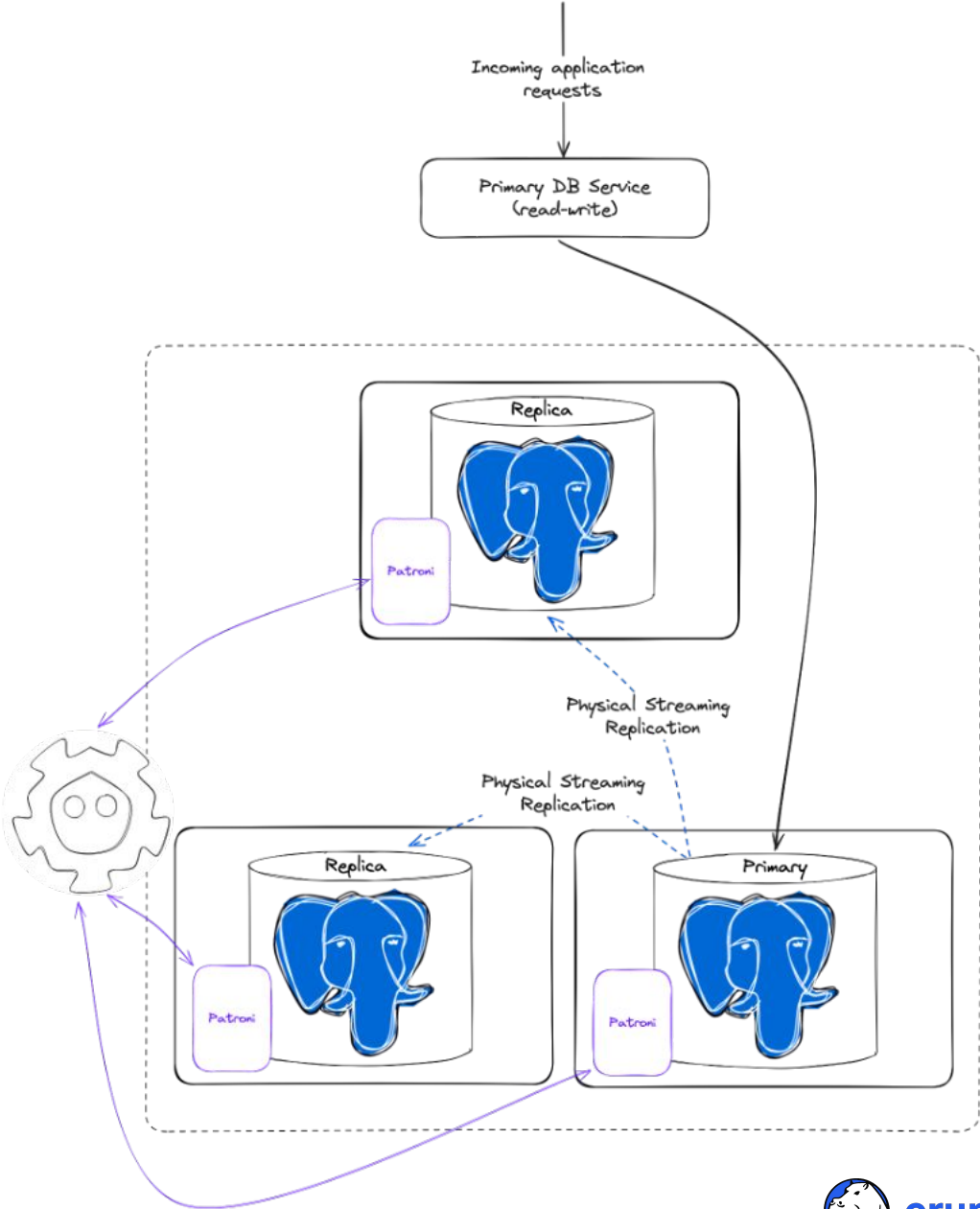


# CPK: High Availability





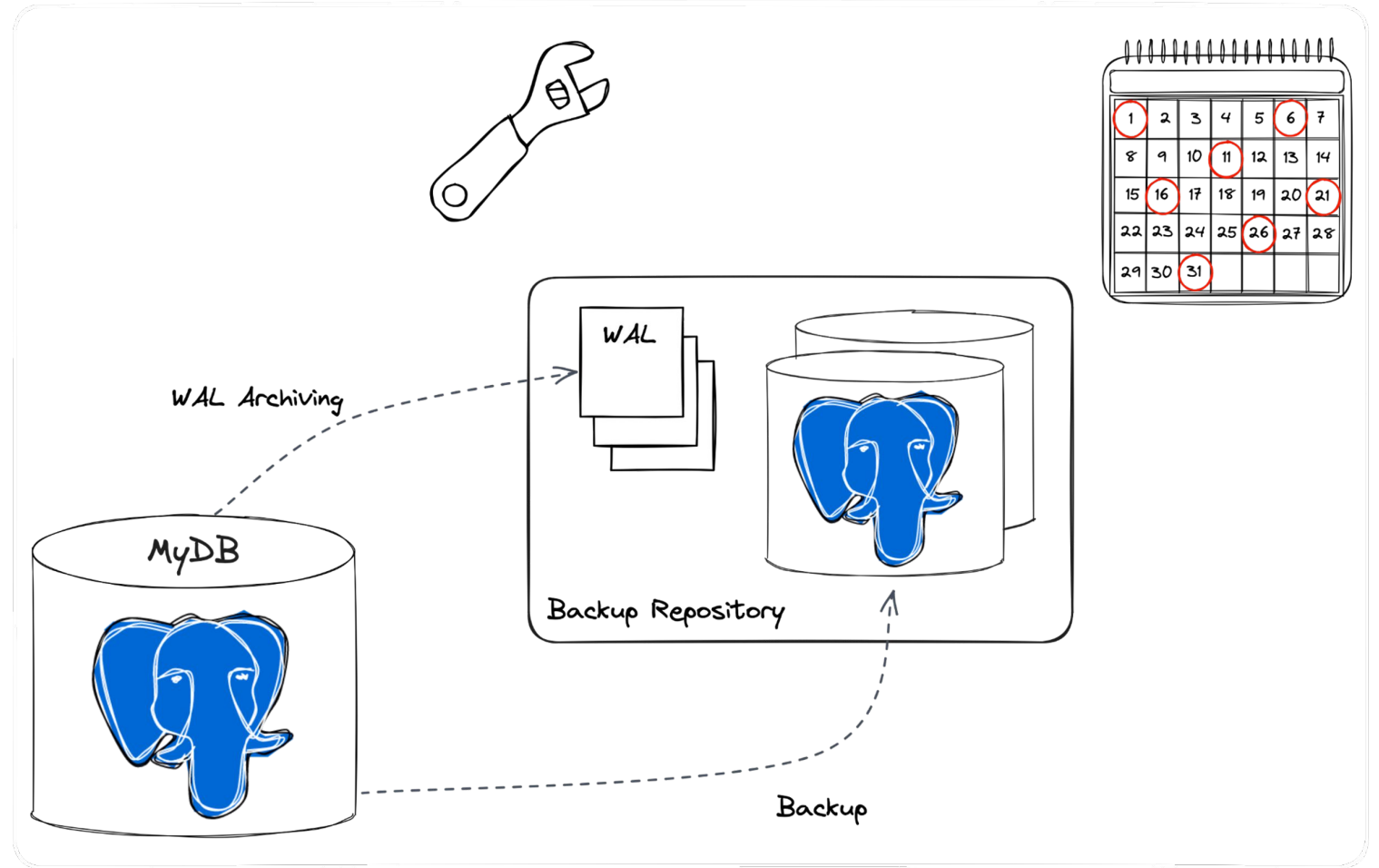
# CPK: High Availability



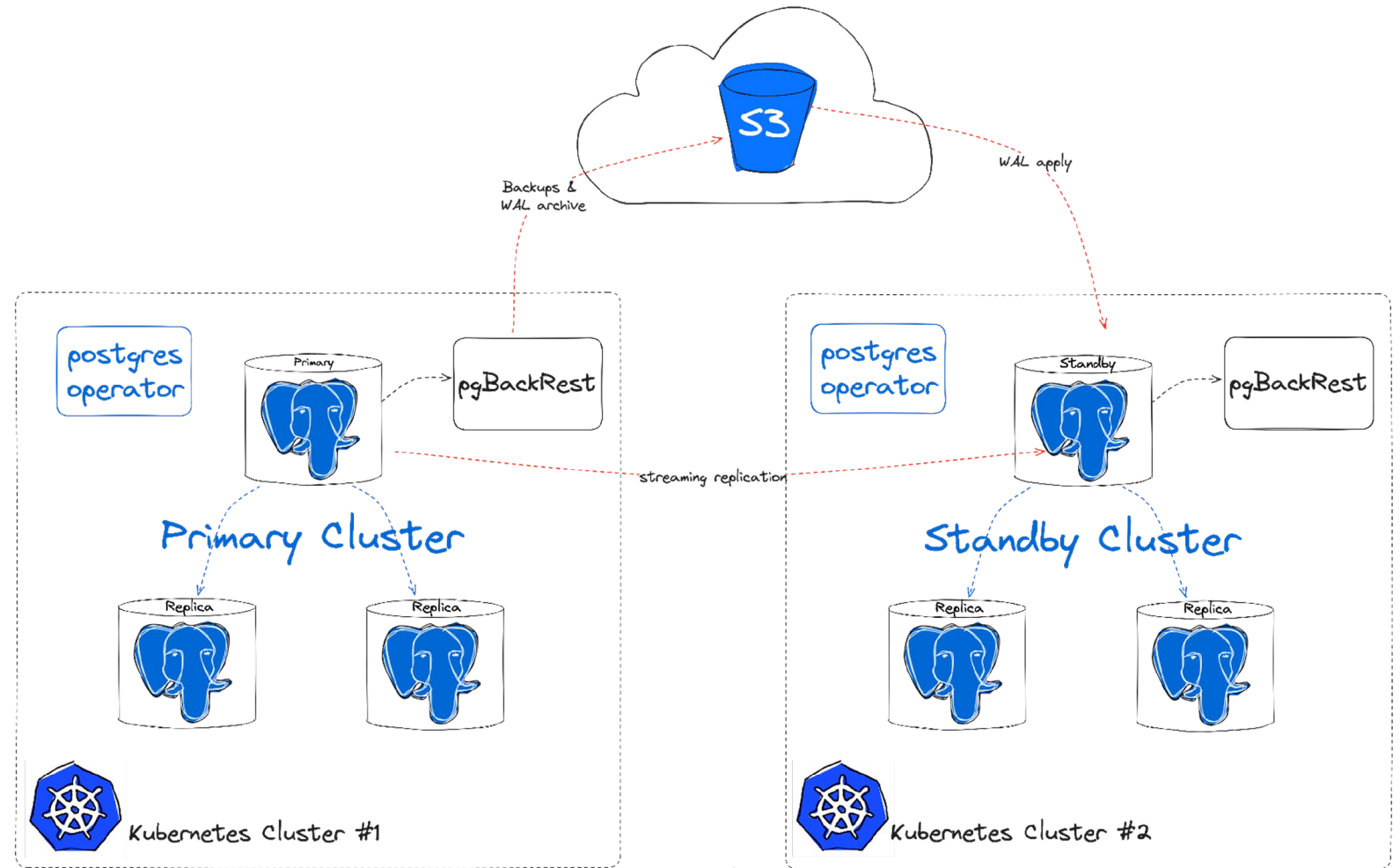
**CPK:**  
High  
Availability

- OpenShift
