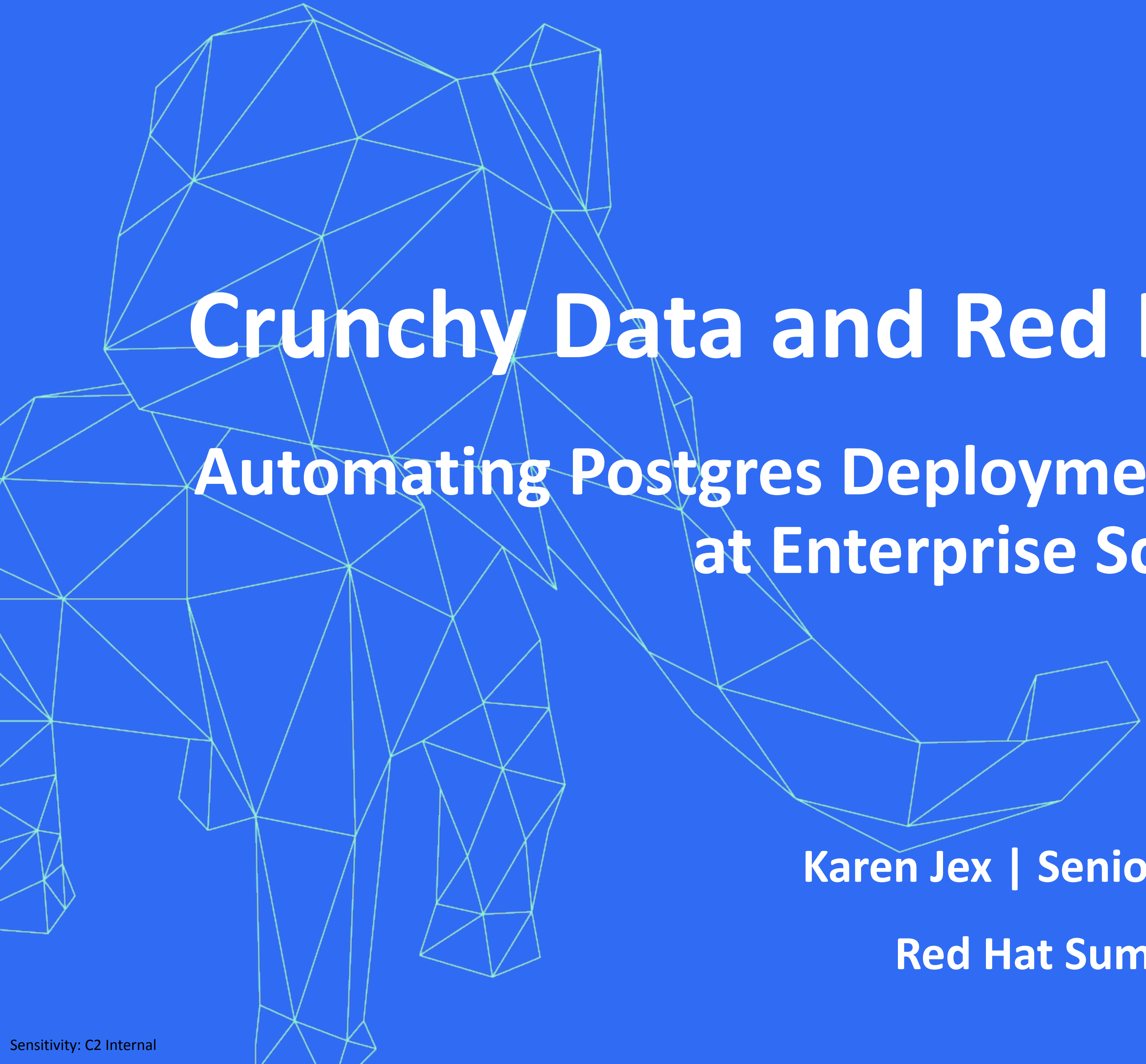


Crunchy Data and Red Hat OpenShift:

Automating Postgres Deployment & Administration at Enterprise Scale

Karen Jex | Senior Solutions Architect @ Crunchy Data

Red Hat Summit: Connect | Zurich | January 2024

Agenda

- Why Crunchy Data?
- Why Postgres on OpenShift?
- Whoa, that sounds complicated!
- What is Crunchy Postgres for Kubernetes?
- Create a High Availability PostgresCluster

Why Crunchy Data

- Major Contributor to Postgres
- World Class Technical Talent
- Enterprise Focus and Go to Market Approach
- 24 x 365 Support delivered by Skilled Postgres Engineers
- Commitment to Security Conscious Enterprises
- Leader in Postgres Technology for Kubernetes
- Certified Open Source Postgres Distribution

crunchy data

Crunchy Data + Red Hat

Crunchy Data is a Global Technology Partner with Certified PostgreSQL Technology for a Variety of Platforms

Crunchy Data can help **Red Hat** Customers confidently deploy **PostgreSQL** as an **alternative to legacy technologies** ensuring the Red Hat Customers continue to rely on trusted technology.

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**

crunchydata

SDX is a Crunchy Data Customer for 4+ Years

We will contribute during this session as to how SDX is successful using the Crunchy Data Postgres Operator to run a complex, distributed DLT (Digital Ledger Technology) Network

Janurary 2024

Oliver Kuepfer - oliver.kuepfer@sdx.com

a SIX company

Why Postgres

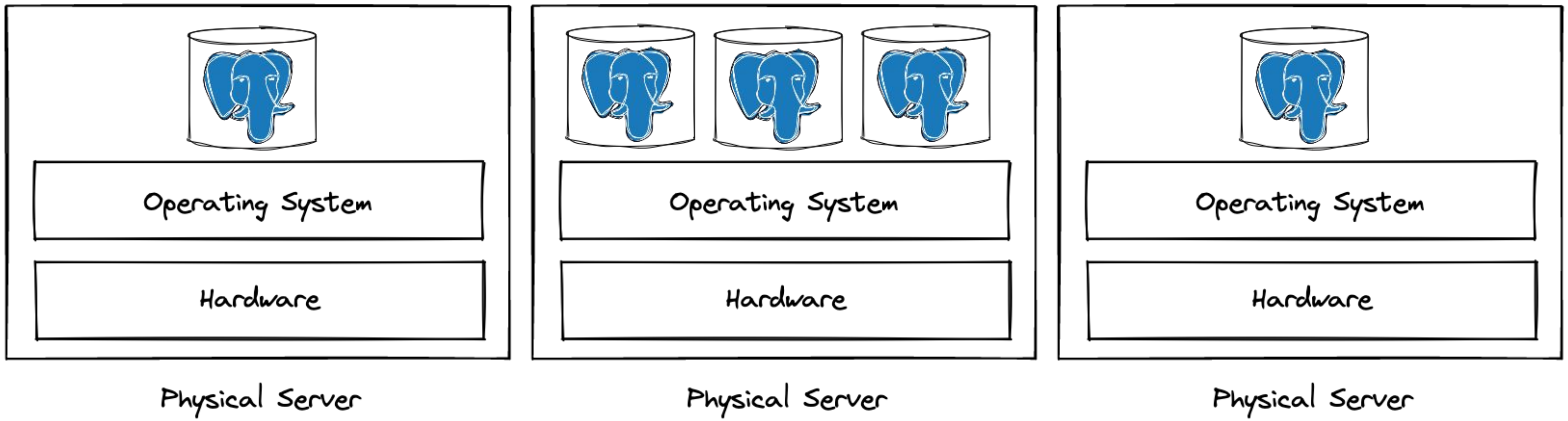
- ✔ Established, Reliable & Secure
- ✔ Feature Rich
- ✔ Extensibility
- ✔ No Central Owner
- ✔ Hiring
- ✔ 35+ year evolution

