# Automating Postgres databases in your clusters

Cloud native PostgreSQL operator

# > whoami

- DevOps / SysAdmin @ Walkbase
- Started on ZX spectrum and Win98, nowadays Linux only
- Kubernetes user since 1.9

- Padel, badminton and table tennis
- Winter swimming
- Brewing and breaking my homelab

https://github.com/LinAnt/

But they said you shouldn’t run stateful workloads in Kubernetes?

# What is this operator thing?

# Kubernetes - an extensible platform

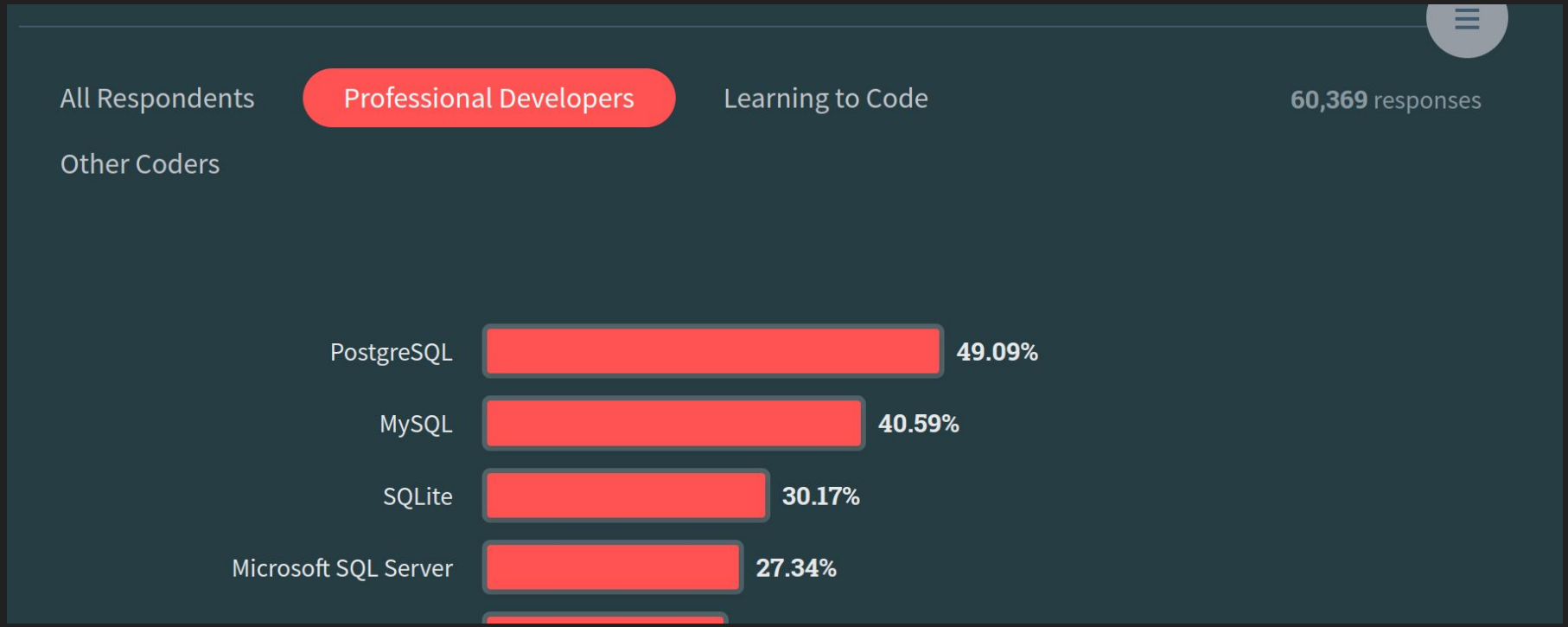
- Custom Resource Definitions (CRDs)
  - Custom Resources (CRs)

- Operator (fancy reconcile loop)
  - Watches for CRs and ensures the cluster state matches
  - Life-cycle management
    - Deploy
    - Update / Configure
    - Monitoring?
    - Backups / Snapshots etc ?
    - Upgrades?

