Streamlining MongoDB Deployments with Kubernetes Operators

Percona Live 2026 - Ivan Groenewold

PERCONA

Agenda

- Intro to PSMDB Operator
- Deployment Modes
- Backup & Restore
- Scaling
- Multi Cluster
- What's new and what's coming

PERCONA

Intro to PSMDB Operator

Streamlining MongoDB Deployments with Kubernetes Operators

PERCONA

Operators

- Extend the capabilities of the Kubernetes API
- Useful for managing applications in Kubernetes
- Control Loop Pattern
- Operators run inside pods

PERCONA®

Percona Operator for MongoDB

- Automates the lifecycle of Percona Server for MongoDB (PSMDB)
- Supports Replica sets or Sharded clusters
- Based on configuration best practices
- Backup & Restore abstractions
- Built-in Observability

PERCONA®

Supported Platforms

- Compatible with any CNCF-certified distribution
- Official builds tested on:
  - Google (GKE)
  - Amazon (EKS)
  - OpenShift
  - Azure (AKS)
  - Minikube

PERCONA®

Resources required

- A Kubernetes cluster (3+ nodes)
- Minimum resources per node:
  - 2 CPU
  - 2 GB of RAM
  - 60GB storage
- If sharding:
  - 4 CPU
  - 6 GB of RAM

PERCONA®

Custom Resource Definitions (CRD)

- CRD extend the standard set of resources which Kubernetes "knows" about
- MongoDB Operator creates these new resources:
  - **`PerconaServerMongoDB`**
  - **`PerconaServerMongoDBBackup`**
  - **`PerconaServerMongoDBRestore`**