The Technical Details

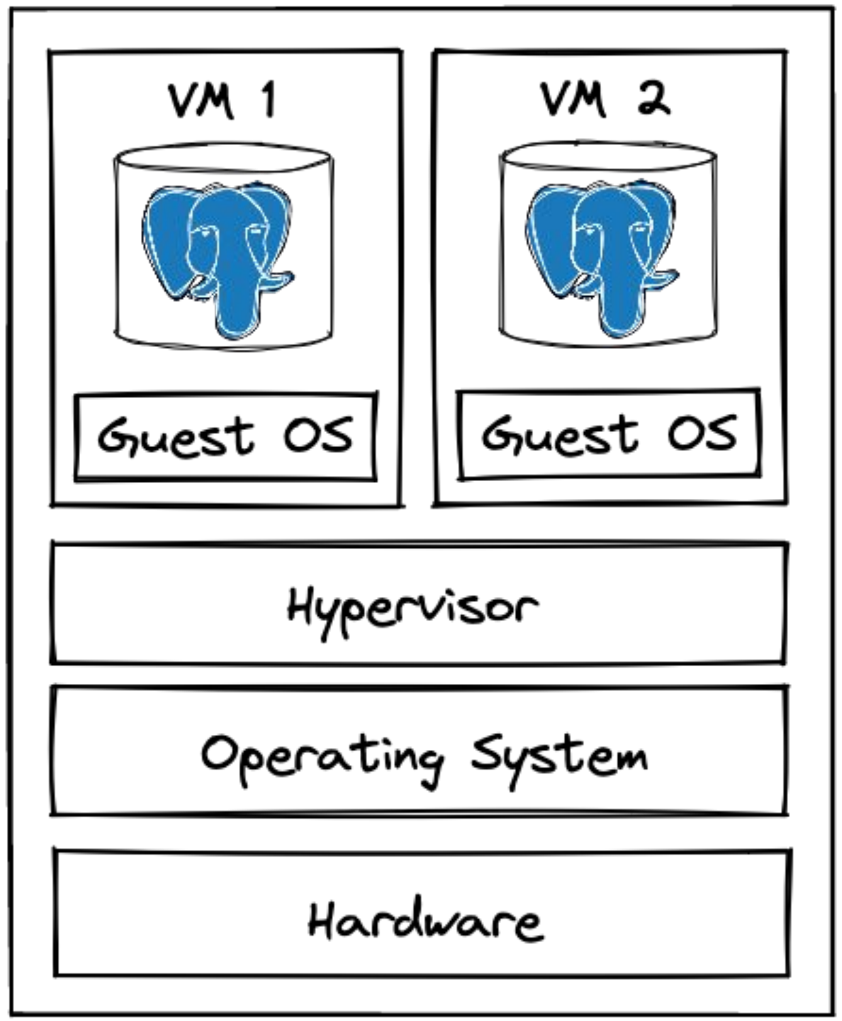
- Datatypes
- Transactional DDL
- Foreign Data Wrappers
- Concurrent Index Creation
- Conditional indexes
- JSON/JSONB
- Common Table Expressions
- Fast column addition
- Listen/Notify
- Upsert
- Partitioning
- Per transaction sync replication
- Window function
- Continued innovation

crunchy data

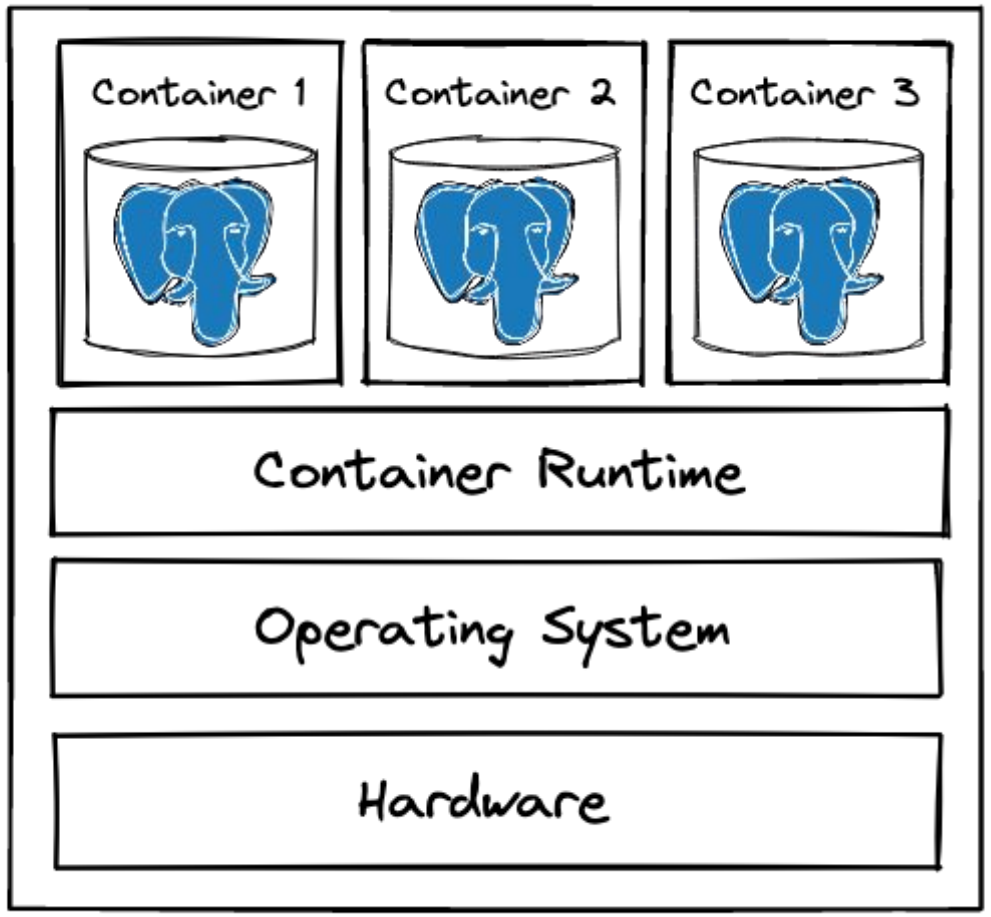
Why Postgres on OpenShift?



crunchydata



Virtual Machines



Containers

crunchy data

Features of Containers

- Isolated
- Lightweight
- Portable
- Scalable

❗ Stateless

❗ Ephemeral

Container Orchestration

- Manage many containers
- Automate container lifecycle
- Integrate with DevOps tools

Persistent Volumes $\longrightarrow$ Storage

• Provisioning	• Storage
• Deployment	• Services
• Configuration	• Resource allocation
• Scheduling	• Load balancing
• Scaling	• Networking
• Self-healing	• Security

crunchydata

Sidecars

(with a little help from my friends)

- Pod: “wrapper” around one or more containers
- “Helper” container
- Tightly coupled with main container in a pod
- e.g. metrics exporter, database backup tool

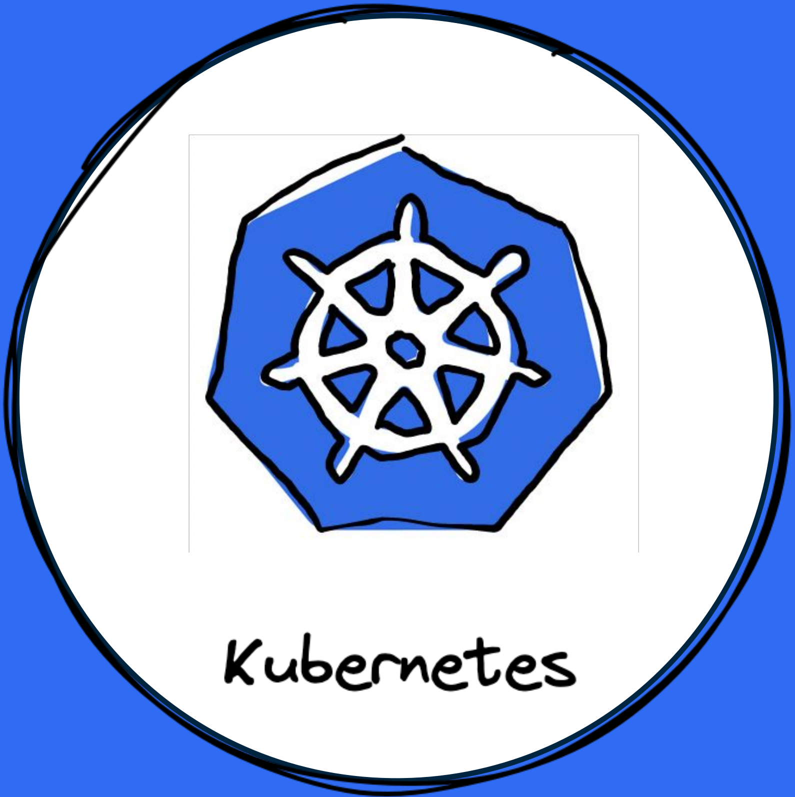
crunchydata

StatefulSets

(not all containers are created equal)