- Operator (PostgresCluster)
- Patroni

# CPK: Backup & Recovery



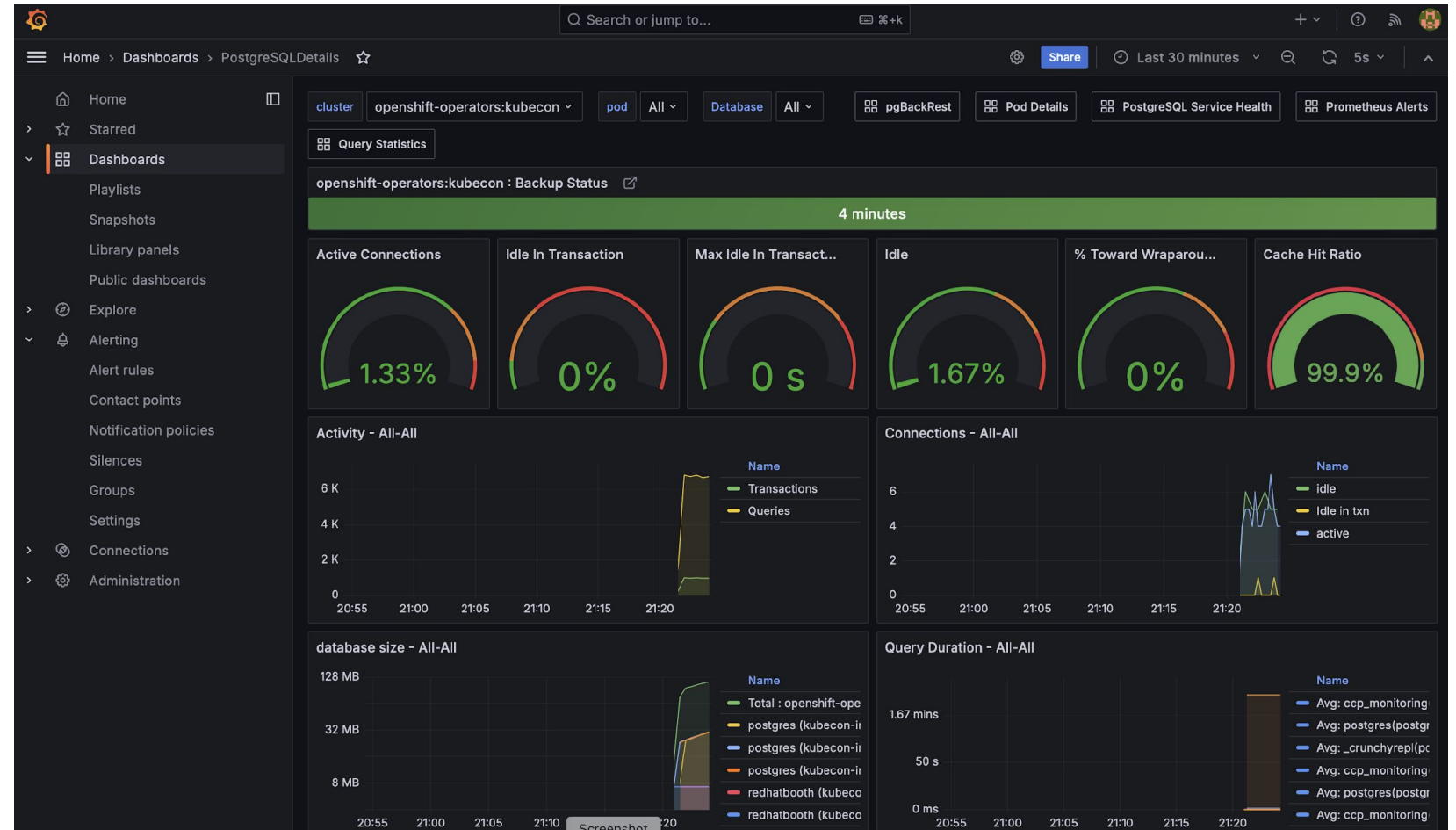
# CPK: Backup & Recovery



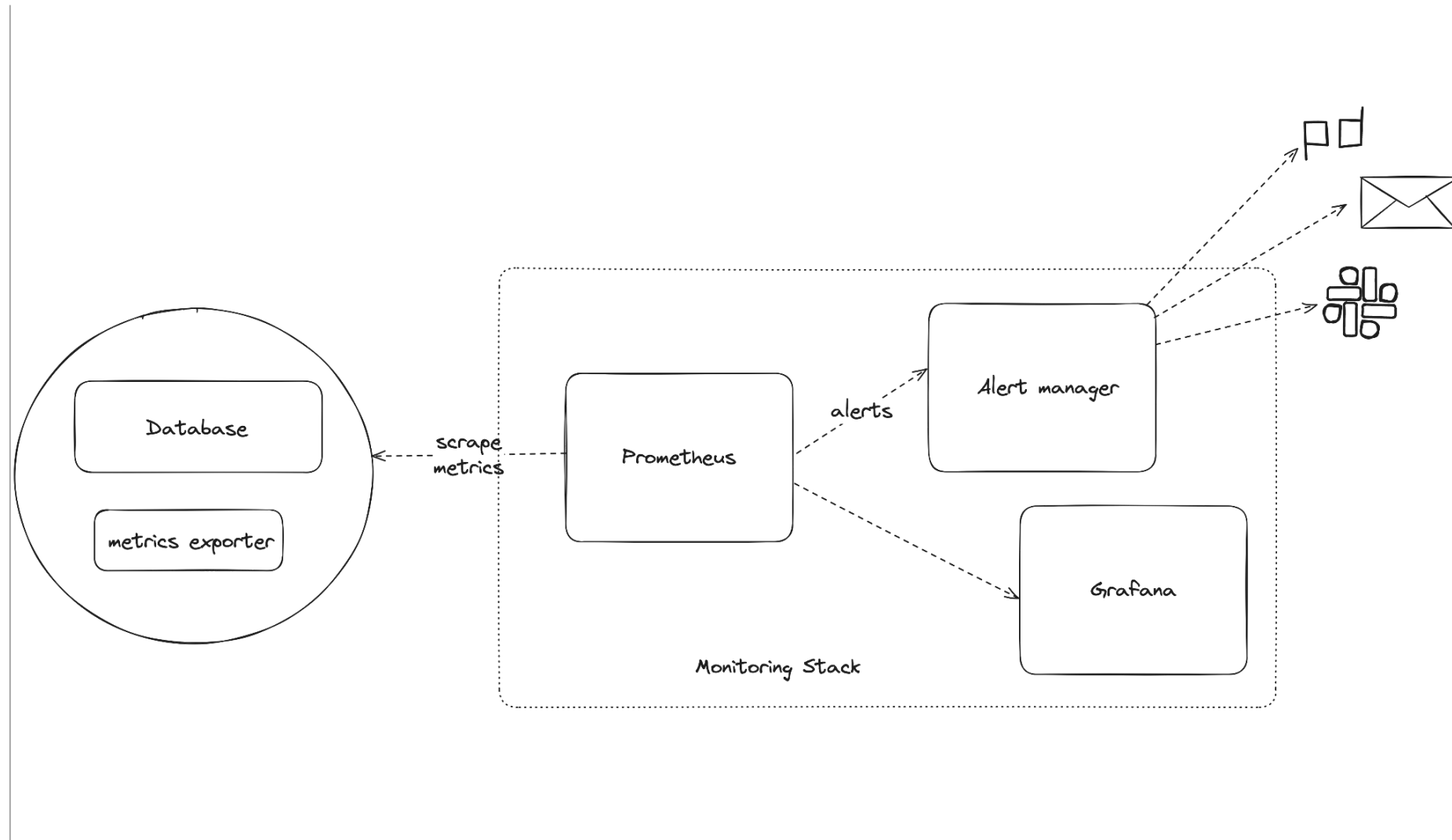
## CPK: Security

- Database Access
- Password Encryption
- Secrets