# Operator Summary

- Keep infrastructure in control
- Resource Scalability
- Monitoring Scalability
- Knowledge Scalability

# So, what about PostgreSQL?




https://survey.stackoverflow.co/2023/#most-popular-technologies-database-prof

# Lots of buttons to press

- PGTune -> get the them defaults optimized for your hardware
- Replication?
  - Availability zones
- Users?
- Backups
  - Pg_dump
  - Tools like pgBackRest, barman, pg_basebackup

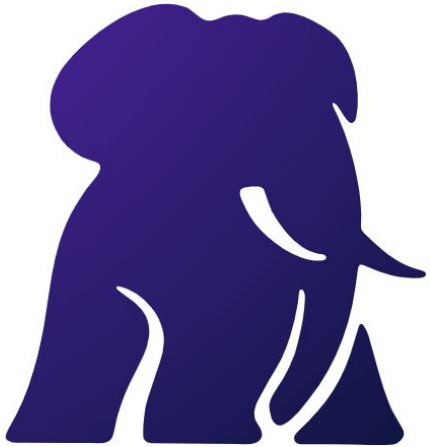

ONE DOES NOT SIMPLY
MANUALLY OPERATE A DATABASE IN PRODUCTION
imgflip.com

# Run PostgreSQL.

## The Kubernetes way.

CloudNativePG is the Kubernetes operator that covers the full lifecycle of a highly available PostgreSQL database cluster with a primary/standby architecture, using native streaming replication.

View on GitHub



### Autopilot

It automates the steps that a human operator would do to deploy and to manage a Postgres database inside Kubernetes, including automated failover.

### Data persistence

It doesn't rely on statefulsets and uses its own way to manage persistent volume claims where the PGDATA is stored.

### Designed for Kubernetes

It's entirely declarative, and directly integrates with the Kubernetes API server to update the state of the cluster — for this reason, it does not require an external failover management tool.

# Let’s give it a spin!

```
helm repo add cnpg https://cloudnative-pg.github.io/charts

helm upgrade --install cnpg \
  --namespace cnpg-system \
  --create-namespace \
   cnpg/cloudnative-pg
```

```
> kubectl krew install cnpg
```