- Stable, persistent storage
- Ordered, graceful deployment and scaling
- Ordered, automated rolling updates

Running Postgres on Kubernetes



crunchy data

Sensitivity: C2 Internal

Crunchy Data Postgres SQL clusters @SDX

- SDX is following a Cloud native approach to the extent possible**
- SDX is using Kubernetes/OpenShift as its Micro service platform as abstraction Layer (unifying)
- SDX runs per member a "SDX DLT Node" based on Corda R3
- Each DLT Member Node requires its own Crunchy Data Postgres SQL cluster (DLT=Digital Ledger Technology)

= SDX runs a substantial amount of Postgres clusters with a small operations team

**regulatory and technical limitations apply

Agenda

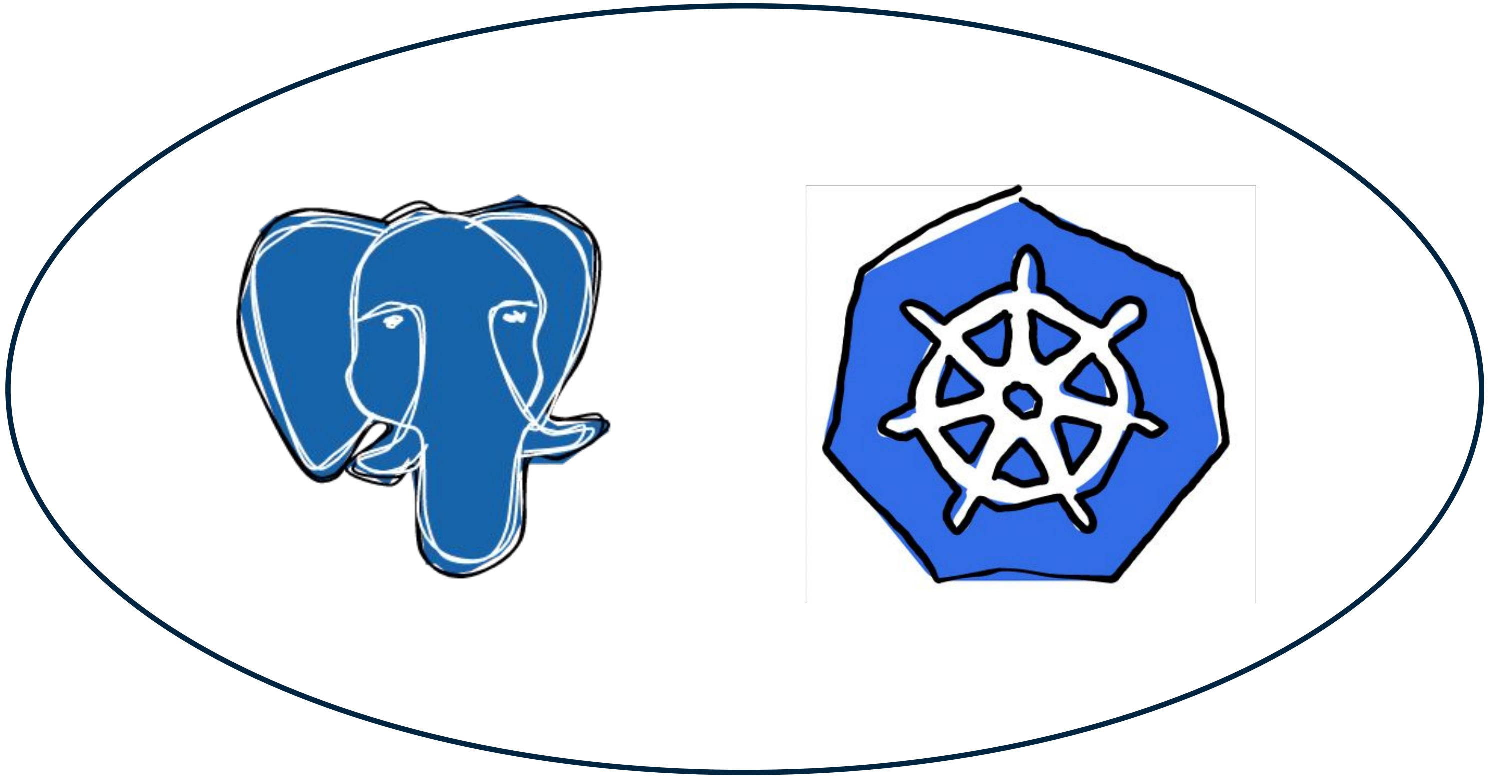
- Why Crunchy Data?
- Why Postgres on OpenShift?
- **Whoa, that sounds complicated!**
- What is Crunchy Postgres for Kubernetes?
- Demo

Kubernetes Operators

“Operators are software extensions to Kubernetes that make use of **custom resources** to manage applications and their components. Operators follow Kubernetes principles, notably **the control loop**.”

https://kubernetes.io/docs/concepts/extend-kubernetes/operator/

PGO
The Postgres Operator from Crunchy Data



crunchydata

Crunchy Postgres for Kubernetes

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

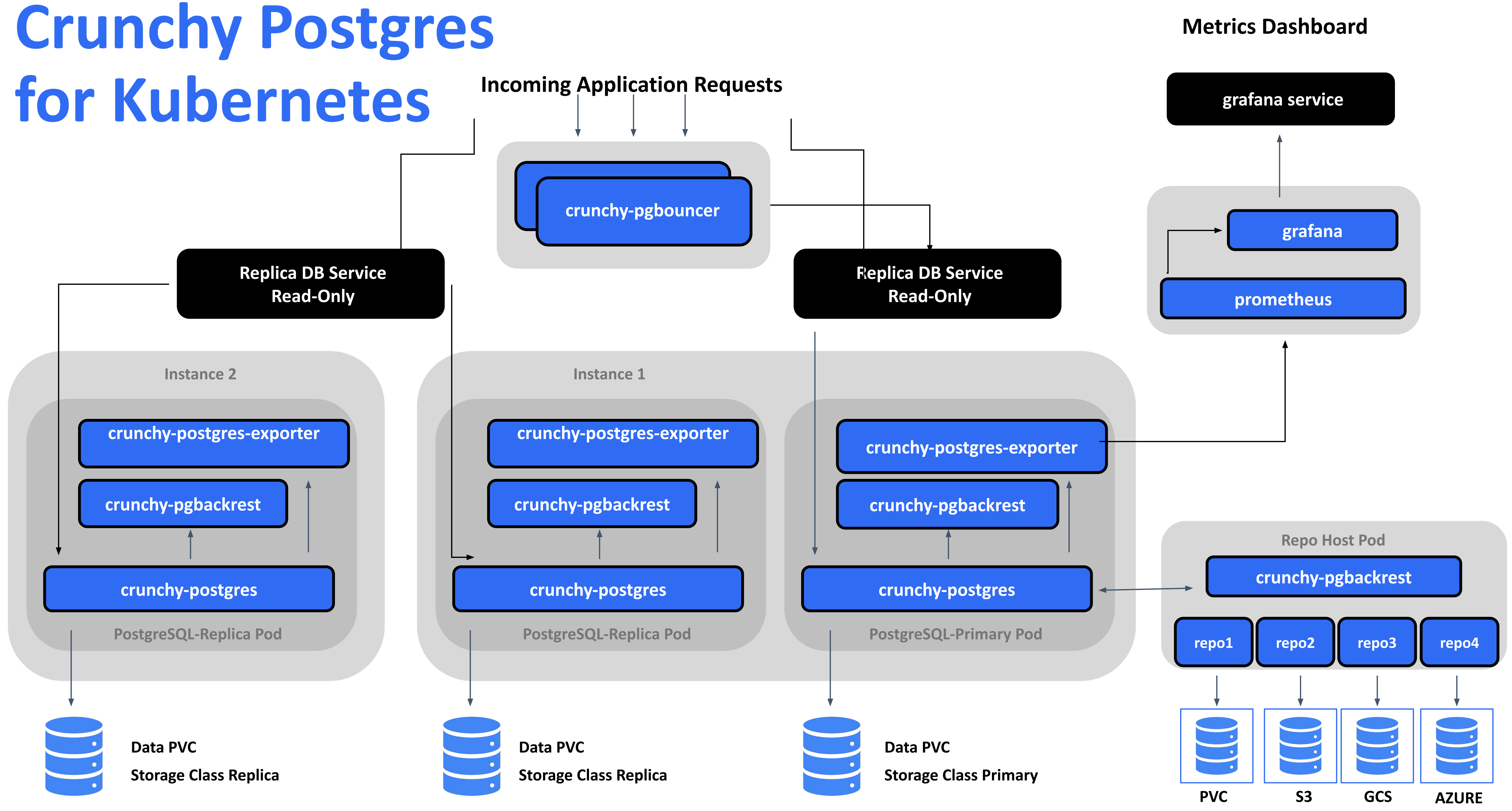
crunchy data

Why Postgres on OpenShift?

