Deadly Sins when running CloudNativePG

Boriss Mejías
Holistic System Software Engineer
Air Guitar Player
PgConf Belgium – May 5th, 2026

Socrates

Heavy Metal

as religion
(UK census 2011)

Heavy Metal

as religion
(UK census 2011)

Saints and Sinners

Heavy Metal

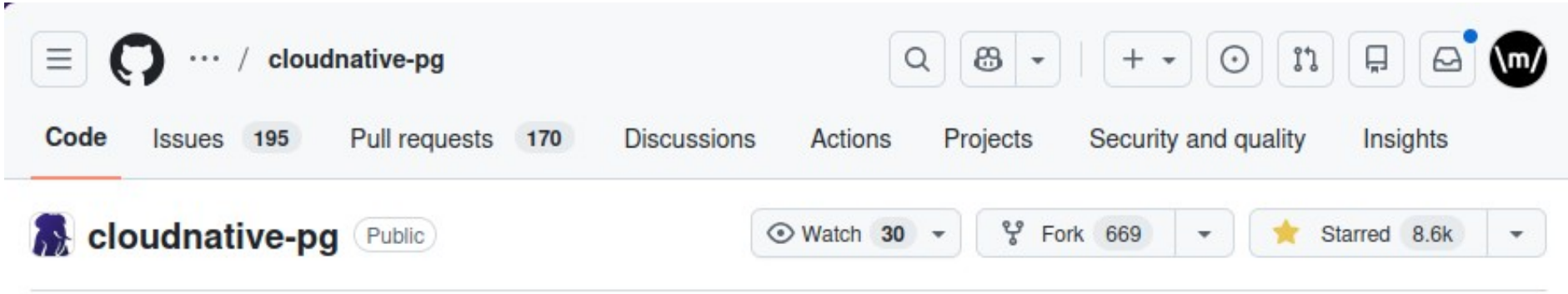
as religion
(UK census 2011)

True Metalheads and Posers

Heavy Metal

🤘 6,242

as religion
(UK census 2011)



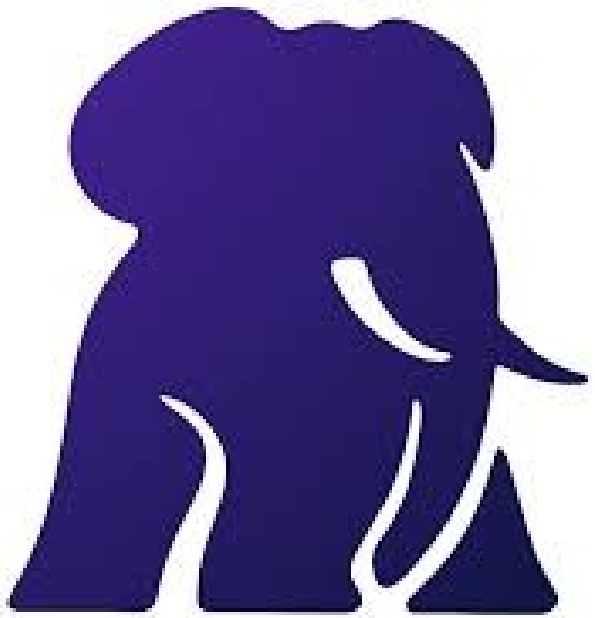
Heavy Metal

🤘 6,242

as religion
(UK census 2011)

Druids
Scientology
Satanism

programmerhumor.io



Version: 1.29

Before you start

Before we get started, it is essential to go over some terminology that is specific to Kubernetes and PostgreSQL.

Kubernetes terminology

Node : A *node* is a worker machine in Kubernetes, either virtual or physical, where all services necessary to run pods are managed by the control plane node(s).

Postgres Node : A *Postgres node* is a Kubernetes worker node dedicated to running PostgreSQL workloads. This is achieved by applying the `node-role.kubernetes.io` label and taint, as proposed by CloudNativePG. It is also referred to as a `postgres` node.

Pod : A *pod* is the smallest computing unit that can be deployed in a Kubernetes cluster and is composed of one or more containers that share network and storage.