- SSL/TLS
- Certificates

# CPK: Monitoring

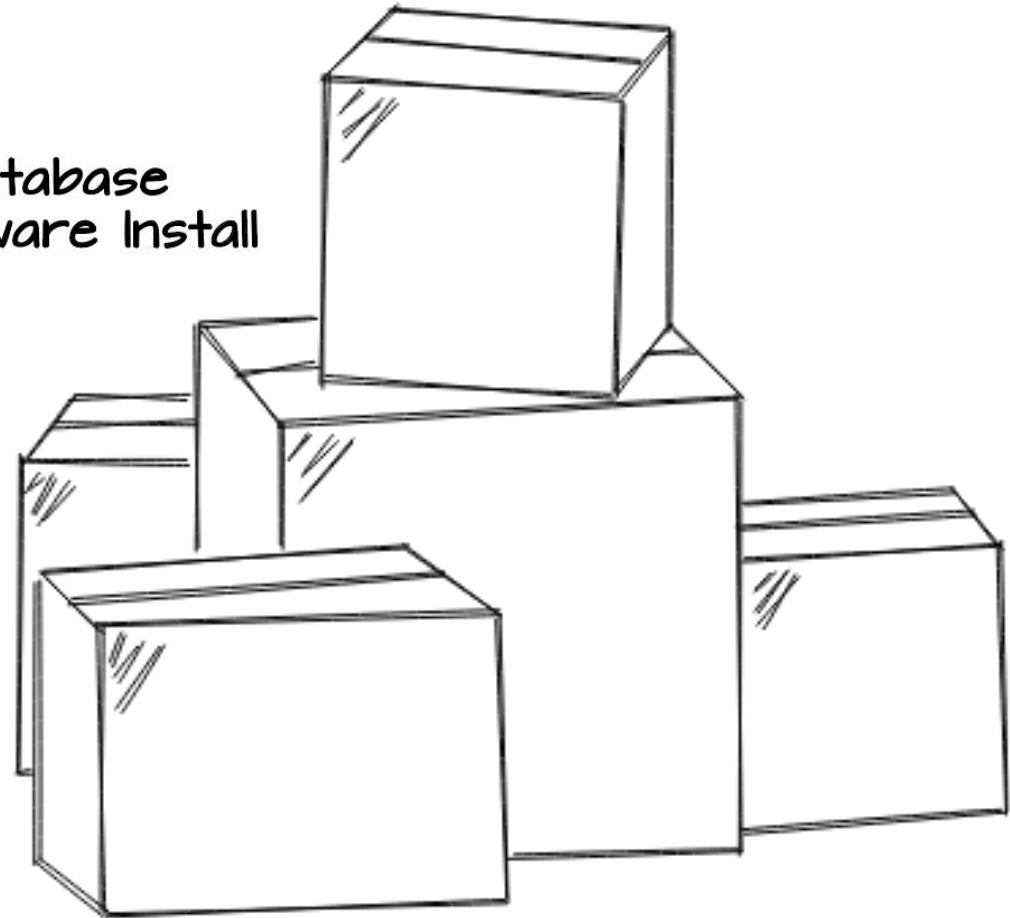


# CPK: Monitoring



**CPK:**  
Database  
Software Install  
& Upgrade

Database  
Software Install

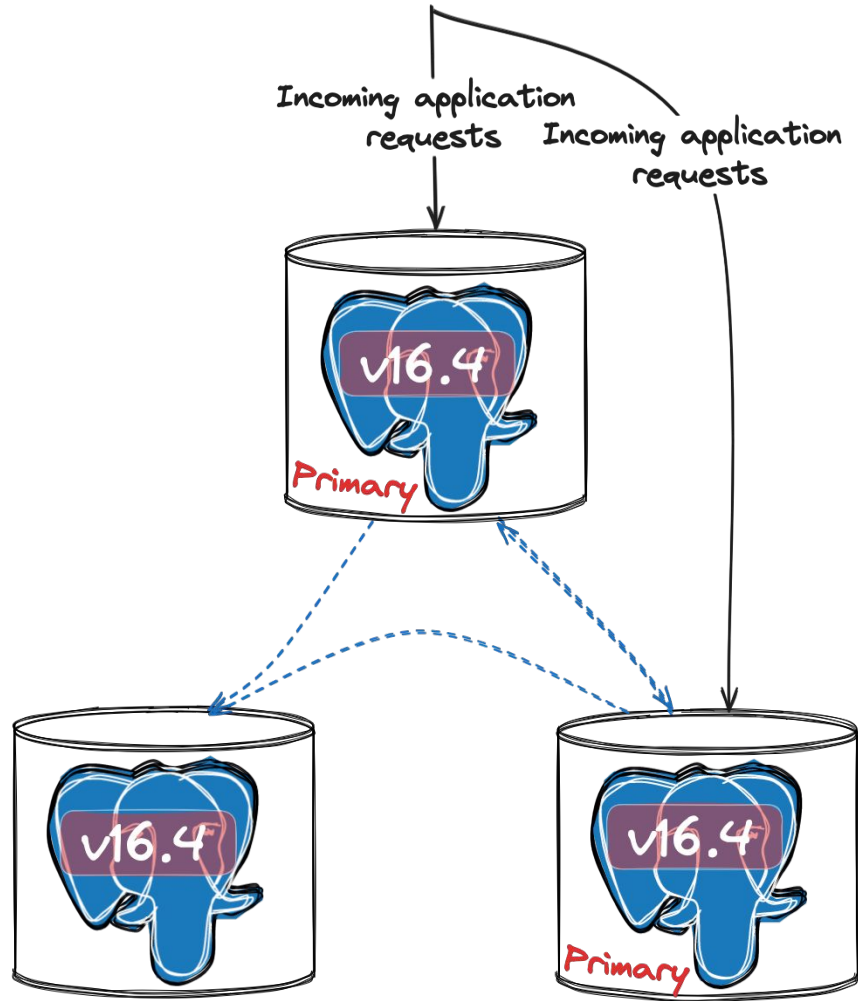




