Stateful Superpowers: Explore High Performance and Scaleable Stateful Workloads on K8s

Alex Chircop, Lori Lorusso,
Alex Reid and Chris Milsted

Why should we be thinking about …

Cloud Native Storage … ?

Why should we be thinking about …

Cloud Native Storage … ?

I mean, seriously …

isn’t everything stateless ?

Cloud Native Storage!

Cloud Native Storage!

… all applications store state somewhere!

Cloud Native Storage!

… all applications store state somewhere!

… so why would we want our storage to be cloud native?

Cloud Native Storage!

Stateful workloads can also benefit from:

- Automation
- Scale
- Failover
- Performance

Cloud Native Storage!

Stateful workloads can also benefit from:

- Automation — Declarative
- Scale — Distributed
- Failover — Self Healing
- Performance — Native, Deterministic

CNS is more than just “Storage” …

Volumes

- Block
- Filesystems
- Shared Filesystems

APIs

- Object
- Databases
- Key-Value

CNS is not just “Storage” …

Volumes

- Block
- Filesystems
- Shared Filesystems

APIs

- Object
- Databases
- Key-Value

Declarative Operations

- K8s integrations: CSI, COSI, Operators
- Day 2: Upgrades, Backups, Failover, DR
- Management: Observability, Security, Encryption
- Scaling & Elasticity

CNCF Storage Whitepaper

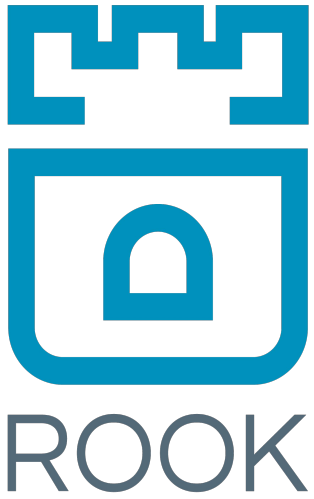
https://bit.ly/cncf-storage-whitepaperV2

- **Attributes** of a storage system, so you can work out what your application needs
- **Layers** of a storage system, so you can understand how they interact and impact the attributes
- **Deployment**, **Management**, **Access Interfaces**

CNCF Storage Projects

KubeCon CloudNativeCon Europe 2025

Graduated



Incubating

Dragonfly

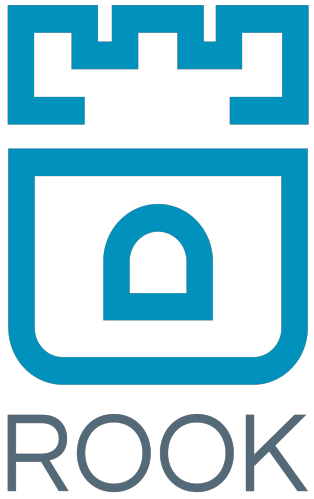
LONGHORN

CNCF Projects:
https://www.cncf.io/projects/
Sandbox Projects:
https://www.cncf.io/sandbox-projects/

CNCF Storage Projects

KubeCon CloudNativeCon Europe 2025

Graduated



Incubating

CNCF Projects:
https://www.cncf.io/projects/
Sandbox Projects:
https://www.cncf.io/sandbox-projects/

KubeCon | CloudNativeCon
Europe 2025

and now …

… the demos!

KubeCon | CloudNativeCon
Europe 2025

DEMO: CloudNativePG

https://github.com/chris-milsted/Kubecon-London-2025-talk/tree/main

The “gold” Standard of stateful applications

Oracle® Grid Infrastructure

Grid Infrastructure Installation and Upgrade Guide

This is a 290 page pdf document to follow…

Case Study CNPG

Operator capability levels

These capabilities were implemented by CloudNativePG, classified using the Operator SDK definition of Capability Levels framework.

Important

Based on the Operator Capability Levels model, you can expect a "Level V - Auto Pilot" set of capabilities from the CloudNativePG operator.