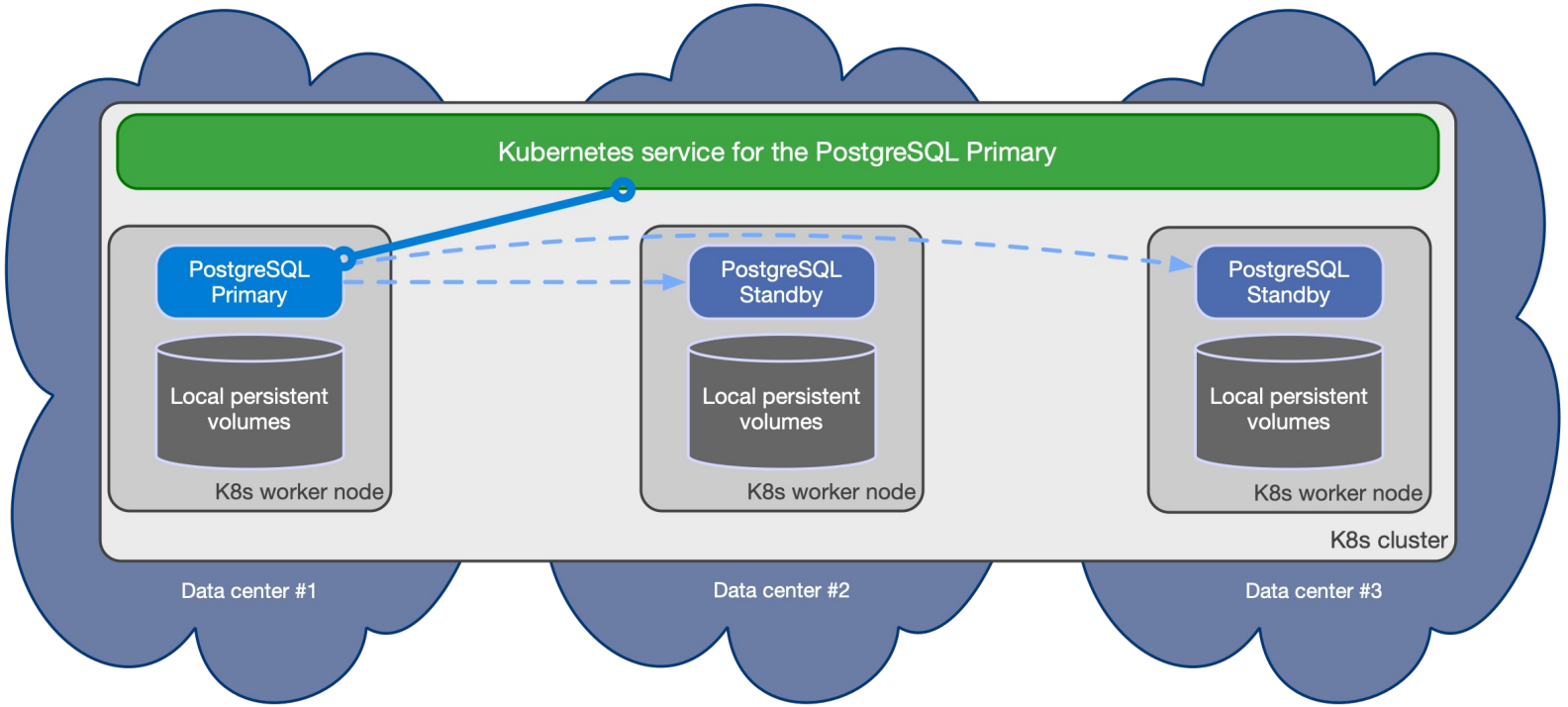
Service : A *service* is an abstraction that exposes as a network service an application that runs on a group of pods and standardizes important features such as service discovery across applications, load balancing, failover, and so on.

Secret : A *secret* is an object that is designed to store small amounts of sensitive data such as passwords, access keys, or tokens, and use them in pods.

Storage Class : A *storage class* allows an administrator to define the classes of storage in a cluster, including provisioner (such as AWS EBS), reclaim policies, mount options, volume expansion, and so on.