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

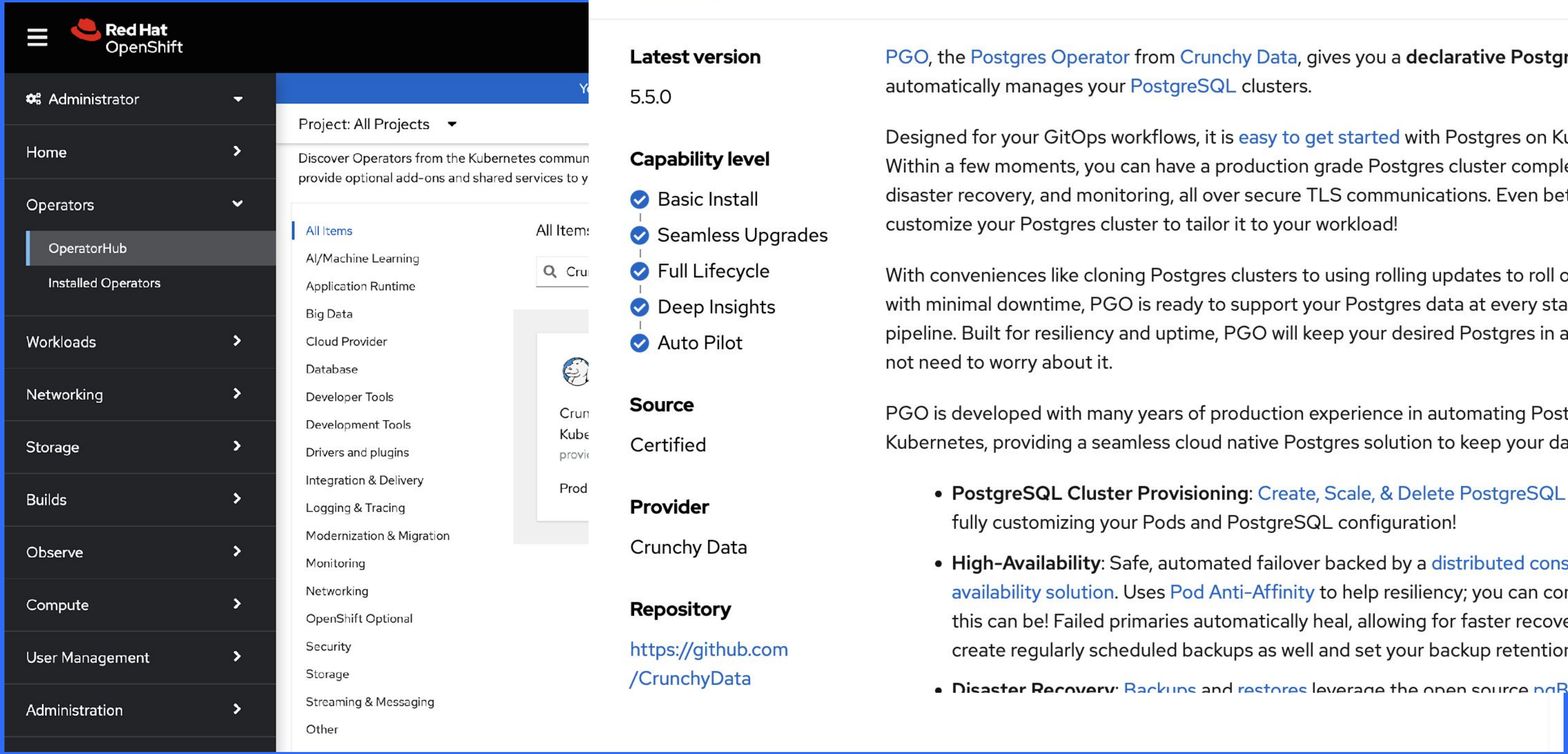
crunchy data

Crunchy Postgres for Kubernetes

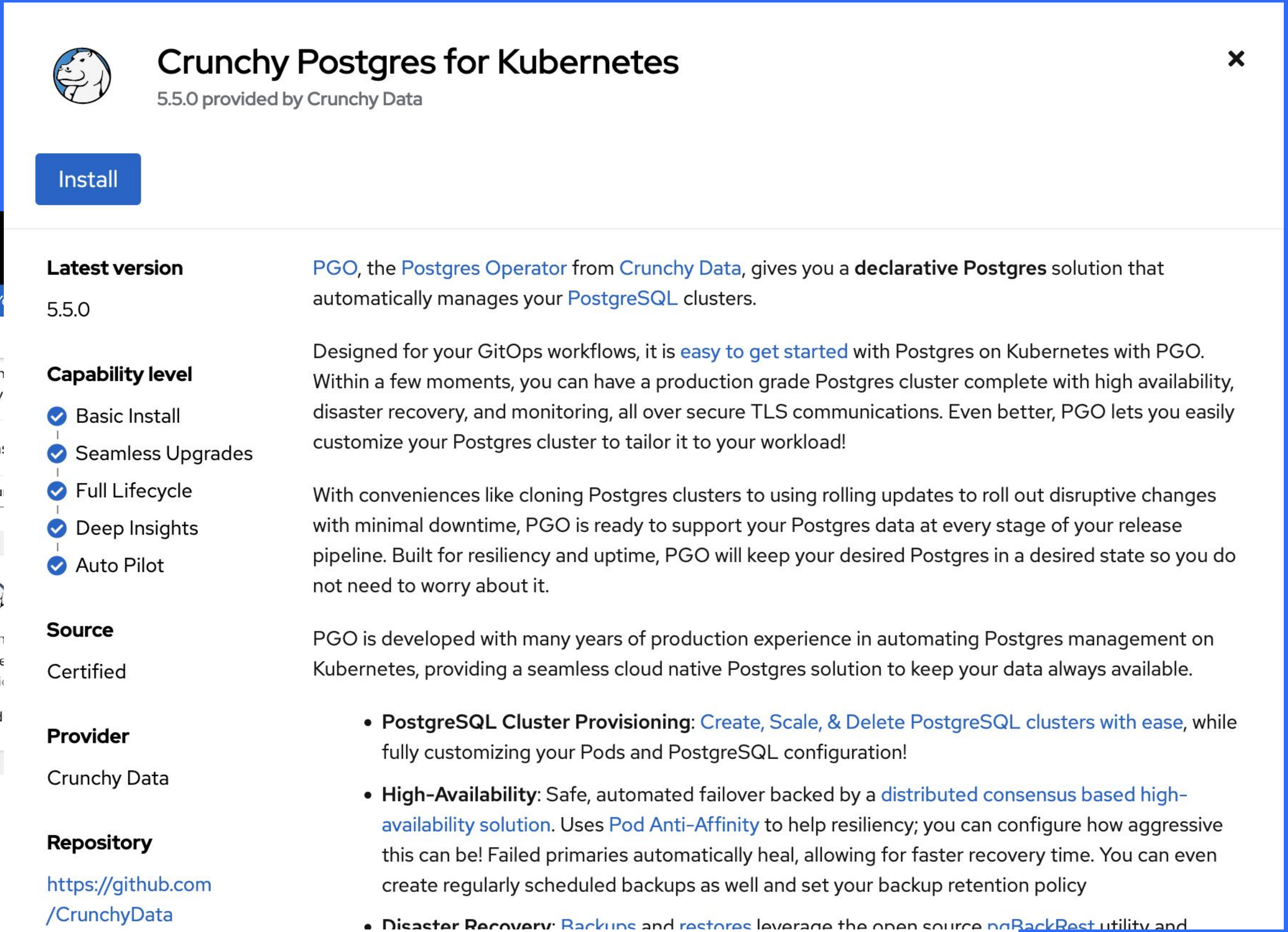


“Whoa, that sounds complicated!”

