

CloudNativePG

PostgreSQL på Kubernetes

CNPG Docs | Workshop Repo

Hvorfor CloudNativePG?

Utfordringen / Løsningen

Utfordringen

Å kjøre databaser på Kubernetes har tradisjonelt vært komplisert:

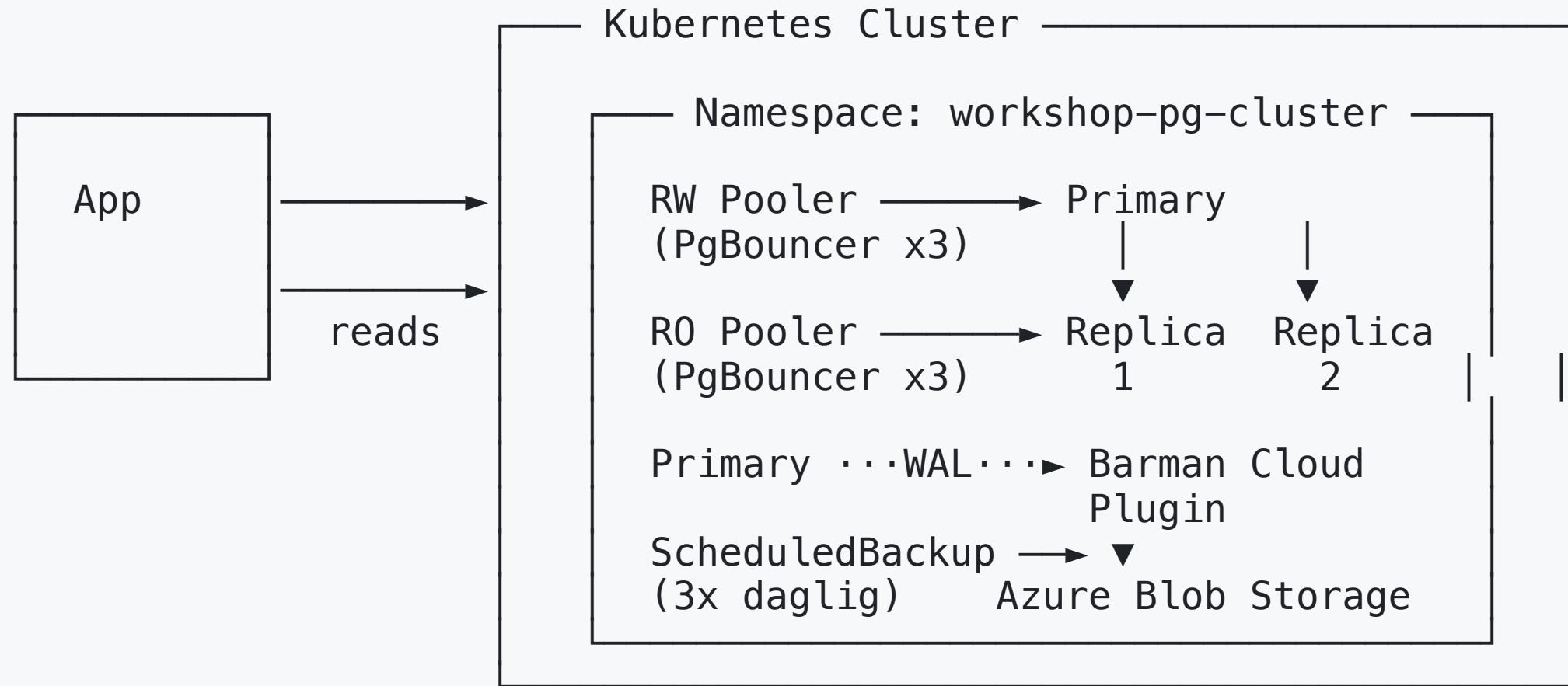
- Stateful workloads i en stateless-plattform
- Manuell konfigurasjon av replikering
- Failover krever eksterne verktøy
- Backup og restore er opp til deg selv
- TLS og sikkerhet er ekstraarbeid

Løsningen: CNPG

CloudNativePG er en Kubernetes-operator bygd for PostgreSQL:

- Open source (Apache 2.0)
- Direkte integrasjon med Kubernetes API
- Ingen eksterne avhengigheter
- Deklarativ konfigurasjon
- Self healing cluster
- Innebygd backup og restore
- Connection pooling med PgBouncer

Eksempel arkitekturoversikt



Høy tilgjengelighet

CNPG sørger for at databasen alltid er tilgjengelig

Automatisk failover

Når primær instansen går ned, blir replicaen med mest oppdatert data promotet automatisk.

Ingen manuell arbeid kreves.

Self-healing

Feilede instanser blir automatisk gjenskapt. Clusteret reparerer seg selv.
Operatoren overvåker kontinuerlig.

Pod anti-affinity

Instanser blir spredt over ulike noder for å tåle nodefeil.

Konfigurerbart per sone eller host.