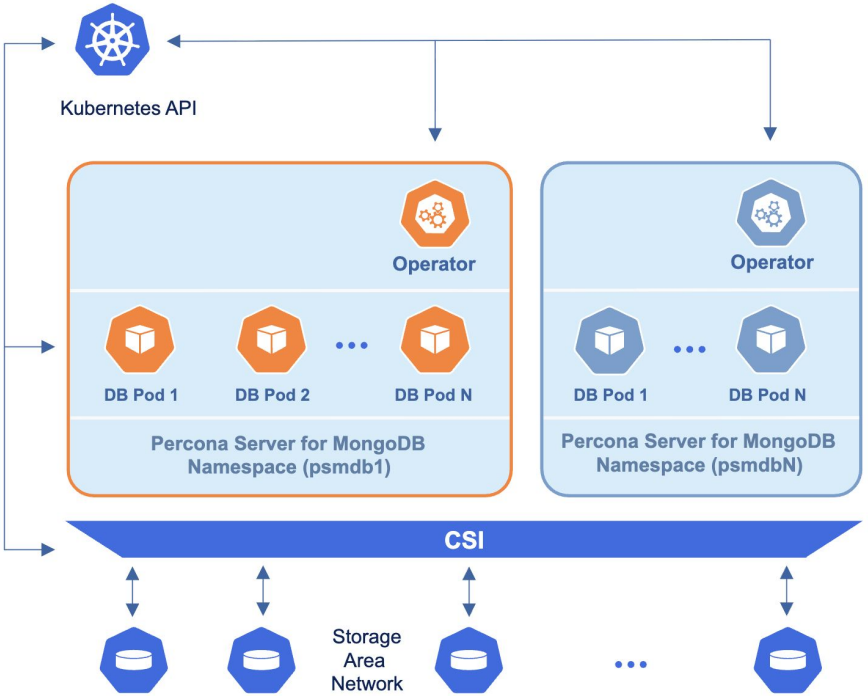
PERCONA®

Deployment Modes

Streamlining MongoDB Deployments with Kubernetes Operators

PERCONA®

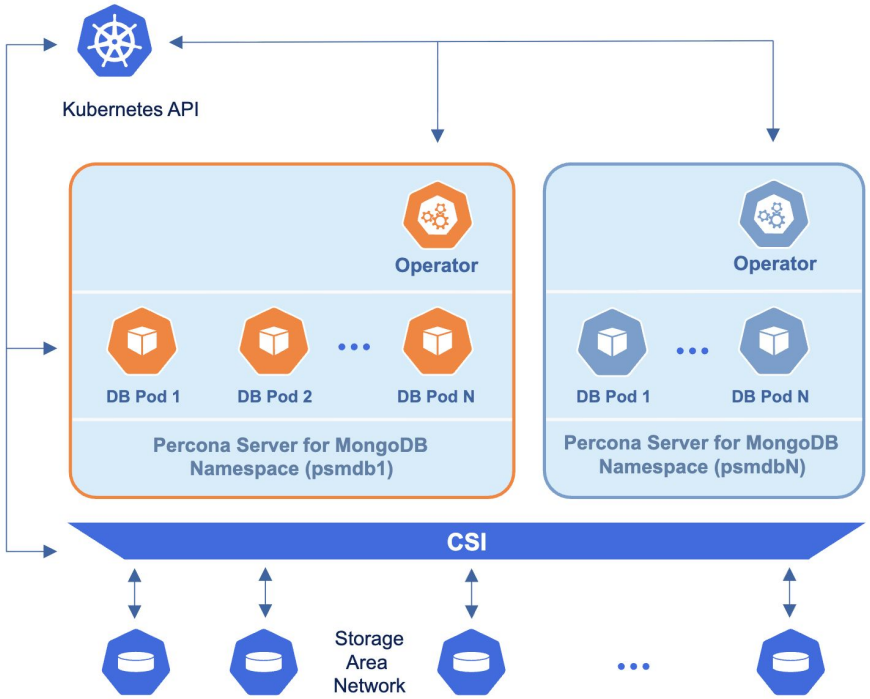
Deployment Mode 1: Single Namespace Mode



PERCONA

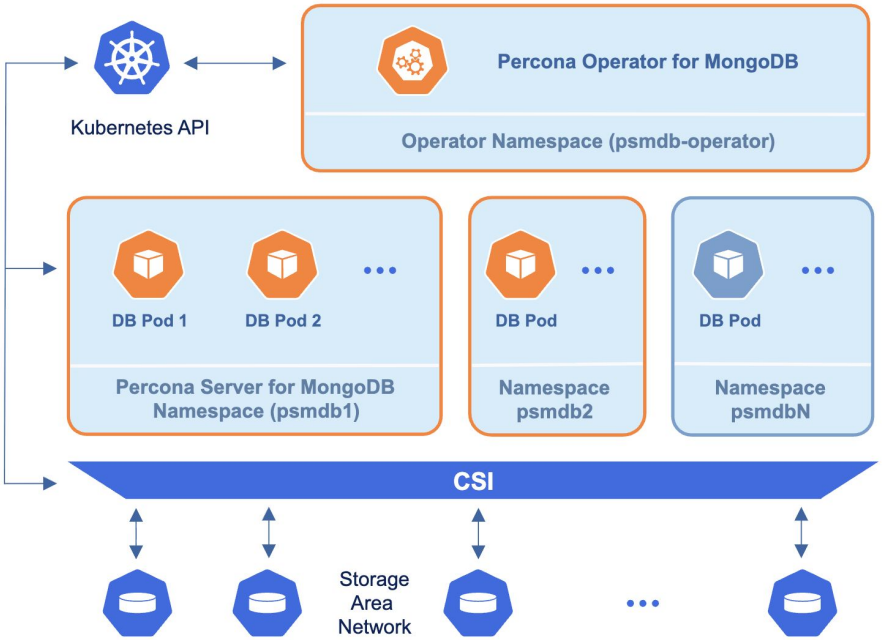
Deployment Mode 1: Single Namespace Mode

- Stronger isolation
- Reduced blast radius
- More operational overhead
- Easier multi-tenancy