OperatorHub



Discover Operators from the Kubernetes commun
provide optional add-ons and shared services to y



Crunchy Postgres for Kubernetes

5.5.0 provided by Crunchy Data

Latest version

5.5.0

Capability level

- ✓ Basic Install
- ✓ Seamless Upgrades
- ✓ Full Lifecycle
- ✓ Deep Insights
- ✓ Auto Pilot

Source

Certified

Provider

Crunchy Data

Repository

https://github.com/CrunchyData

PGO, the Postgres Operator from Crunchy Data, gives you a **declarative Postgres** solution that automatically manages your PostgreSQL clusters.

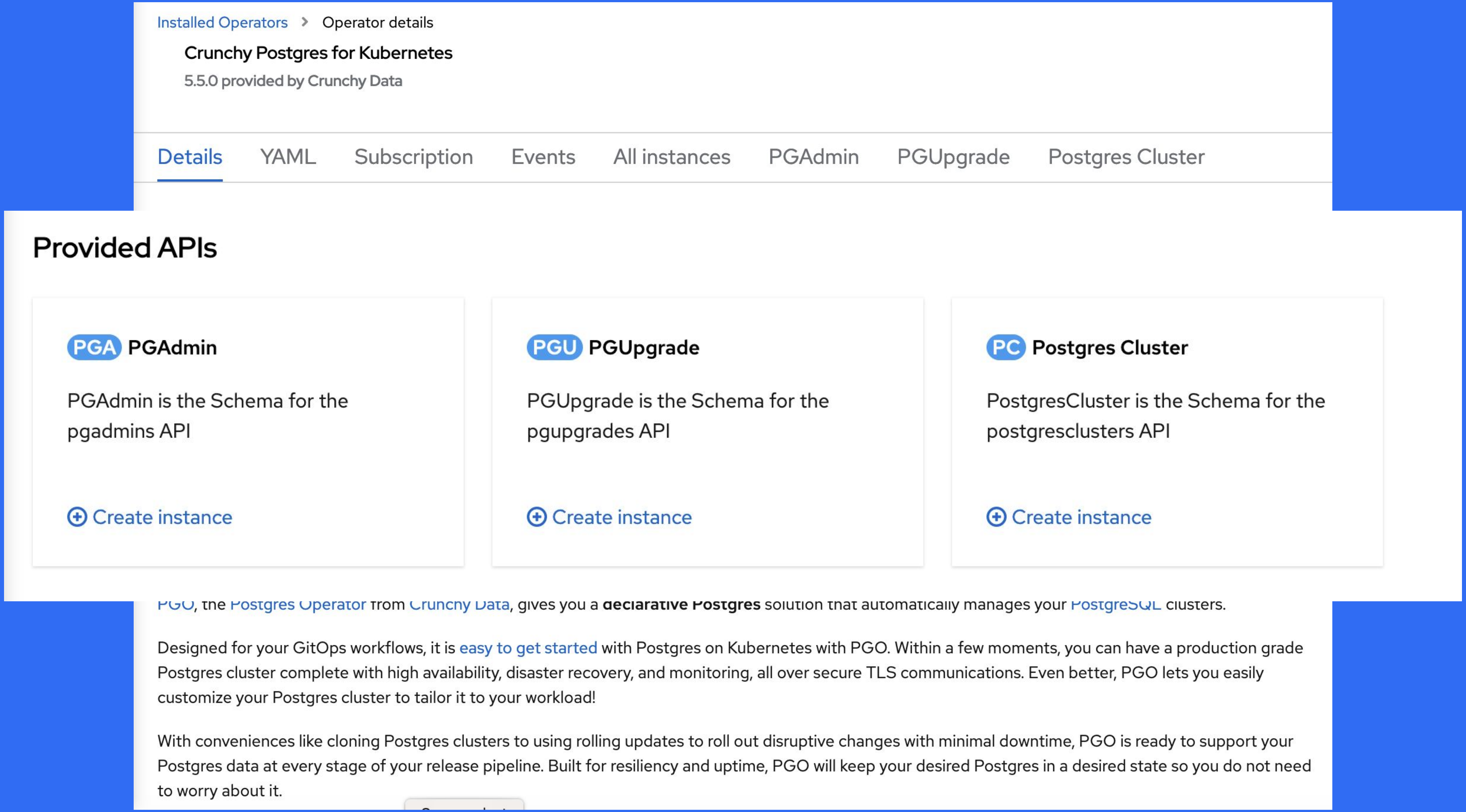
Designed for your GitOps workflows, it is easy to get started with Postgres on Kubernetes with PGO. Within a few moments, you can have a production grade Postgres cluster complete with high availability, disaster recovery, and monitoring, all over secure TLS communications. Even better, PGO lets you easily customize your Postgres cluster to tailor it to your workload!

With conveniences like cloning Postgres clusters to using rolling updates to roll out disruptive changes with minimal downtime, PGO is ready to support your Postgres data at every stage of your release pipeline. Built for resiliency and uptime, PGO will keep your desired Postgres in a desired state so you do not need to worry about it.

PGO is developed with many years of production experience in automating Postgres management on Kubernetes, providing a seamless cloud native Postgres solution to keep your data always available.

- **PostgreSQL Cluster Provisioning**: Create, Scale, & Delete PostgreSQL clusters with ease, while fully customizing your Pods and PostgreSQL configuration!
- **High-Availability**: Safe, automated failover backed by a distributed consensus based high-availability solution. Uses Pod Anti-Affinity to help resiliency; you can configure how aggressive this can be! Failed primaries automatically heal, allowing for faster recovery time. You can even create regularly scheduled backups as well and set your backup retention policy

crunchydata



Installed Operators > Operator details

Crunchy Postgres for Kubernetes

5.5.0 provided by Crunchy Data

Provided APIs

PGA PGAdmin

PGAdmin is the Schema for the pgadmins API

PGU PGUpgrade

PGUpgrade is the Schema for the pgupgrades API

PC Postgres Cluster

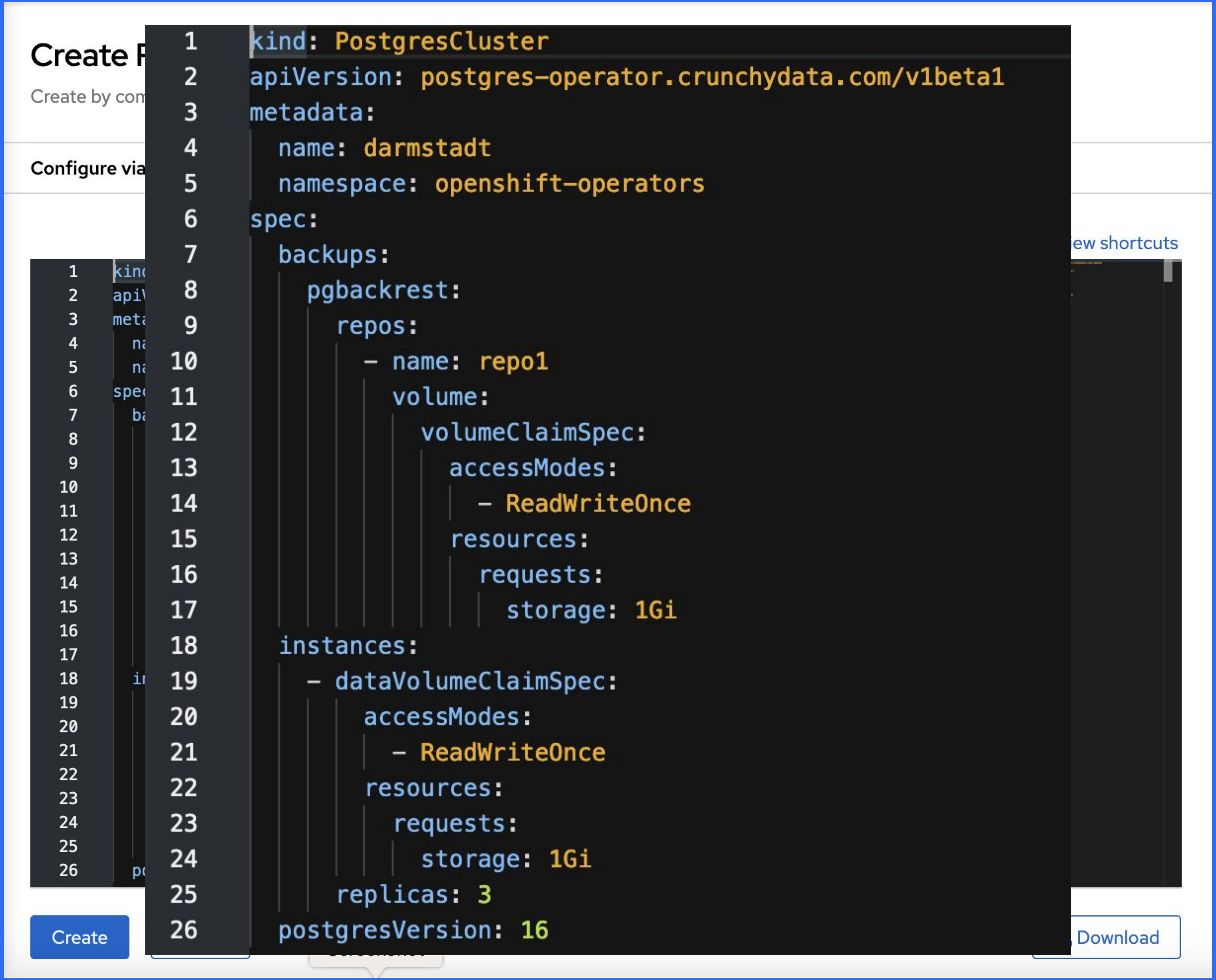
PostgresCluster is the Schema for the postgresclusters API