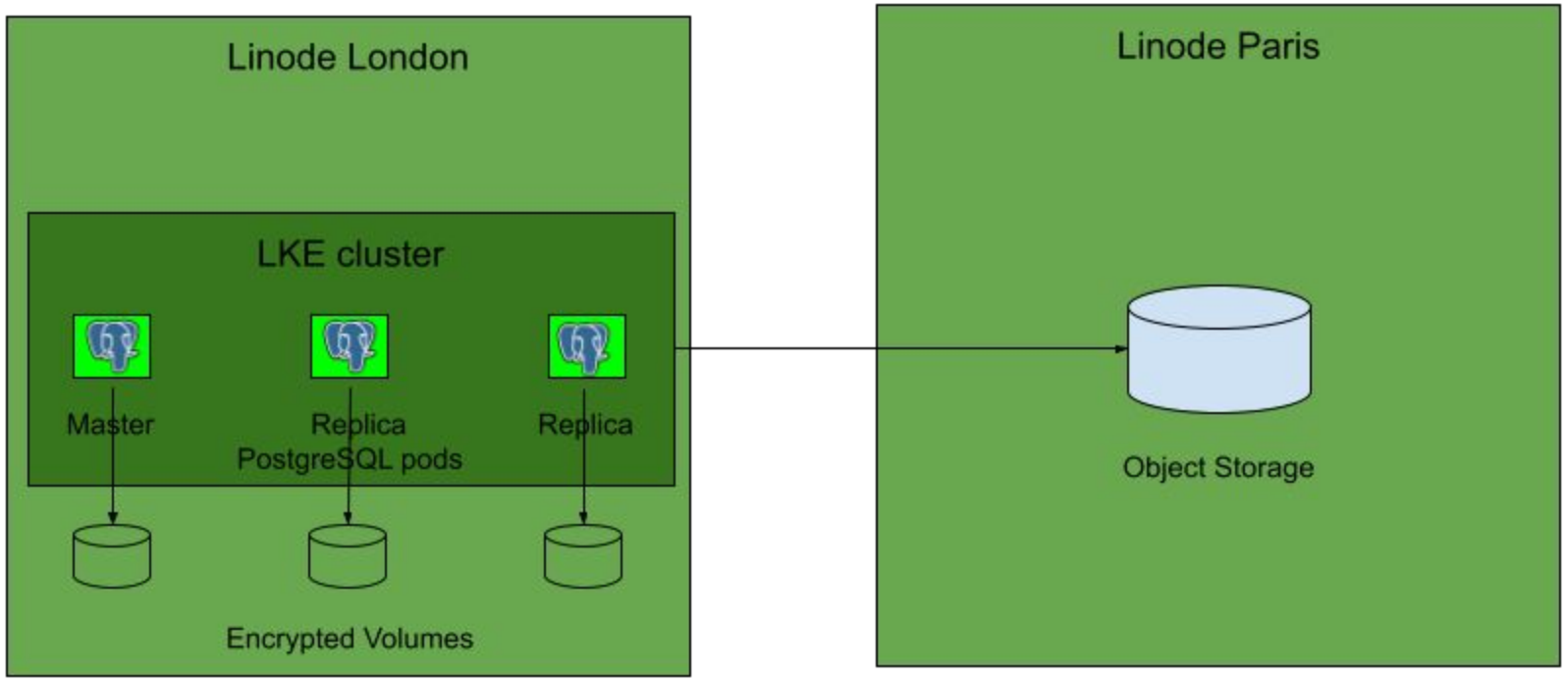
A recipe for a Postgres database deployment!


```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: cluster-kubecon-london
spec:
  description: "Kubecon London Cluster"
  imageName: registry.hub.docker.com/cmilsted/postgresql:17.2
  instances: 3
  startDelay: 300
  stopDelay: 300
  primaryUpdateStrategy: unsupervised
  bootstrap:
        initdb:
        database: app
        owner: app
        dataChecksums: true
        walSegmentSize: 32
        postInitSQL:
        - CREATE DATABASE pgbench OWNER app
  enableSuperuserAccess: true
  postgresql:
        synchronous:
        method: any
        number: 1
        dataDurability: required
  plugins:
  - name: barman-cloud.cloudnative-pg.io
        parameters:
        barmanObjectName: paris-object
        encryption: ""
  storage:
        storageClass: linode-block-storage-retain-encrypted
        size: 10Gi
  resources:
        requests:
        memory: "4Gi"
        cpu: "2"
        limits:
        memory: "6Gi"
        cpu: "4"
  affinity:
        enablePodAntiAffinity: true
        topologyKey: kubernetes.io/hostname
```