PERCONA

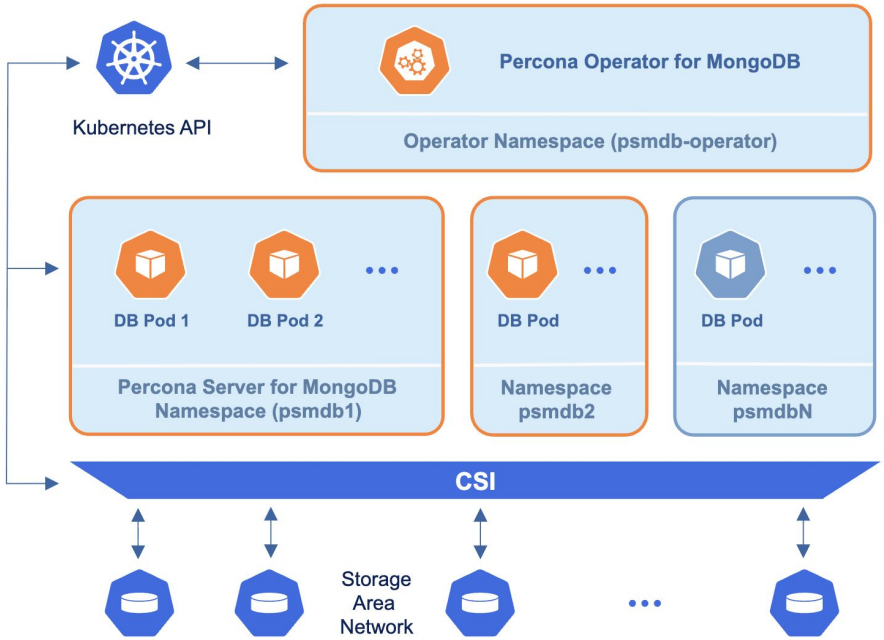
Deployment Mode 2: Cluster-wide Mode



PERCONA

Deployment Mode 2: Cluster-wide Mode

- Weaker isolation
- Larger blast radius
- Simpler operations
- Harder multi-tenancy



PERCONA

Installation

- Available Options:
  - Helm
  - Manual install (kubectl)
- Create a namespace
  - `kubectl create ns percona`
- Create a database
  - Apply a cr.yaml via kubectl or run "helm install"
  - Creates a `PerconaServerMongoDB` object

PERCONA

Backup & Restore

Streamlining MongoDB Deployments with Kubernetes Operators

PERCONA®

Backup & Restore

- Backups are automatically managed with Percona Backup for MongoDB
- You can store backups outside Kubernetes:
  - Amazon S3 (or S3-compatible) Storage
  - Google Cloud Storage
  - Azure Blob Storage

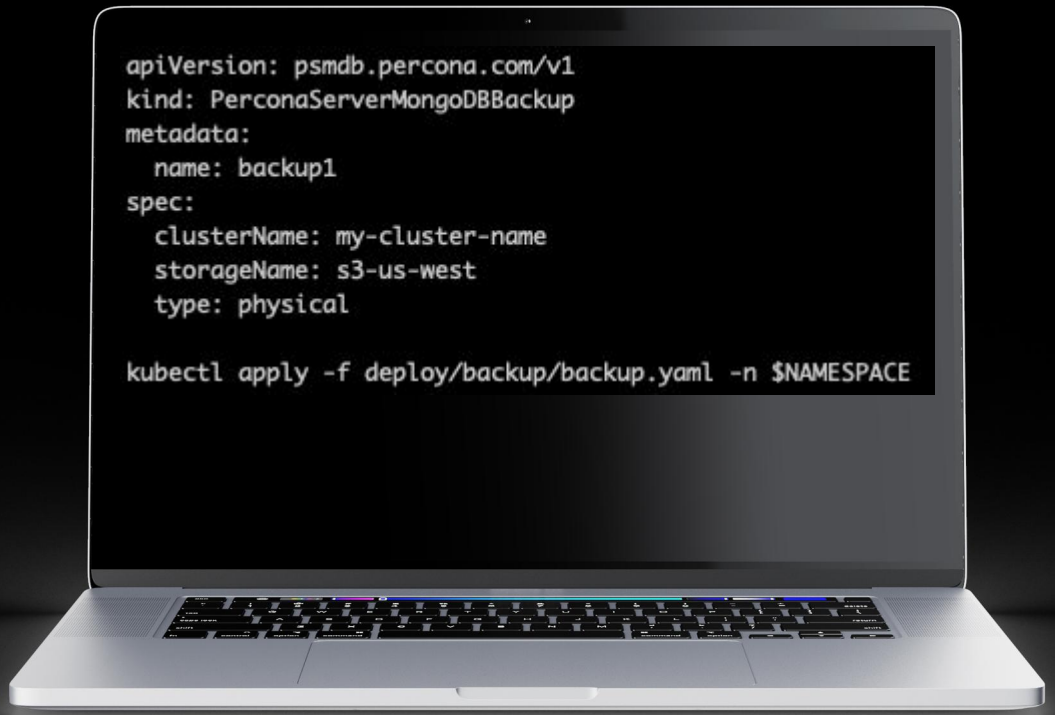
PERCONA®

Backup & Restore (2)

- Available backup types:
  - Logical Backup
  - Physical Backup
  - Incremental Physical backup
- Point-in-time recovery (PITR)