PGO, the Postgres Operator from Crunchy Data, gives you a **declarative Postgres** solution that automatically manages your PostgreSQL clusters.

Designed for your GitOps workflows, it is easy to get started with Postgres on Kubernetes with PGO. Within a few moments, you can have a production grade Postgres cluster complete with high availability, disaster recovery, and monitoring, all over secure TLS communications. Even better, PGO lets you easily customize your Postgres cluster to tailor it to your workload!

With conveniences like cloning Postgres clusters to using rolling updates to roll out disruptive changes with minimal downtime, PGO is ready to support your Postgres data at every stage of your release pipeline. Built for resiliency and uptime, PGO will keep your desired Postgres in a desired state so you do not need to worry about it.

crunchydata



```yaml
kind: PostgresCluster
apiVersion: postgres-operator.crunchydata.com/v1beta1
metadata:
  name: darmstadt
  namespace: openshift-operators
spec:
  backups:
    pgbackrest:
      repos:
        - name: repo1
          volume:
            volumeClaimSpec:
              accessModes:
                - ReadWriteOnce
              resources:
                requests:
                  storage: 1Gi
  instances:
    - dataVolumeClaimSpec:
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 1Gi
      replicas: 3
  postgresVersion: 16
```

Important Features

- Fully Featured PostgreSQL
- High Availability
- Database backup and recovery
- Automated rolling Upgrades

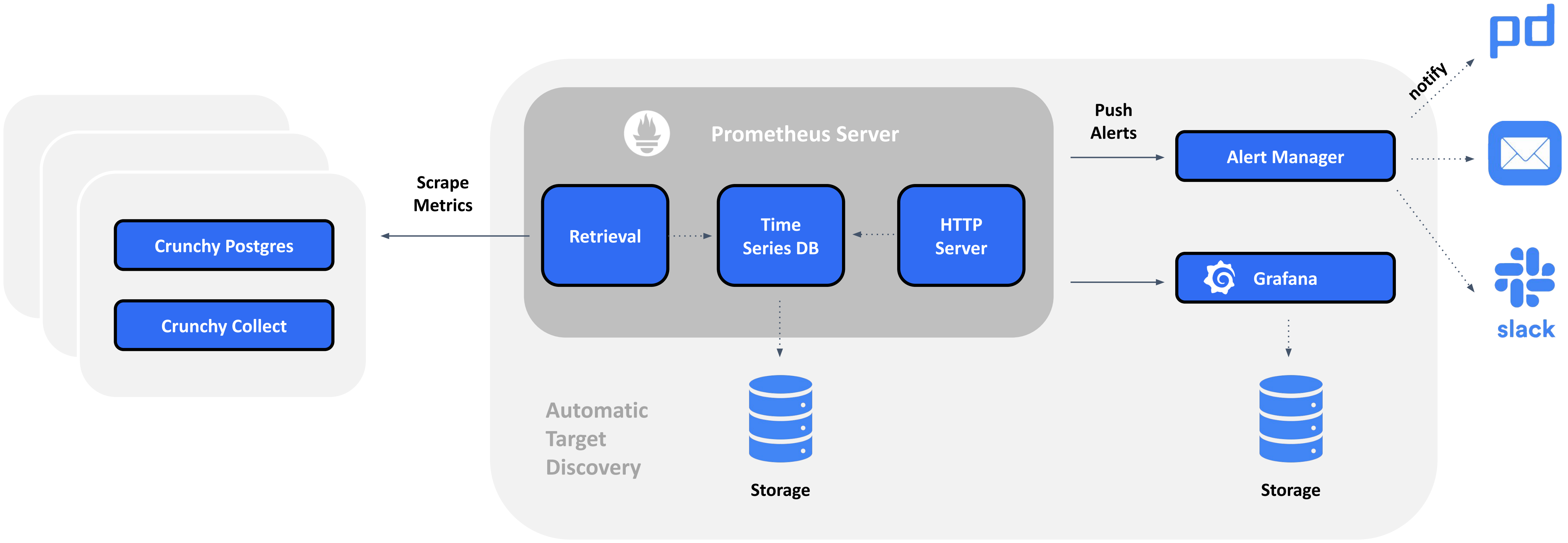
Fully featured PostgreSQL

- The full featured PostgreSQL compatibility ensured, complex applications like Corda DLT solution was working flawless
- SDX built on high availability feature it’s disaster failover scenarios
- SDX executes its regulatory DR exercises, using Crunchy Data Cluster features

Configuration

- Configure multiple backup repositories
- Implement database monitoring
- Create DR cluster
- Security – Custom TLS, host-based authentication…
- Personalise your database cluster