https://artifacthub.io/packages/krew/krew-index/cnpg

# Important decisions to be made!

# Type of Nodes?

# Type of storage?

# Initial settings?

# Let’s design our demo cluster

- 3 instances
- 20G storage
- 5G WAL storage
- Automated backups
- mTLS

# kubectl apply -f cluster.yaml

File: **cluster.yaml**

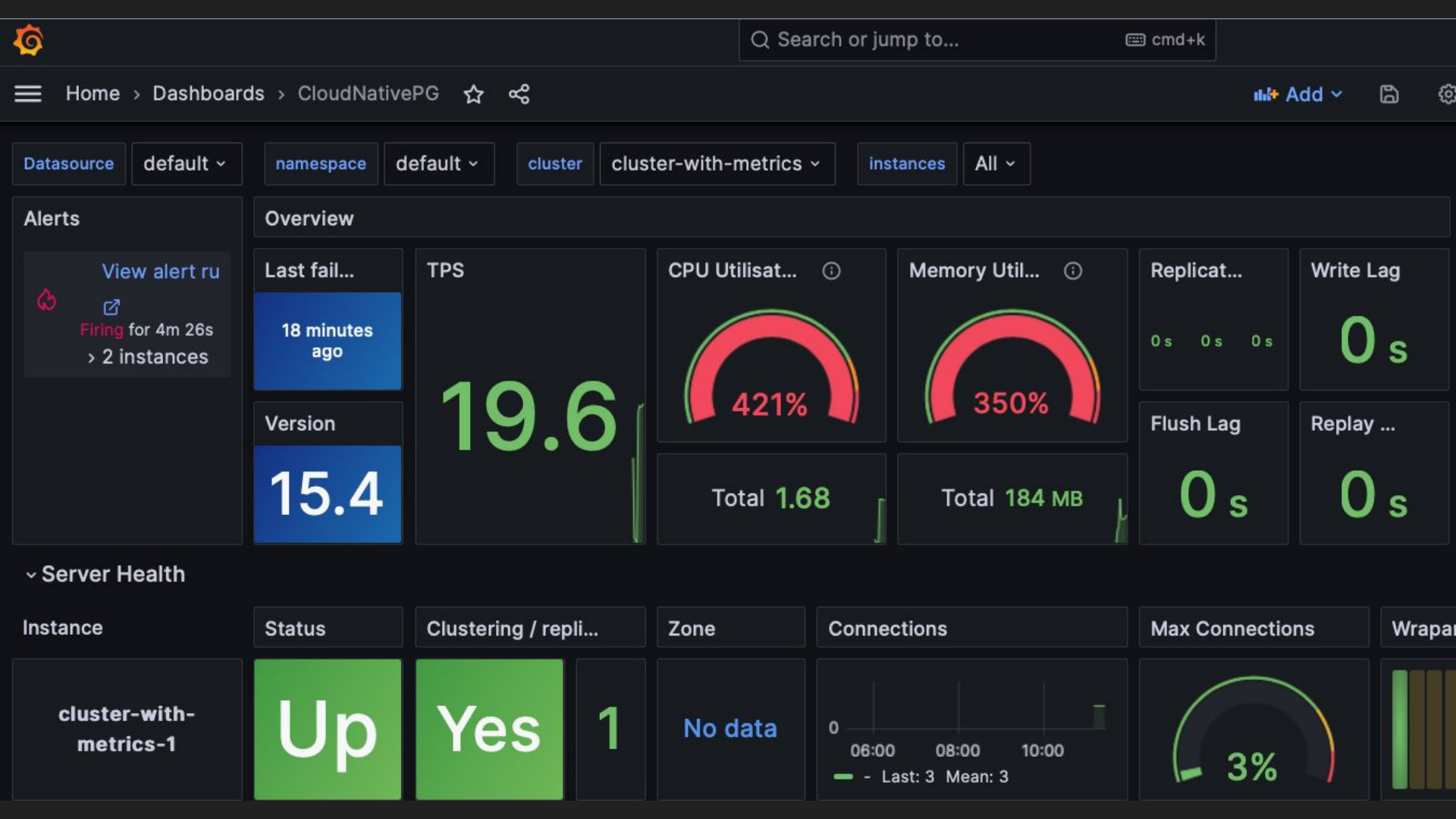
```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: demo-cluster
spec:
  instances: 3
  storage:
    pvcTemplate:
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 20Gi
      storageClassName: longhorn
      volumeMode: Filesystem
  walStorage:
    pvcTemplate:
      accessModes:
        - ReadWriteOnce
      resources:
        requests:
          storage: 5Gi
      storageClassName: longhorn
      volumeMode: Filesystem

```

# Stuff I didn’t show

# Connection Pooling

# Point in time recovery

# Custom plugins like Timescaledb


Search or jump to...
cmd+k
Home › Dashboards › CloudNativePG
Add
Datasource
default
namespace
default
cluster
cluster-with-metrics
instances
All
Alerts
View alert ru
Firing for 4m 26s
› 2 instances
Overview
Last fail...
18 minutes ago
Version
15.4
TPS
19.6
CPU Utilisat...
421%
Total 1.68
Memory Util...
350%
Total 184 MB
Replicat...
0 s 0 s 0 s
Write Lag
0 s
Flush Lag
0 s
Replay ...
0 s
Server Health
Instance
Status
Clustering / repli...
Zone
Connections
Max Connections
cluster-with-metrics-1
Up
Yes
1
No data
0
06:00
08:00
10:00
- Last: 3 Mean: 3
3%

# Evaluation

+ Documentation
+ Easy to manage running clusters
+ Keeps clusters in sync
+ GitOps friendly
+ Office Hours

+ Slack Channel
+ Well managed github repo / org

- Documentation
- Clusters sometime break
- No major version upgrades AFAIK

# Questions?