PERCONA®

Example



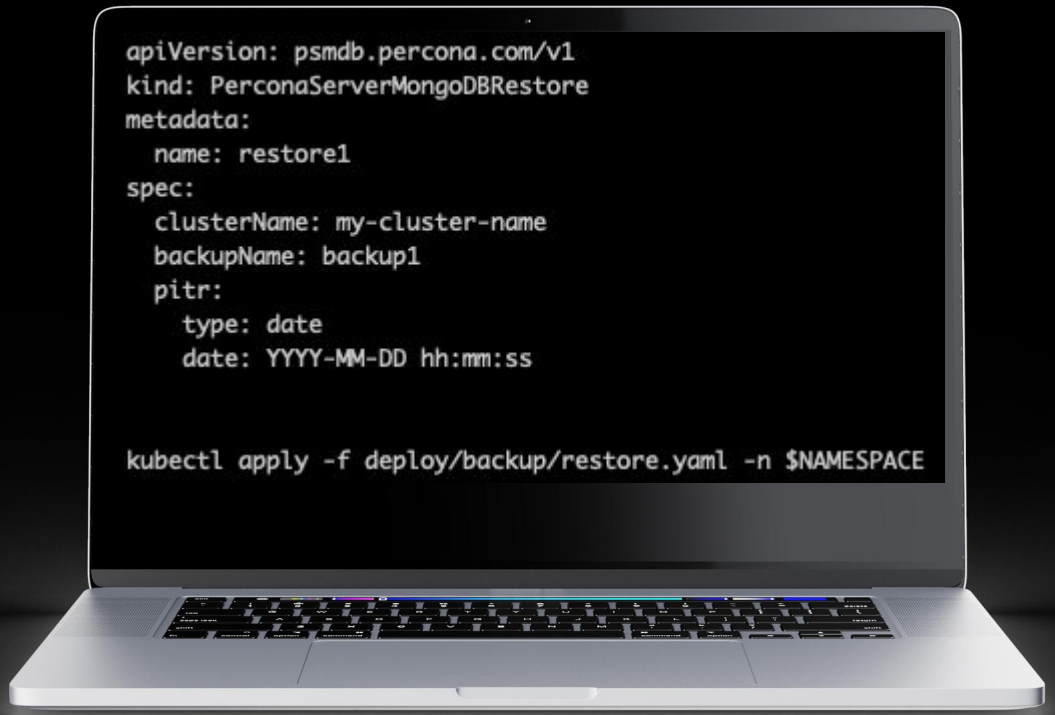
PERCONA

Restore

- Restore can be performed on the same cluster or to a new cluster
  - Logical restore requires the same number or more shards
  - Physical restore requires the exact same number of shards
- Point in time recovery is automatically handled

PERCONA®

Example



PERCONA

Scaling

Streamlining MongoDB Deployments with Kubernetes Operators

PERCONA®

Vertical Scaling

- Manage CPU or memory for every component separately:

```
spec:
  replsets:
    resources:
      requests:
        memory: 4G
        cpu: 2
      limits:
        memory: 4G
        cpu: 2
```

PERCONA

Vertical Scaling

- Automatic resize of PVCs via Volume Expansion capability

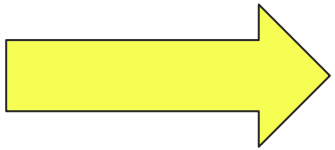
```
spec:
  ...
  enableVolumeExpansion: true
  ...
  replsets:
    ...
    volumeSpec:
      persistentVolumeClaim:
        resources:
          requests:
            storage: <NEW STORAGE SIZE>
```

PERCONA

Horizontal Scaling

- Change the size of your MongoDB replica set:

```
spec:
  ...
  replsets:
  - name: rs0
    size: 3
    ...
```



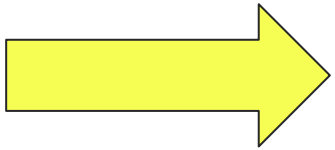
```
spec:
  ...
  replsets:
  - name: rs0
    size: 5
    ...
```

PERCONA

Horizontal Scaling

Change the number of shards by adding or removing members:

```
spec:
  ...
  replsets:
  - name: rs0
    size: 3
    ...
  - name: rs1
    size: 3
    ...
```



```
spec:
  ...
  replsets:
  - name: rs0
    size: 3
    ...
  - name: rs1
    size: 3
    ...
  - name: rs2
    size: 3
    ...
```