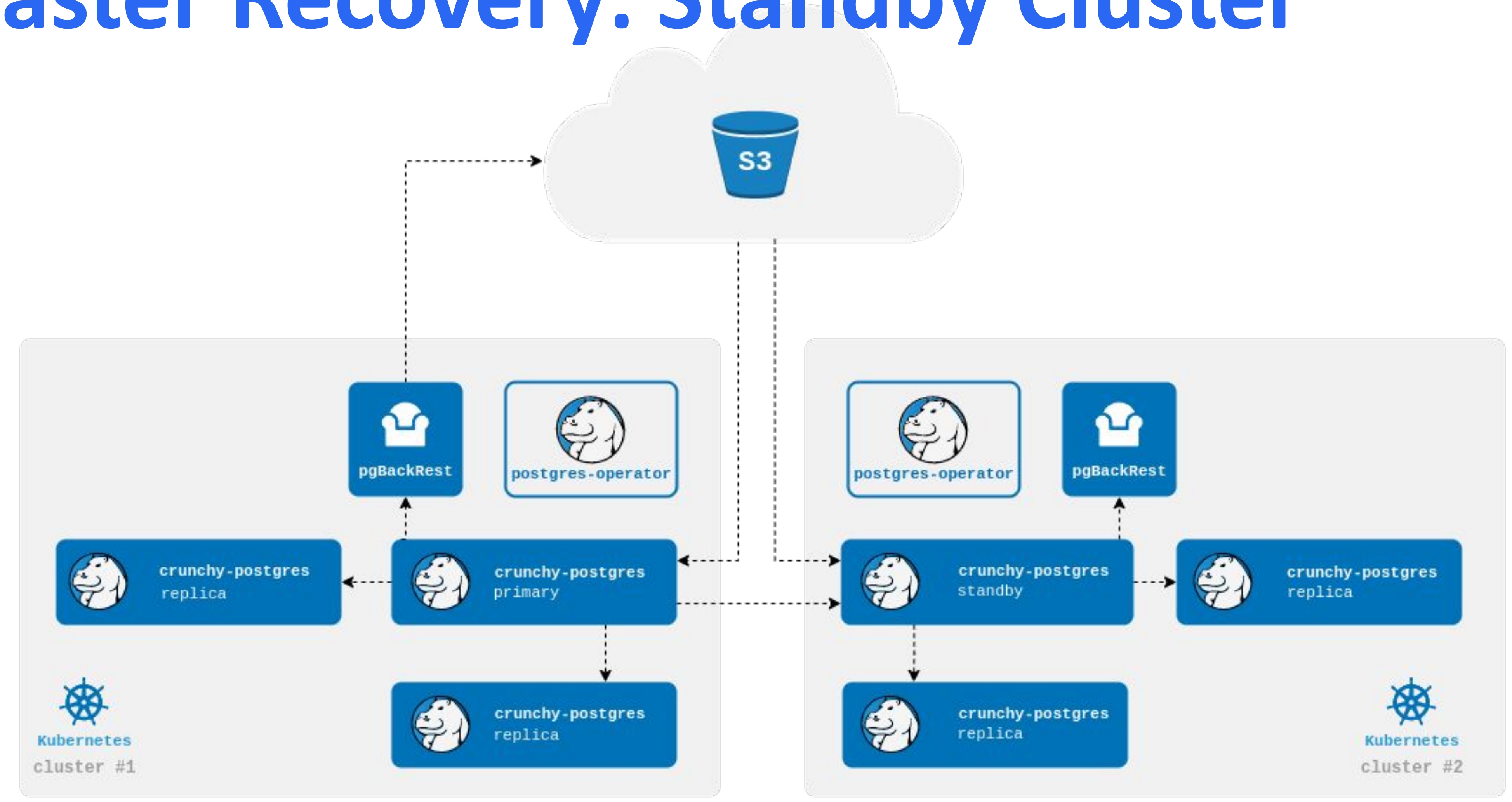
Monitoring: pgMonitor



Prometheus pgMonitoring @ SDX

- The integration with Prometheus & Grafana enables SDX to monitor all running PG Clusters seamlessly in its solution health monitoring

Disaster Recovery: Standby Cluster



crunchydata

PostgreSQL Major Version Upgrades

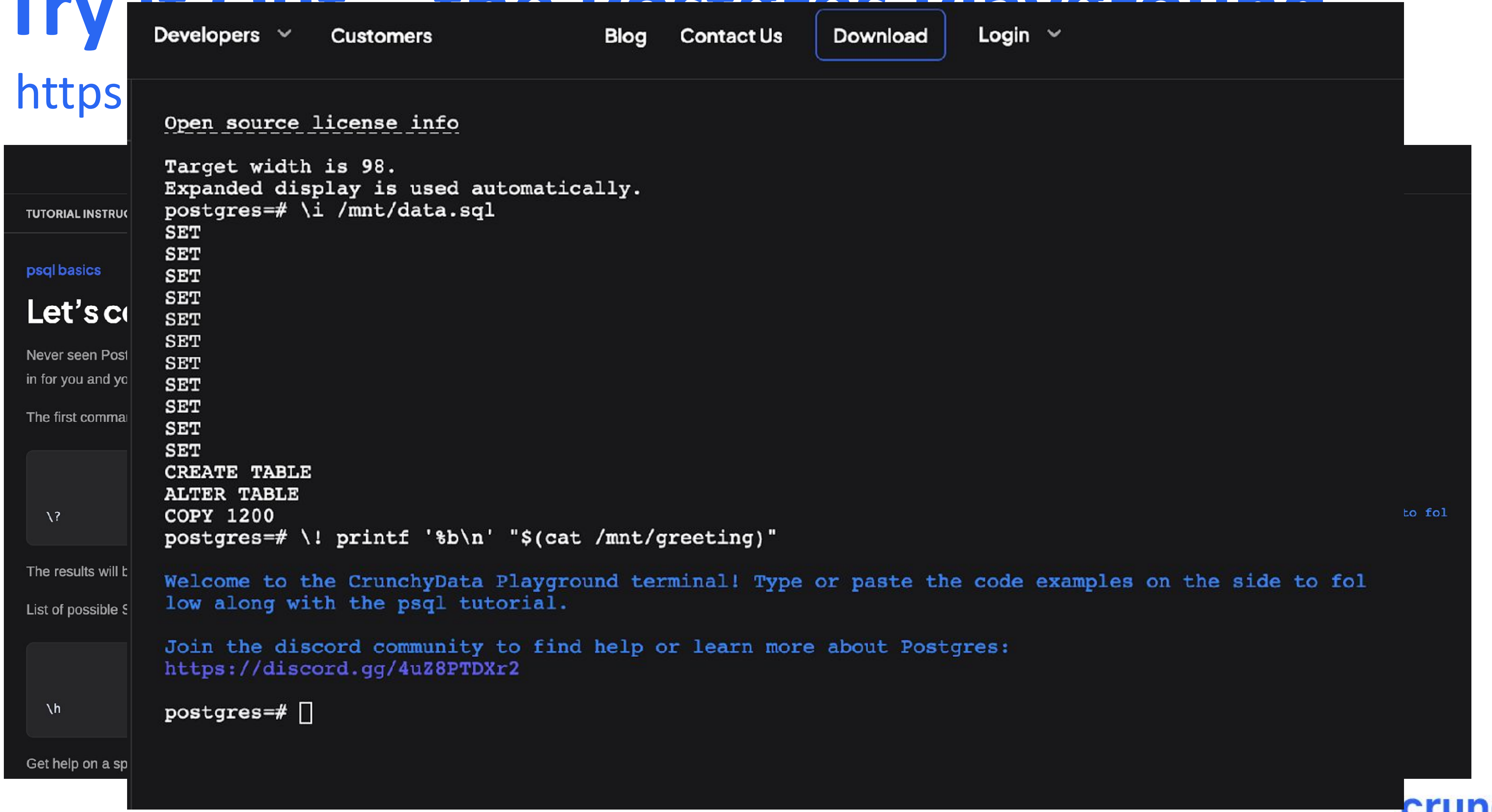
- Automated via PGUpgrade API
- Fast, In-Place Upgrades
- 1. Create PGUpgrade resource
- 2. Shutdown and annotate Postgres Cluster
- 3. Restart Postgres Cluster with new version

pgAdmin
Administration and Development Platform

- GUI tool
- pgAdmin API
- Automatic database discovery
- Server groups

Try it Out – the Postgres Playground

https

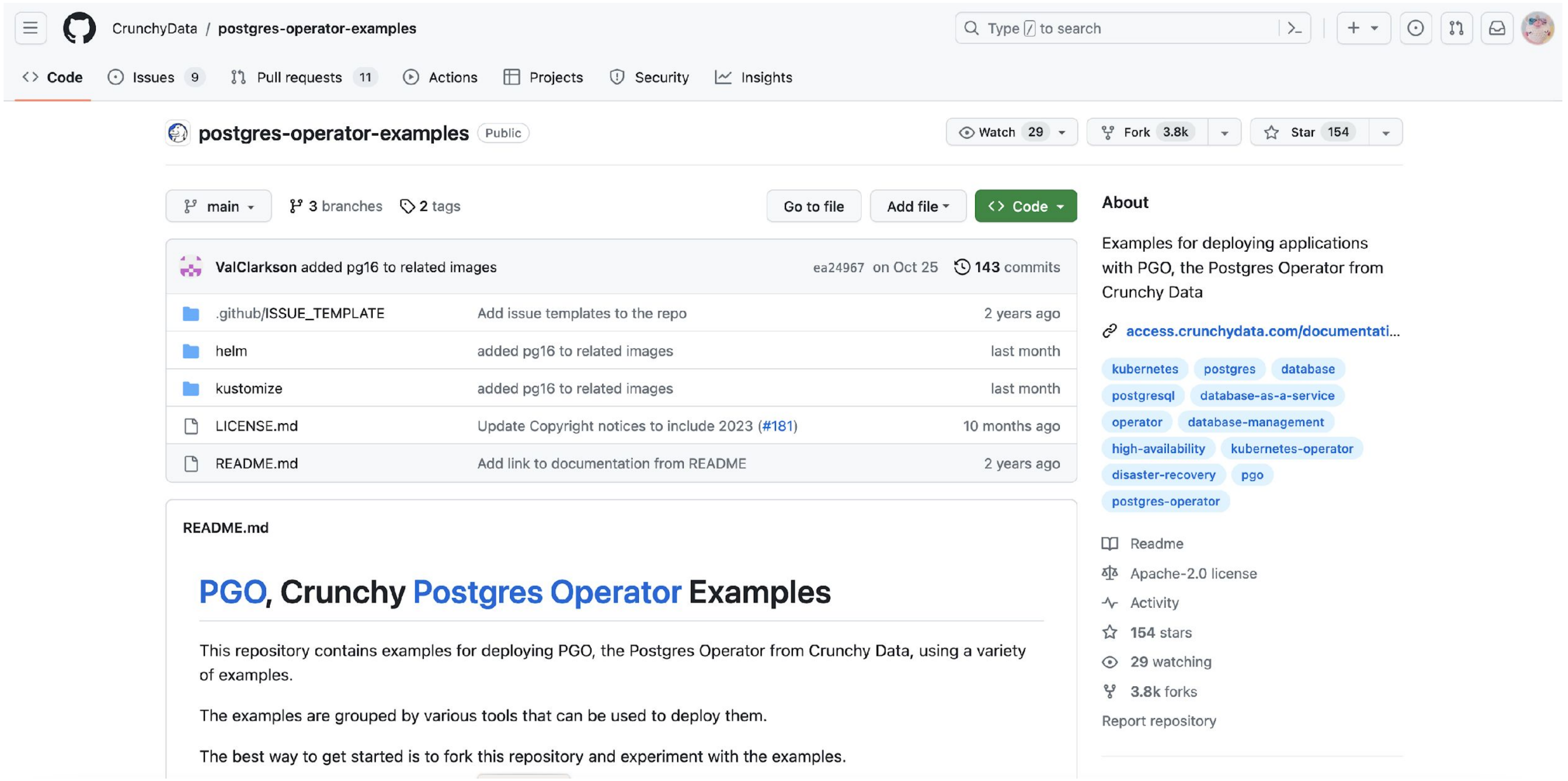


```
Open source license info

Target width is 98.
Expanded display is used automatically.
postgres=# \i /mnt/data.sql
SET
SET
SET
SET
SET
SET
SET
SET
SET
SET
SET
CREATE TABLE
ALTER TABLE
COPY 1200
postgres=# \! printf '%b\n' "$(cat /mnt/greeting)"

Welcome to the CrunchyData Playground terminal! Type or paste the code examples on the side to fol
low along with the psql tutorial.

Join the discord community to find help or learn more about Postgres:
https://discord.gg/4uZ8PTDXr2

postgres=#
```