90 lines of Yaml

```yaml
apiVersion: postgresql.cnpg.io/v1
kind: Cluster
metadata:
  name: cluster-restore
spec:
  description: "Kubecon London restore"
  imageName: registry.hub.docker.com/cmilsted/postgresql:17.2
  instances: 3
  startDelay: 300
  stopDelay: 300
  primaryUpdateStrategy: unsupervised
  enableSuperuserAccess: true
  superuserSecret:
        name: cluster-kubecon-london-superuser
  bootstrap:
        recovery:
        source: backup-example
  externalClusters:
        - name: backup-example
        barmanObjectStore:
        destinationPath: "s3://paris-bucket/"
        endpointURL: "https://fr-par-1.linodeobjects.com"
        s3Credentials:
        accessKeyId:
        name: backup-creds
        key: ACCESS_KEY_ID
        secretAccessKey:
        name: backup-creds
        key: ACCESS_SECRET_KEY
        region:
        name: backup-creds
        key: REGION
        wal:
        compression: "bzip2"
  storage:
        storageClass: linode-block-storage-retain-encrypted
        size: 10Gi
  resources:
        requests:
        memory: "1Gi"
        cpu: "1"
        limits:
        memory: "2Gi"
        cpu: "2"
  affinity:
        enablePodAntiAffinity: true
        topologyKey: kubernetes.io/hostname
```

What are we deploying today?



Database and restored backup in action

KubeCon CloudNativeCon Europe 2025

```
cmilsted@lon-lp98uiib:~/Documents/Kubecon-London$ kubectl cnpg status cluster-kubecon-london
Cluster Summary
Name                 default/cluster-kubecon-london
System ID:           7479005939425804310
PostgreSQL Image:    registry.hub.docker.com/cmilsted/postgresql:17.2
Primary instance:    cluster-kubecon-london-1
Primary start time:  2025-03-07 10:03:10 +0000 UTC (uptime 1h55m28s)
Status:              Cluster in healthy state
Instances:           3
Ready instances:     3
Size:                262M
Current Write LSN:   0/10000000 (Timeline: 1 - WAL File: 000000010000000000000008)

Continuous Backup status
Not configured

Streaming Replication status
Replication Slots Enabled
Name                     Sent LSN    Write LSN   Flush LSN   Replay LSN  Write Lag  Flush Lag  Repla
y Lag  State      Sync State  Sync Priority  Replication Slot
----                     --------    ---------   ---------   ----------  ---------  ---------  -----
-----  -----      ----------  -------------  ---------------
cluster-kubecon-london-2  0/10000000  0/10000000  0/10000000  0/10000000  00:00:00   00:00:00   00:00
:00    streaming  quorum      1              active
cluster-kubecon-london-3  0/10000000  0/10000000  0/10000000  0/10000000  00:00:00   00:00:00   00:00
:00    streaming  quorum      1              active

Instances status
Name                     Current LSN  Replication role  Status  QoS         Manager Version  Node
----                     -----------  ----------------  ------  ---         ---------------  ----
cluster-kubecon-london-1  0/10000000   Primary           OK      Burstable   1.25.1           lke36550
8-569473-0d18d8520000
cluster-kubecon-london-2  0/10000000   Standby (sync)    OK      Burstable   1.25.1           lke36550
8-569473-0a85de430000
cluster-kubecon-london-3  0/10000000   Standby (sync)    OK      Burstable   1.25.1           lke36550
8-569473-4ca8132b0000

Plugins status
Name                          Version  Status  Reported Operator Capabilities
----                          -------  ------  ------------------------------
barman-cloud.cloudnative-pg.io  0.0.1    N/A     Reconciler Hooks, Lifecycle Service

cmilsted@lon-lp98uiib:~/Documents/Kubecon-London$
```