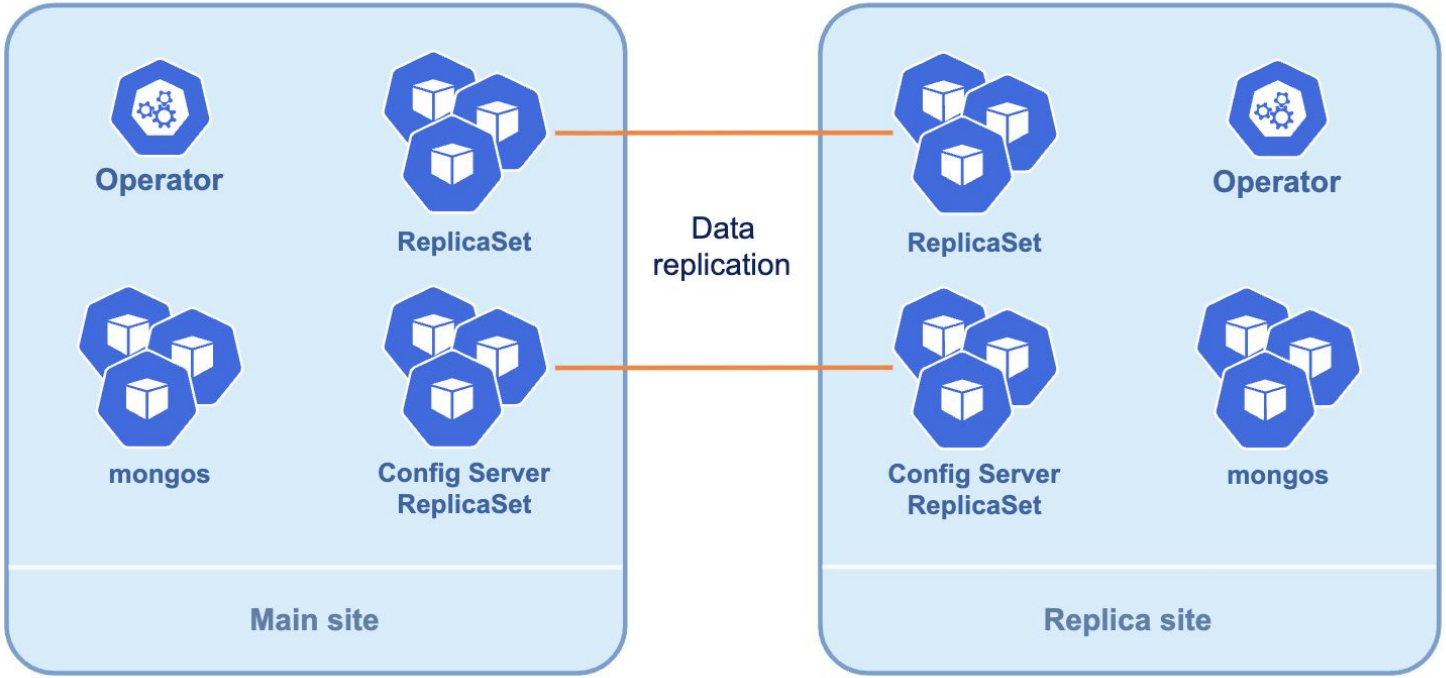
PERCONA

Multi-Cluster

Streamlining MongoDB Deployments with Kubernetes Operators

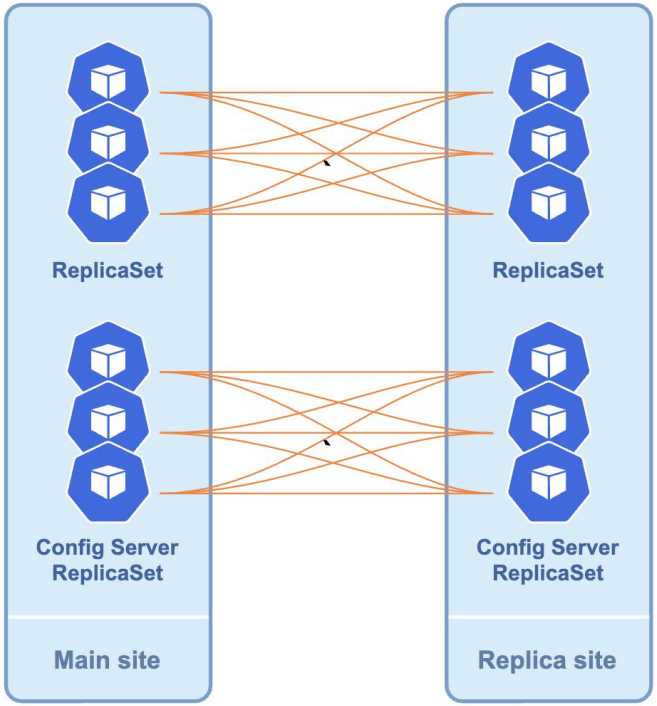
PERCONA®

Multi-cluster



PERCONA

Multi-cluster



PERCONA

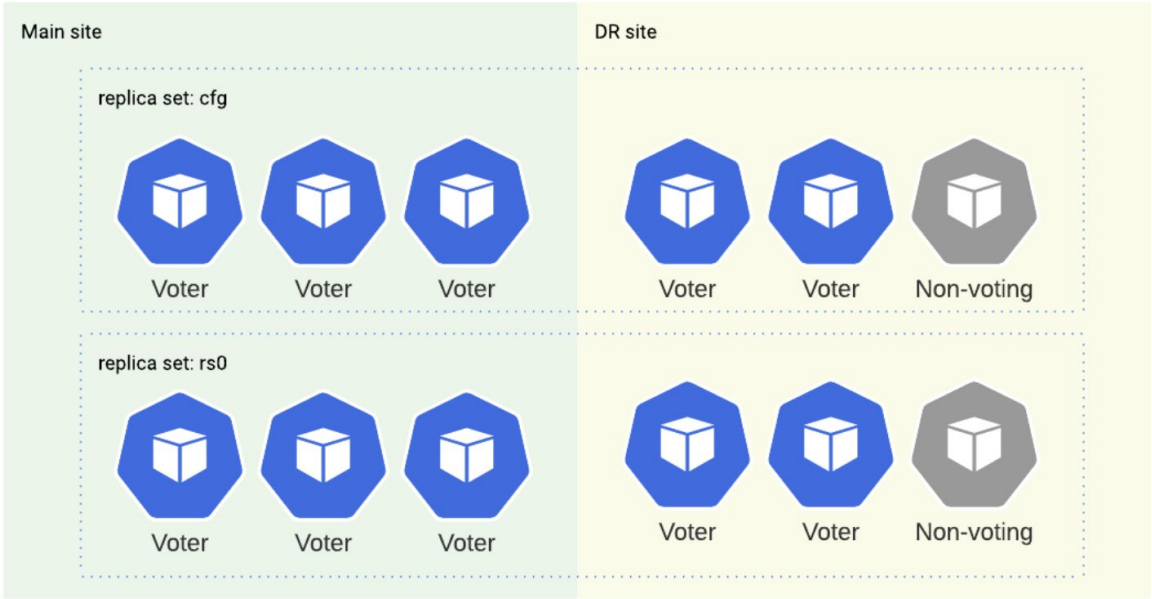
Multi-cluster

- Operator utilizes Kubernetes Multi-Cluster Services (MCS)
- `ServiceExport` and `ServiceImport` objects are created automatically
- The Replica site runs the Operator in "unmanaged" mode

```
multiCluster:
  enabled: true
  DNSSuffix: svc.clusterset.local
```

PERCONA

Multi-cluster typical setup



PERCONA

Multi-cluster

- Graceful Switchover: semi-automatic
  - Make main site "unmanaged", make replica site "managed"
- Failover: manual process
  - Reconfigure the replica set with only the replica site nodes
  - Initial sync will be required for failback
- Backups can be taken on both primary and replica sites

PERCONA

What's new and what's coming

Streamlining MongoDB Deployments with Kubernetes Operators

PERCONA

What's new

- Restore with replica set name remapping
- Minio driver for S3-compatible storage
- Automatic resize of storage based on usage thresholds
- Vault integration for system user password management

PERCONA®

What's new (2)

- Persistent cluster-level logging for MongoDB pods with Fluent Bit
- Concurrent reconciling to process several clusters simultaneously
- Incremental physical backups
- Selective restore from logical backups

PERCONA®

Other cool features

- LDAP Integration
- Data-at-rest encryption
- File-copy-based initial sync
- Audit plugin

PERCONA®

Coming soon...

- Kubernetes Snapshot Backups
- Config shards support
- Workload Identity Authentication
- External arbiters
- Automatic live migration with Percona ClusterSync

PERCONA®

Questions?

PERCONA®


PERCONA®