# CPK: Database Software Install & Upgrade

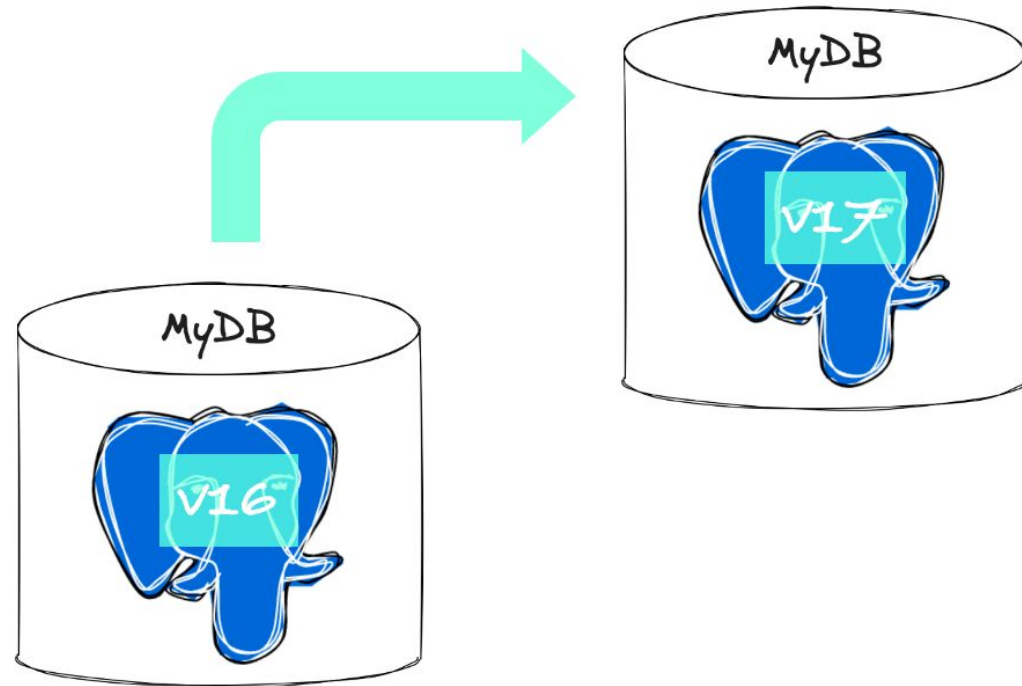
## Postgres Minor Upgrade

Ordered, automated rolling updates



**CPK:**  
Database  
Software Install  
& Upgrade

**Postgres Major Upgrade**  
Automated by Operator