Persistent Volume : A *persistent volume* (PV) is a resource in a Kubernetes cluster that represents storage that has been either manually provisioned by an administrator or dynamically

CloudNativePG



```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: sinner
spec:
  instances: 3

  storage:
    size: 1Gi
```

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: sinner
spec:
  instances: 3

  storage:
    size: 1Gi
```

Declarative Mindset

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: sinner
spec:
  instances: 3

  storage:
    size: 1Gi
  walStorage:
    size: 1Gi
```

psql?

kubectl cnpg psql sinner

Deadly Sins

ALTER SYSTEM SET

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: sinner
spec:
  instances: 3

  storage:
    size: 1Gi

  postgresql:
    parameters:
      max_connections: "42"
```

CREATE TABLESPACE

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: sinner
spec:
  instances: 3
  storage:
    size: 1Gi

  tablespaces:
  - name: data
    storage:
      size: 1Gi
  - name: idx
    storage:
      size: 1Gi
```

CREATE USER

```
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: sinner
spec:
  instances: 3
  storage:
    size: 1Gi

  managed:
    roles:
    - name: sinner
      ensure: present
      login: true
      superuser: false
      passwordSecret:
        name: sinner-credentials
```

```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: sinner
spec:
  instances: 3
  storage:
    size: 1Gi

  managed:
    roles:
    - name: sinner
      ensure: present
      login: true
      superuser: false
      passwordSecret:
        name: sinner-credentials
```

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: sinner-credentials
  namespace: default
  labels:
    cnpg.io/reload: "true"
type: kubernetes.io/basic-auth
data:
  username: c2lubmVy
  password: cGVjYWRv
```

CREATE DATABASE

```
apiVersion: postgresql.cnpg.io/v1
kind: Database
metadata:
  name: sinner-db
spec:
  name: sinner
  owner: sinner
  cluster:
    name: sinner
```

CREATE EXTENSION

```
apiVersion: postgresql.cnpg.io/v1
kind: Database
metadata:
  name: sinner-db
spec:
  name: sinner
  owner: sinner
  cluster:
    name: sinner
```

```
apiVersion: postgresql.cnpg.io/v1
kind: Database
metadata:
  name: sinner-db
spec:
  name: sinner
  owner: sinner
  cluster:
    name: sinner
  extensions:
  - name: vector
```

Use md5 or scram-sha-256 for replication

```
$ kubectl get secrets
NAME                 TYPE                                  DATA   AGE
sinner-app           kubernetes.io/basic-auth              11     152m
sinner-ca            Opaque                                2      152m
sinner-credentials   kubernetes.io/basic-auth              2      15h
sinner-replication   kubernetes.io/tls                     2      152m
sinner-server        kubernetes.io/tls                     2      152m
```

```
$ kubectl get secrets
NAME                  TYPE                                  DATA   AGE
sinner-app            kubernetes.io/basic-auth              11     152m
sinner-ca             Opaque                                2      152m
sinner-credentials    kubernetes.io/basic-auth              2      15h
sinner-replication    kubernetes.io/tls                     2      152m
sinner-server         kubernetes.io/tls                     2      152m
```

```
$ kubectl view-secret sinner-credentials
| Secret Data
| Found 2 keys in secret "sinner-credentials". Choose
one or select 'all' to view.
| > all
|   password
|   username

$ kubectl view-secret sinner-replication
| Secret Data
| Found 2 keys in secret "sinner-replication". Choose
one or select 'all' to view.
| > all
|   tls.crt
|   tls.key
```

```
sinner=# select * from pg_hba_file_rules;

sinner=# select usename, passwd from pg_shadow;
```

pg_ctl promote

```
$ kubectl cnpg status sinner

$ kubectl describe pod sinner-3 | grep instanceRole

$ kubectl cnpg promote sinner sinner-3

$ kubectl describe pod sinner-3 | grep instanceRole
```

Closing Words

Think declarative

Use the cnpg plugin

Get a k8s expert in your team

Delegate security to k8s declaratively

Thank You

🐘 @tchorix@mastodon.world
🦋 @tchorix.bsky.social
💼 linkedin.com/in/boriss-mejías-4637401
🐙 github.com/bmejias