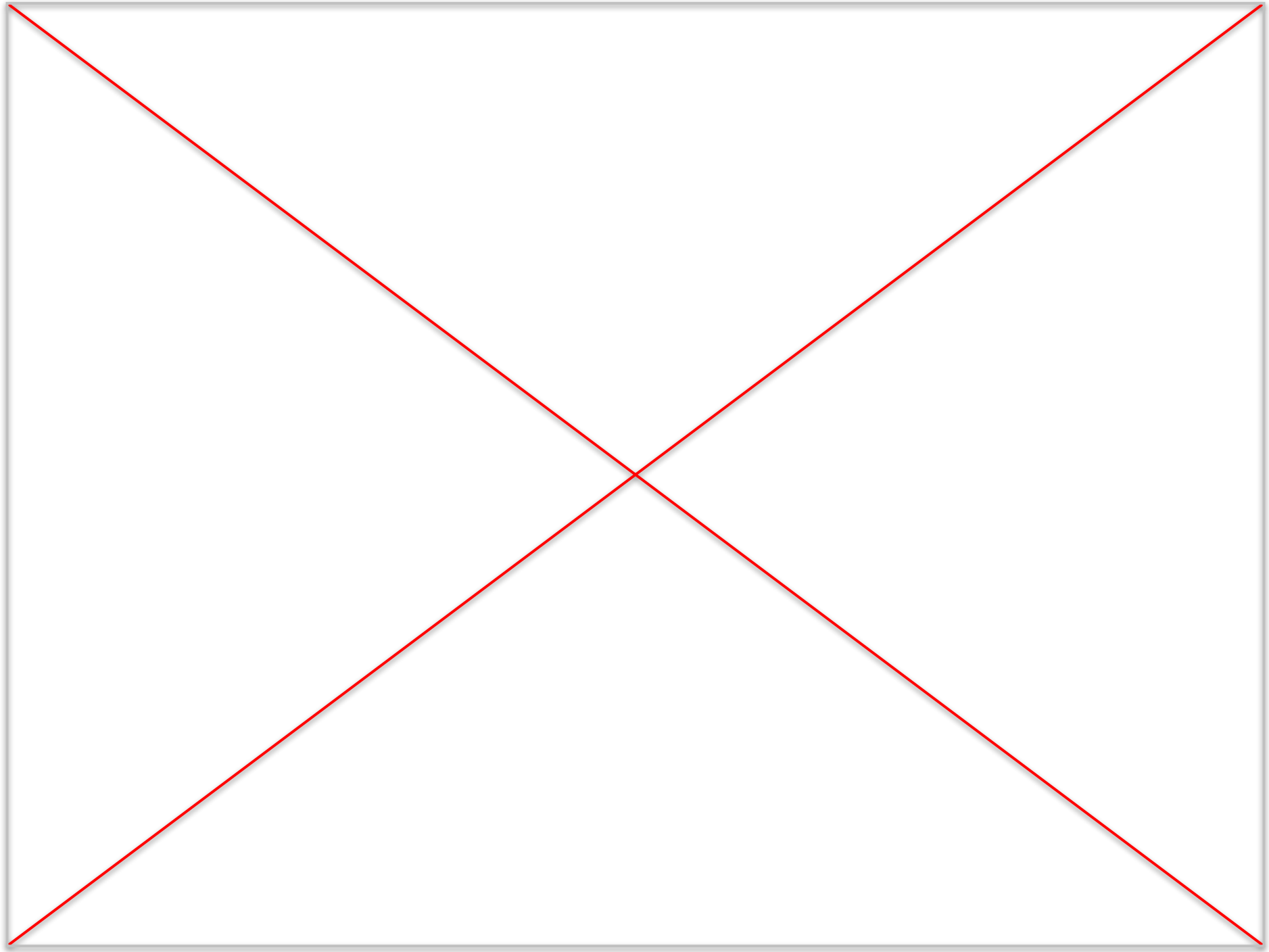
```
cmilsted@lon-lp98uiib:~/Documents/Kubecon-London$ kubectl cnpg status cluster-restore
Cluster Summary
Name                 default/cluster-restore
System ID:           7479005939425804310
PostgreSQL Image:    registry.hub.docker.com/cmilsted/postgresql:17.2
Primary instance:    cluster-restore-1
Primary start time:  2025-03-07 11:51:52 +0000 UTC (uptime 6m59s)
Status:              Cluster in healthy state
Instances:           3
Ready instances:     3
Size:                390M
Current Write LSN:   0/1E000060 (Timeline: 2 - WAL File: 00000002000000000000000F)

Continuous Backup status
Not configured

Streaming Replication status
Replication Slots Enabled
Name              Sent LSN    Write LSN   Flush LSN   Replay LSN  Write Lag  Flush Lag  Replay Lag
State      Sync State  Sync Priority  Replication Slot
----              --------    ---------   ---------   ----------  ---------  ---------  ----------
-----      ----------  -------------  ---------------
cluster-restore-2  0/1E000060  0/1E000060  0/1E000060  0/1E000060  00:00:00   00:00:00   00:00:00
streaming  async       0              active
cluster-restore-3  0/1E000060  0/1E000060  0/1E000060  0/1E000060  00:00:00   00:00:00   00:00:00
streaming  async       0              active

Instances status
Name              Current LSN  Replication role  Status  QoS         Manager Version  Node
----              -----------  ----------------  ------  ---         ---------------  ----
cluster-restore-1  0/1E000060   Primary           OK      Burstable   1.25.1           lke365508-56947
3-0d18d8520000
cluster-restore-2  0/1E000060   Standby (async)   OK      Burstable   1.25.1           lke365508-56947
3-0a85de430000
cluster-restore-3  0/1E000060   Standby (async)   OK      Burstable   1.25.1           lke365508-56947
3-4ca8132b0000

cmilsted@lon-lp98uiib:~/Documents/Kubecon-London$
```