Cluster-konfigurasjon

Deklarativt PostgreSQL-cluster med YAML

```
apiVersion: postgresql.cnpq.io/v1
kind: Cluster
metadata:
  name: workshop-pg-cluster
spec:
  instances: 3
  enableSuperuserAccess: true
  superuserSecret:
    name: admin-user
  bootstrap:
    initdb:
      database: app
      owner: app
      secret:
        name: app-user
  plugins:
    - name: barman-cloud.cloudnative-pg.io
      isWALArchiver: true
      parameters:
        barmanObjectName: workshop-backup-azure-object-store
  imageName: ghcr.io/cloudnative-pg/postgresql:18.1-system-trixie
  resources:
    requests:
      memory: 100Mi
      cpu: 200m
    limits:
      memory: 300Mi
  storage:
    storageClass: kubevirt-csi-infra-default
    size: 2Gi
```

Backup og gjenopprettning

Automatisert med Barman Cloud Plugin

Scheduled/planlagte backups

```
apiVersion: postgresql.cnpq.io/v1
kind: ScheduledBackup
metadata:
  name: workshop-scheduled-backup
spec:
  schedule: "0 0 8,12,18 * * *"
  cluster:
    name: workshop-pg-cluster
  immediate: true
  method: plugin
  pluginConfiguration:
    name: barman-cloud.cloudnative-pg.io
```

Object Store

```
apiVersion: barmancloud.cnpq.io/v1
kind: ObjectStore
metadata:
  name: workshop-backup-azure-object-store
spec:
  retentionPolicy: "1d"
  configuration:
    azureCredentials:
      storageAccount:
        name: azure-storage-account-credentials
        key: account_name
      storageKey:
        name: azure-storage-account-credentials
        key: account_key
  destinationPath: >-
    https://cnpqworkshop.blob.core.windows.net/backup/v1
  wal:
    compression: gzip
```

3x daglig backup - kl. 08, 12 og 18.

Støtter Azure, S3 og GCS.

Point-in-Time Recovery (PITR)

Gjenopprett databasen til et vilkårlig tidspunkt

```
bootstrap:  
  recovery:  
    database: app  
    owner: app  
    source: clusterBackup  
    recoveryTarget:  
      backupID: "20251009T072751"  
externalClusters:  
  - name: clusterBackup  
    plugin:  
      name: barman-cloud.cloudnative-pg.io  
    parameters:  
      barmanObjectName: workshop-restore-azure-object-store  
      serverName: workshop-pg-cluster
```

Connection Pooling med PgBouncer

Innebygd støtte for connection pooling

```
apiVersion: postgresql.cnpq.io/v1
kind: Pooler
metadata:
  name: workshop-pg-pooler-rw
spec:
  cluster:
    name: workshop-pg-cluster
  instances: 3
  type: rw          # rw = read-write, ro = read-only
  pgbouncer:
    poolMode: session
    parameters:
      max_client_conn: "100"
      default_pool_size: "20"
```

Lagring

Egne volum for data og WAL

```
storage:  
  storageClass: kubevirt-csi-infra-default  
  size: 2Gi  
walStorage:  
  storageClass: kubevirt-csi-infra-default  
  size: 2Gi
```

Hvorfor eget WAL-volum?

- Bedre I/O-ytelse
- WAL og data konkurrerer ikke om disk
- Mer forutsigbar latency
- Enklere å overvåke plass

Lagringsredundans

For produksjon bør du velge riktig redundansnivå:

Type	Beskrivelse
LRS	Lokalt redundant - kun demo
ZRS	Sone-redundant - minstekrav for prod
GRS	Geo-redundant - ideelt for prod

Tips: Denne workshopen bruker LRS for demo. I produksjon bør du bruke minst ZRS for å tåle sonefeil.

Demo

Workshop-repoet

Oppsett

- 3-instans PostgreSQL-cluster
- PgBouncer RW + RO poolere
- Barman backup til Azure
- PgAdmin for administrasjon
- ArgoCD for GitOps

Testskript

- `populate-database.sql` - Opprett skjema
- `generate-load.sql` - Load test kjør backup her
- `generate-wal-load.sql` - WAL/PITR-test, kjør dette, så drep clusteret
- `cleanup-database.sql` - Rydd opp

Monitoring og observability

Innebygd støtte for Prometheus og logging

Prometheus-metrikk

- Metrics-eksportør på port **9187**
- Ferdiglagde Grafana-dashbord, krever litt jobb men du må ikke lage alt selv.
- Metrikker for:
 - Replication lag
 - Transaksjoner per sekund
 - Tilkoblinger
 - Buffer cache hit ratio

Strukturert logging

- JSON-formatert logging til stdout
- Enkel integrasjon med log-aggregering
- Egnet for ELK, Loki, Splunk osv.

```
{  
  "level": "info",  
  "ts": "2025-10-09T07:27:51Z",  
  "logger": "postgres",  
  "msg": "checkpoint complete",  
  "logging_pod": "workshop-pg-cluster-1"  
}
```

Andre nyttige funksjoner

Deklarative rolling updates

- Automatisk ved minor-versjonsoppgradering
- Ingen nedetid ved operatoroppgradering
- Kontrollert utrulling

Synkron replikering

- Quorum-basert eller prioritetsbasert
- Økt dataholdbarhet
- Konfigurerbart per cluster

TLS-støtte

- Sikre tilkoblinger som standard
- Klientsertifikat-autentisering
- Integrasjon med cert-manager, man kan laste inn egne sertifikater om man vil.

cnpq kubectl-plugin

- Forenkler clusteroperasjoner
- Status, promote, fencing osv.

```
kubectl cnpq status workshop-pg-cluster  
kubectl cnpq promote workshop-pg-cluster ?
```

Oppsummering

Egenskap	CNPG-løsning
Høy tilgjengelighet	Automatisk failover, selvhelbredelse
Backup	Barman Cloud, PITR, planlagte backups
Connection pooling	Innebygd PgBouncer (RW + RO)
Lagring	Egne volum for data og WAL
Sikkerhet	TLS, sertifikater, secrets
Overvåking	Prometheus-metrikk, JSON-logging
CLI	kubectl-plugin, hibernation(dvale), fencing

CNPG gjør det enkelt å kjøre PostgreSQL i produksjon på Kubernetes.

Takk for oppmerksomheten!

Workshop-repo: github.com/cmarter0/cnpg-workshop

CNPG-dokumentasjon: cloudnative-pg.io

Spørsmål?