# CPK: Scaling

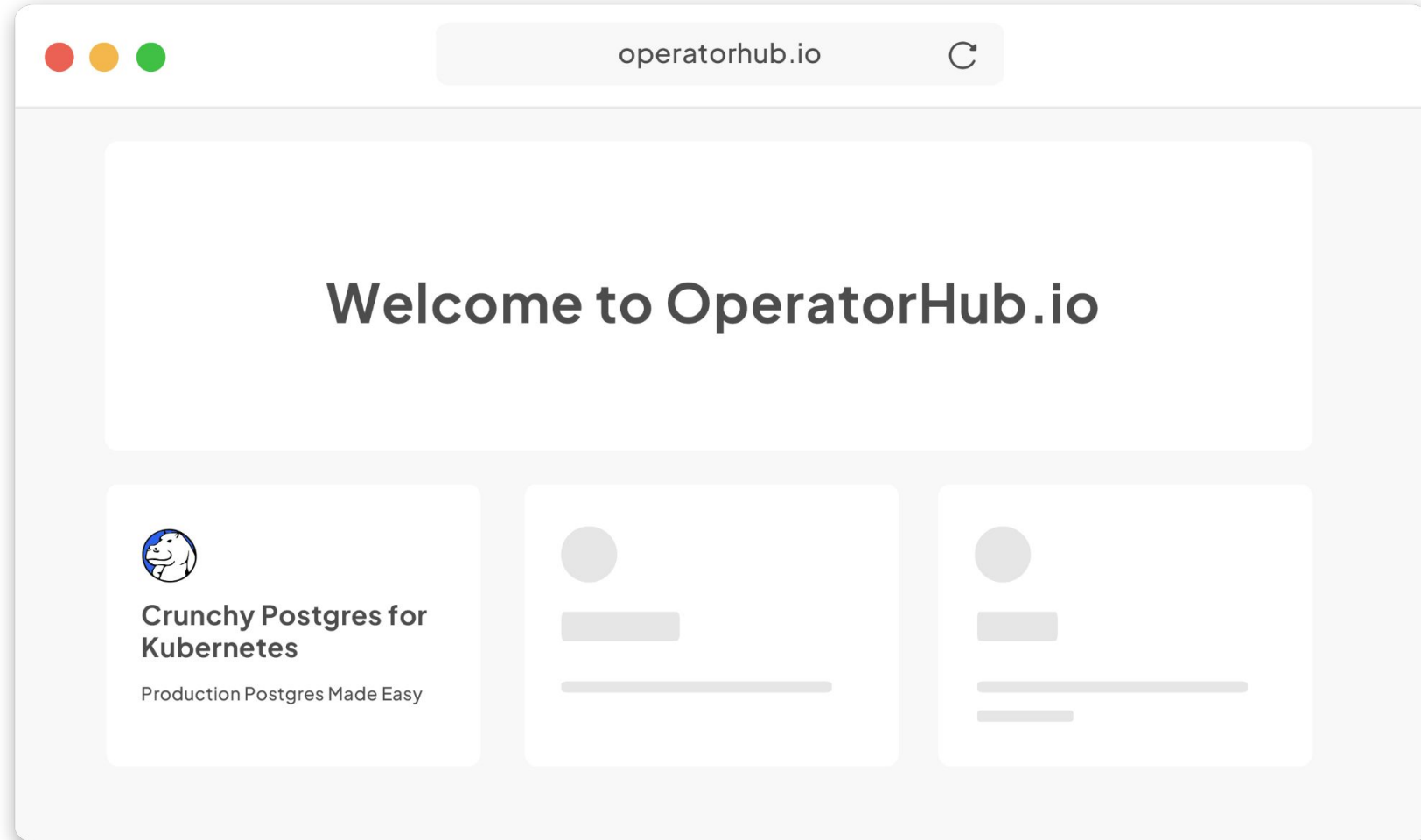
```
dataVolumeClaimSpec:  
  accessModes:  
  - "ReadWriteOnce"  
  resources:  
    requests:  
      storage: 1Gi  
      storage: 10Gi
```



# Agenda

- 1 Postgres
- 2 Crunchy Data & Red Hat
- 3 CPK
- 4 Customer Experience
- 5 How does it Work?
- 6 Getting up and Running

# Easy Button



# Crunchy Postgres for Kubernetes

- Enterprise scale PostgreSQL on OpenShift
- Combined strength of OpenShift and Postgres
- Robust, secure, scalable architecture
- Virtual database administrator

A white wireframe elephant is positioned on the left side of the slide, facing right. The elephant is composed of a network of interconnected lines forming a mesh that outlines its body, head, and trunk. The background is a solid, vibrant blue.

# Thank You!

Dr Christoph Burandt | @ T-Systems

Karen Jex | Senior Solutions Architect @ Crunchy Data

<https://www.linkedin.com/in/karenhjex>