Some of what's included by the operator

- Pod Disruption Budgets
- Pod affinity
- Backup and log archiving
- Block volumes (encryption at rest from the platform)
- Controllers (Reconciliation of state)

And much much more
https://cloudnative-pg.io/documentation/1.25/operator_capability_levels/

- Thoughts for future K8s releases, should there be two kinds of stateful set, one for remote disk and one for local disk…

KubeCon | CloudNativeCon
Europe 2025

DEMO: TiKV

https://github.com/tikv/tikv
https://github.com/healthwaite/tikvbench

What is TiKV?

- Highly scalable, low-latency distributed key-value database
- It provides a raw KV API and an ACID-compliant transactional API
- Easy to deploy in Kubernetes
- TiKV is a Graduated CNCF Project (https://www.cncf.io/projects/)

dailymotion ZaloPay 中国平安 PING AN JD Cloud TiDB Cloud

美图秀秀 转转 Ctrip 携程 tuya 涂鸦智能 JuiceFS

Akamai

TiKV Highlights

Highly scalable

- TiKV scales horizontally to 100+ terabytes of data, billions of keys and hundreds of thousands of RPS.
- Data is split into regions which are balanced evenly among your storage nodes.
- As your capacity/RPS requirements grow you can add more nodes to the TiKV cluster to scale linearly.

Low latency

- Capable of operating at ~1-10 ms latency.
- Based on RocksDB which is very fast.

Cloud native

- It has a fantastic Kubernetes operator to manage: deployment, upgrades, automated failover etc.
- Detailed Prometheus metrics and alerts.
- Pre-configured Grafana dashboards.

Installation

Install the CRDs:

```
$ kubectl create -f \
https://raw.githubusercontent.com/pingcap/tidb-operator/v1.6.1
/manifests/crd.yaml
```

Install the operator:

