



## Percona is cloud-native

Percona Operator for MongoDB provides:

- Consistent deployment configuration
- Build once, use multiple times
- Reliably deploy across environments with confidence and ease

Percona Operator for MongoDB provides the same configuration regardless of hosting environment.

This enables:

- Efficient scaling
- Simpler benchmarking
- Stable testing environments

## What is a Kubernetes Operator?

Kubernetes Operators provide a way to package, deploy, and manage a Kubernetes application. A Kubernetes application is deployed on Kubernetes and managed using the Kubernetes APIs and tooling. Operators help in building cloud-native applications by delivering automation advantages like deploying, scaling, and backup and restore while being able to run anywhere Kubernetes is deployed.

Kubernetes Operators are not just a tool for managing containers. They enable you to easily balance resources against demand and provide an avenue for consistent and reliable deployment of new services. With Kubernetes Operators, you are assured that your new environments are consistent across cloud providers and on-premises environments.

## Percona Operator for MongoDB

One of the best features of **Percona Server for MongoDB** is that it can easily scale to meet your needs. It would be great if there was a way to ensure consistency across all the nodes and even across multiple environments. With Percona Operator for MongoDB, you have that tool.

Percona Operator for MongoDB is based on the Kubernetes API and enables a highly available Percona Server for MongoDB replica set. Regardless of where it is used, the Operator creates a Percona Server for MongoDB node that is identical to other Percona Server for MongoDB nodes created with the same Operator. This provides an assured level of stability to easily build appropriate test environments or deploy a repeatable, tested database environment that meets Percona Expert-recommended best practices, either on-premises or in different cloud providers. Operators enable scalable and repeatable deployments.

One of the major benefits of Percona Operator for MongoDB is that it is based on best practices for configuration of a Percona Server for MongoDB environment. The Operator is easier and faster to deploy, and incorporates best practices for use of Percona Server for MongoDB.

Percona Operator for MongoDB delivers a tool that can be customized for use in most any environment. Once configured, the Operator can be used to instantiate new Percona Server for MongoDB nodes quickly, easily, and efficiently.

When you want to add a node to an existing replica set or start the building of a new replica set, use the Operator to configure your node(s). The nature of the Kubernetes Operator means that you do not need to make any new decisions; all replica set members are built out the same as each other member created from the same Operator.

The ease of deploying a new replica set or scaling an existing set is evident. Since all members are equal, you don't have any concerns about a misconfiguration negatively impacting your entire replica set. The Operator contains everything needed to build out identical members, no matter if they are on-premises, in the cloud, or in different cloud environments.

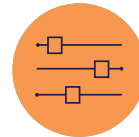
Knowing that your configuration is the same, regardless of environment, enables hybrid environments and multi-cloud environments. Multi-cloud environments are becoming increasingly popular as a way to minimize the impact of an outage event at a specific provider and a method to avoid vendor lock-in.

## Supported features




- **Deploy easily** – create a Percona Server for MongoDB environment with no single point of failure and the ability to span multiple activity or availability zones.
- **Scale your replica set** – change the size of the replica set by altering the size key in the Custom Resource options configuration.
- **Add monitoring** – Percona Monitoring and Management (PMM) can be easily deployed to monitor your Percona Server for MongoDB replica set. The recommended installation process uses helm, the package manager for Kubernetes.
- **Manage your backups** – automate backups using the deploy/cr.yaml file or perform an on-demand backup at any time. Both backup methods use Percona Backup for MongoDB and support simple restores.
- **Rely on ProxySQL to remove single point of failure** – rely on ProxySQL's native clustering with Kubernetes anti-affinity rules to deploy an environment that removes any single point of failure.
- **Automate node recovery** – use the self-healing capability to automatically recover from failure of a single Percona Server for MongoDB node.
- **Provide data encryption** – rely on support for data encryption in transit.
- **Support private container registries** – access information from a private registry to enhance security.


## Common configurations options




- **Set node as arbiter** – set a node as an arbiter, which participates in elections for a new primary node but does not store any data. This reduces storage costs while maintaining replica set integrity.
- **Specify IP address for nodes** – rather than have dynamic IP assignment upon restart, configuration options provide the ability to assign a dedicated IP address to a node.
- **Use existing file or directory for data** – mount existing data from a node's filesystem into the Pod.



**Access Percona Operator for MongoDB Documentation**

Documentation and sample Operator files are available here. 

*All Percona software is fully open source and available to download at no charge.*



## Contact Us

For organizations interested in using Percona Operator for MongoDB for their deployment needs, Percona offers Support and Consulting services.