crunchydata

Try it Out - Postgres Operator Examples

https://github.com/CrunchyData/postgres-operator-examples



ValClarkson added pg16 to related images — ea24967 on Oct 25 — 143 commits

.github/ISSUE_TEMPLATE	Add issue templates to the repo	2 years ago
helm	added pg16 to related images	last month
kustomize	added pg16 to related images	last month
LICENSE.md	Update Copyright notices to include 2023 (#181)	10 months ago
README.md	Add link to documentation from README	2 years ago

About

Examples for deploying applications with PGO, the Postgres Operator from Crunchy Data

access.crunchydata.com/documentati...

kubernetes · postgres · database · postgresql · database-as-a-service · operator · database-management · high-availability · kubernetes-operator · disaster-recovery · pgo · postgres-operator

Readme
Apache-2.0 license
Activity
154 stars
29 watching
3.8k forks
Report repository

README.md

PGO, Crunchy Postgres Operator Examples

This repository contains examples for deploying PGO, the Postgres Operator from Crunchy Data, using a variety of examples.

The examples are grouped by various tools that can be used to deploy them.

The best way to get started is to fork this repository and experiment with the examples.

crunchydata

Conclusions

- Postgres on OpenShift:

  flexible, scalable database architecture

- Crunchy Postgres for Kubernetes:

  expert Postgres & Kubernetes knowledge

- Built-in HA, DR, security, monitoring & alerting

- Automation of day-to-day DBA tasks

crunchydata

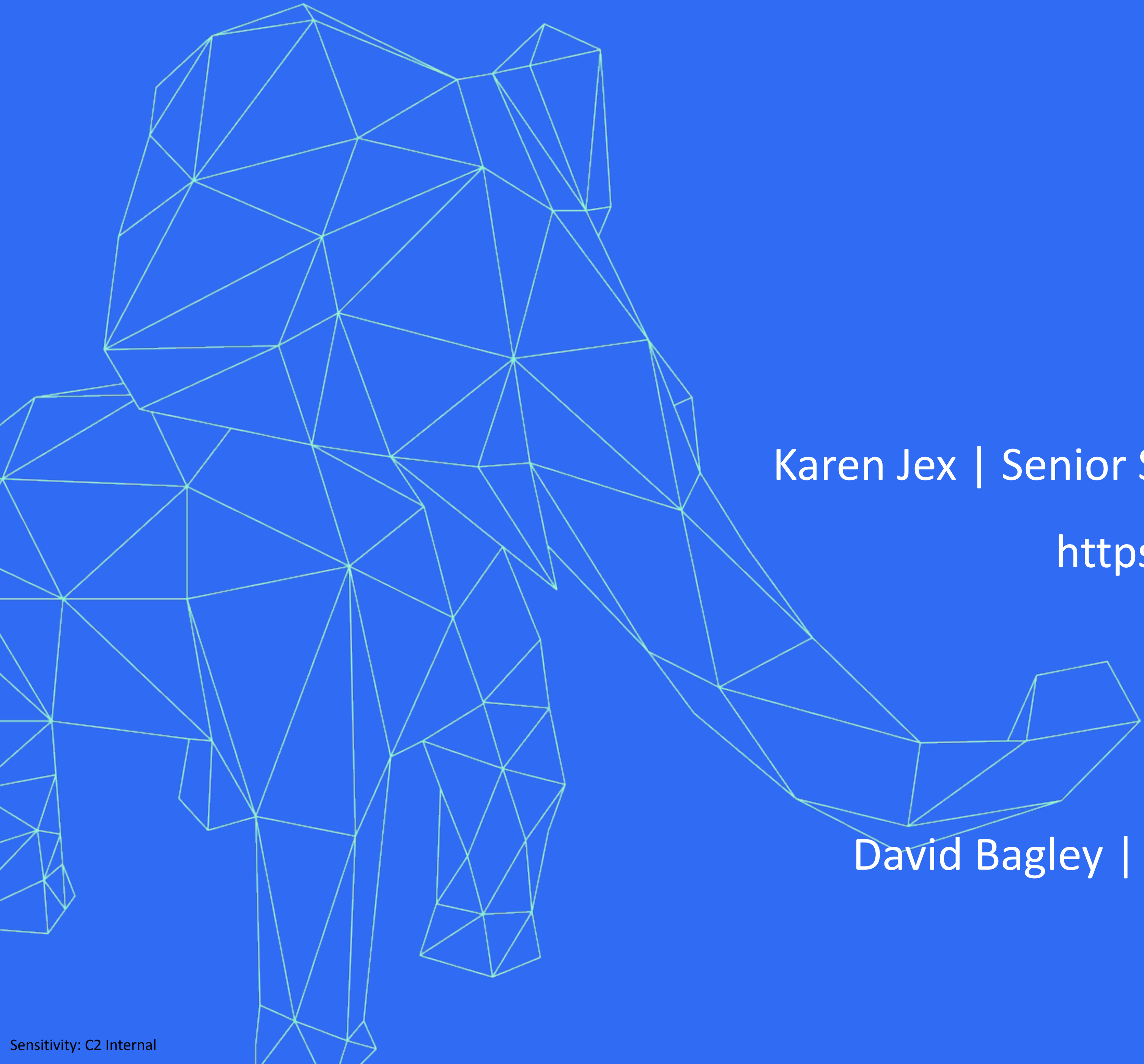
Conclusion @ SDX

Crunchy Data enabled SDX to run its complex PG cluster infrastructure.

As Crunchy provides an off the shelf solution, delivering always tested and up to data package, we are enabled to run the solution with min. amount of people effort.

The High availability features, enable SDX to support the various failure and disaster scenarios, which we have to annually test in our DR exercise.

So yes – SDX can run Crunchy Data Clusters without requiring our own Postgres Experts team.



Thank You

Karen Jex | Senior Solutions Architect @ Crunchy Data

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

For sales / commercial enquiries:

David Bagley | VP Worldwide Sales @ Crunchy Data

david.bagley@crunchydata.com

Session 5: 16:30 - 17:00

Your feedback is important!

Scan the QR-code, select the session and evaluate the presentation. Thank you!

red.ht/rhsc-ch-s5

Red Hat