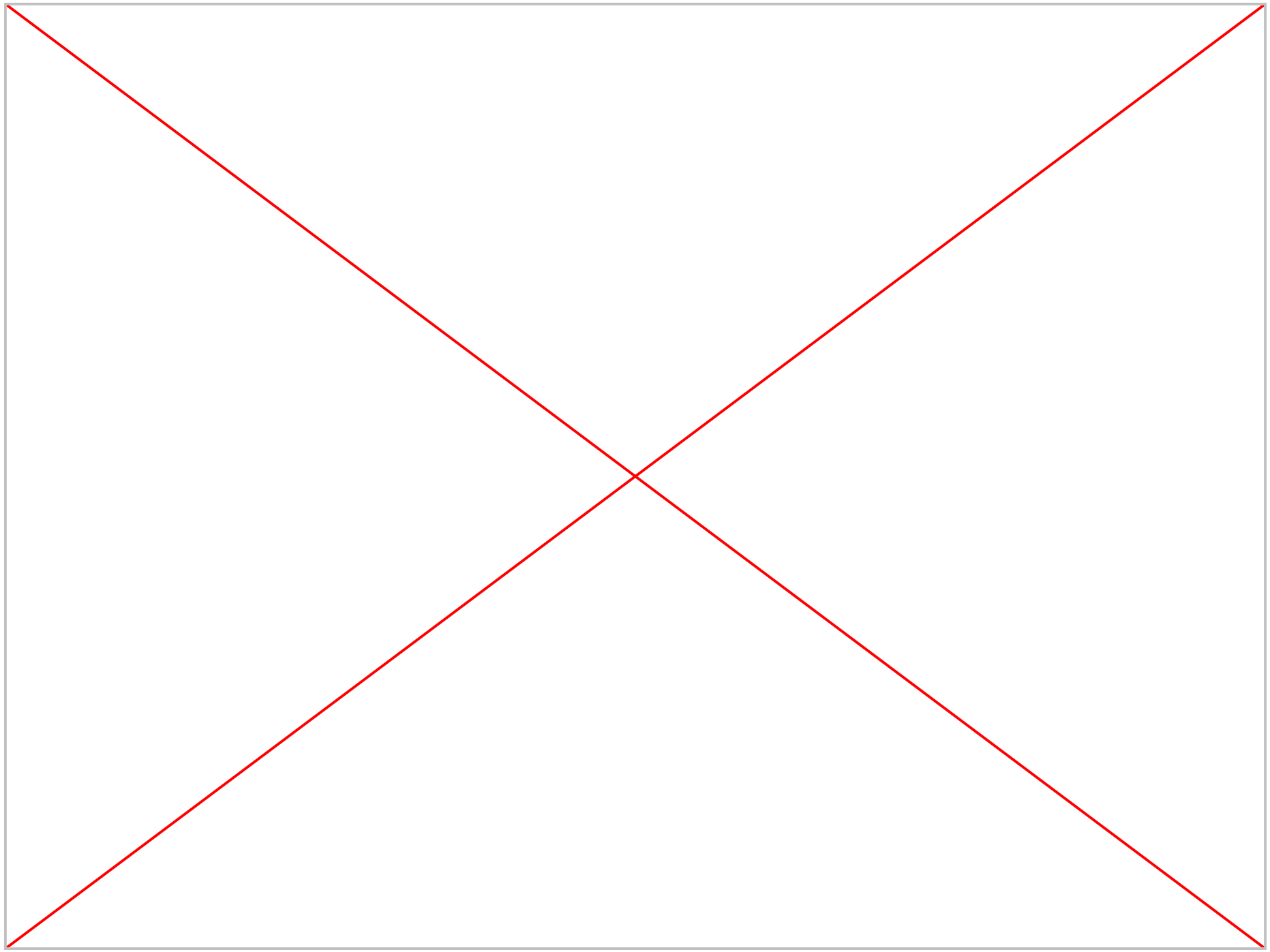
```
$ helm repo add pingcap https://charts.pingcap.org/
$ helm install --namespace tikv-admin tidb-operator
pingcap/tidb-operator --version v1.6.1
```

Install a TiKV Cluster:

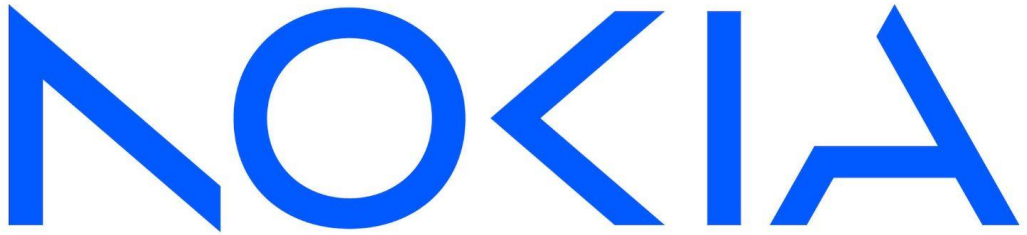
```
$ kubectl create namespace tikv-cluster
$ kubectl -n tikv-cluster apply -f tikv-cluster.yaml
```

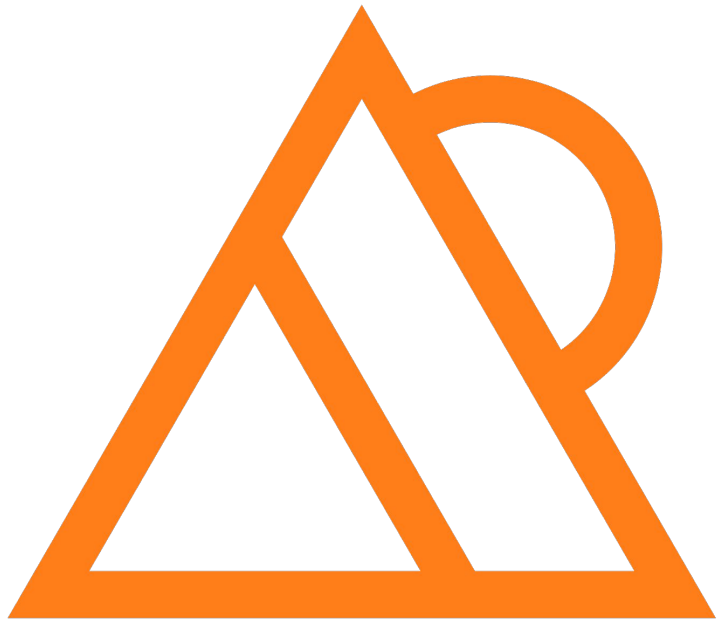
tikv-cluster.yaml


```
apiVersion: pingcap.com/v1alpha1
kind: TidbCluster
  ... snipped ...
  pd:
    baseImage: pingcap/pd
    replicas: 2
    storageClassName: ssd-storage
      ... snipped ...
    config: |
      [replication]
        max-replicas = 5 # 5 copies of the data
        ... snipped ...
  tikv:
    baseImage: pingcap/tikv
    replicas: 12 # 12 nodes for storage
    storageClassName: ssd-storage  # Expose local disks
    ... snipped ...
    config: |
      ... snipped ...
      [readpool.unified]
        max-thread-count = 32 # Optimise for reads
      [server]
        grpc-concurrency = 8 #default is 5
    nodeSelector:
      node-role.kubernetes.io/tikv: "true"
    affinity:
      podAntiAffinity: # Don't schedule the TiKV pods on the same node
        requiredDuringSchedulingIgnoredDuringExecution:
        - labelSelector:
            matchExpressions:
            - key: app.kubernetes.io/component
              operator: In
              values:
              - tikv
          topologyKey: kubernetes.io/hostname
      ... snipped ...
```



Real World Examples





PERCONA®

We believe an open world is a better world. Our mission is to enable everyone to innovate freely, by providing the best open source database software, support, and services.

Nokia NESC :

- 90,000 internal users, 6000 projects
- 5 Data Centers
- 61PB of storage

Main Pain: Operational efficiency

- Lack of the database self-service
- Growing number of microservices

Requirements

- 100% open source
- MySQL and PostgreSQL support

NOKIA

Solution

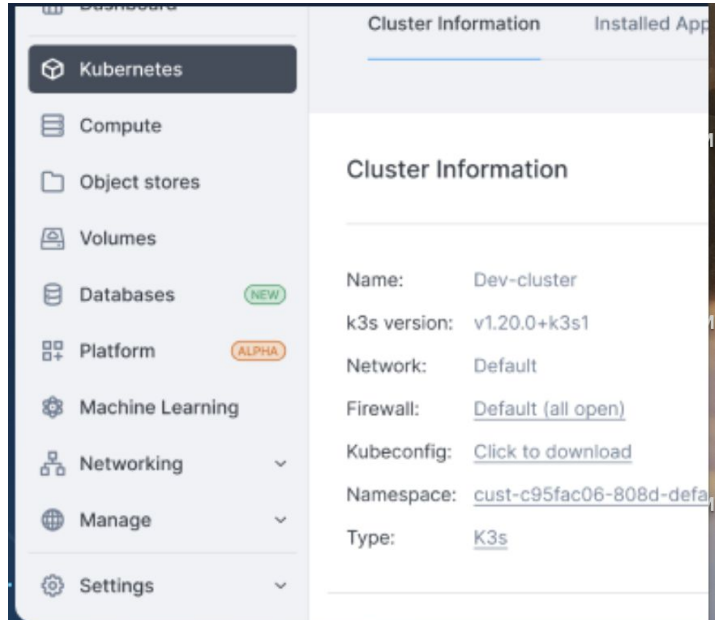
- Decided to run databases on NKS (K8s)
- Used Percona's MySQL and PostgreSQL operators to build a private DBaaS

Allows them to…

- shift databases from virtual environments to Kubernetes
- improve resource utilization and reduce infra costs
- shifts the responsibility for database management left (to dev teams)

Background:

Civo is a cloud-native service provider providing public and private cloud - all on Kubernetes



Problem

- how to launch MySQL and Postgres DBaaS on K8s

Requirement

- reliable and battle-proven database operators
- isolated-tenant and multi-tenant environments support
- open source
- integrated with CIVO Cloud control plane

Solution

- Percona operators for MySQL and PostgreSQL to automate operations in the backend
- Namespaced operator deployments to provide the required separation of tenants

Allows them to…

- launch MySQL and Postgres DBaaS quickly
- keep a cloud-native design approach end-to-end
- contribute to the development of projects

https://www.percona.com/software/percona-operators

Q&A !

Ask us anything!


KubeCon
CloudNativeCon
Europe 2025
LONDON BUS