Becoming Cloud Native:
How Percona Brings Databases to Kubernetes Using Operators

Tyler Duzan
Percona

PERCONA
LIVE EUROPE
AMSTERDAM

Who Am I?

- My name is Tyler Duzan
- Product Manager for MySQL Software and Cloud at Percona
- Prior to joining Percona, I was an operations engineer for 13 years focused on cloud, security, and automation

What are Kubernetes Operators?

- Operators are Cloud Native Automation
  - Automates the full operational lifecycle of an application, such as deployment, backup, restore, failover, upgrades, and eventually destruction.
  - Provides a context for managing all of the infrastructure resources for your application.
- Operators provide a Custom Resource Definition (CRD) for your application
  - CRDs extend the Kubernetes API and allow you to provide custom API functionality via Controllers.
  - Common Controllers are ClusterController, BackupController, RestoreController, UpgradeController.

Challenges and Solutions

Scheduling and Failover

- Kubernetes provides very limited guarantees around the availability of cluster resources.
- Pods can be evicted, pre-empted, rescheduled, or fail at any moment in time.
- Fundamentally, Kubernetes is built to schedule resources with the expectation that the applications built on top of it are stateless in nature.
- Of course, as with any system, failures can also occur at the network, storage, or host level.

Scheduling and Failover

- StatefulSets became available (GA) in Kubernetes 1.9.
- StatefulSets provide guarantees about the ordering of Pods and their uniqueness.
- StatefulSets can help to manage Pod attached resources such as PersistentVolumeClaims (PVCs).
- However, StatefulSets have no awareness or interaction internal to the application being run within its context. It handles Kubernetes resources on behalf of the application only.
- Operators became available in April 2018, and really usable since Kubernetes 1.11.3.

Scheduling and Failover (PXC)

- PXC includes ws_rep (Write Set Replication) and Galera, which provides synchronous replication.
- Additionally, PXC uses Percona XtraBackup (xbstream) to perform Snapshot State Transfer (SST).
- Between Kubernetes remapping data volumes (PVCs) as part of StatefulSet, we can also recover from a single Pod having a total disk failure via SST.
- All members of the PXC cluster contain identical consistent data thanks to ws_rep, ensuring that any given member can fail or be rescheduled without data loss.

Scheduling and Failover (PXC)

- ProxySQL 2.0 includes Native Clustering, ensuring that all ProxySQL instances contain the same configuration and view of PXC cluster state.
- ProxySQL allows us to direct traffic and is SQL-aware, which means if a PXC cluster member fails and is in recovery, ProxySQL will ensure no downtime to the end application.
- When a failure occurs, the ClusterController updates the state within ProxySQL and polls the Kubernetes API to be informed when the Pod becomes rescheduled so it can query it directly for its PXC state.
- Once a failed member becomes in the “Primary” state, it can begin taking traffic and the ClusterController updates the state in ProxySQL.

Scheduling and Failover (PSMDB)

- PSMDB provides replica set, a form of leader/follower asynchronous replication.
- The replication protocol provides for a method to perform initial sync for new members, as well as handle elections of new leaders (primary).
- If a new member is added to a replica set and has a recent enough copy of data (within the window of the oplog on the primary), it can catch up.
- Kubernetes manages remapping data volumes (PVCs), which reduces the time to catch up if a new member is added on a failure.
- All members of a replica set contain identical data (within some bounds), reducing the risk of data loss from Pod failure.

Disk Persistence

- One of the main challenges of having a stateful application is that you must persist data to disk and that disk needs to be available to the application, and follow the application when rescheduled.
- Kubernetes now provides PersistentVolumes and PersistentVolumeClaims using the Container Storage Interface (CSI) and StorageClasses.
- PersistentVolumeClaims bind a PersistentVolume directly to a particular Pod.
- When a Pod is rescheduled, its PVs follow it, meaning that in some cases a failover event takes seconds due to only requiring IST.

Backup and Restore (PXC)

- Backups need to be able to be scheduled or run on-demand, but without requiring heavier container images.
- Backups can take an arbitrary amount of time to run
- For backups to be useful for disaster recovery, they need to be stored outside the context of the Operator itself.
- Need for object storage support, and robust MySQL backup support.
- Percona XtraBackup provides full backup capabilities for MySQL.
- Since version 2.4.14 we now support S3-compatible APIs in xbcloud.
  - Streaming backups to MinIO, S3, and Google Cloud Storage.

Backup and Restore (PXC)

- Restores require rebuilding the PXC cluster entirely, as all members must have consistent data.
- Restores require having data pre-written to disk on initial cluster startup.
- Additional members can be added after the first member by using SST.
- Restores are not zero down-time.
- Point In Time Recovery (PITR) requires a source for binary logs outside the context of the PXC cluster.
  - Not yet supported, but is coming.
  - Likely will be based around using MySQL Ripple for a log source.
  - Will support streaming chunked binary logs to S3-compatible APIs.

Backup and Restore (PSMDB)

- Backups need to be able to be scheduled or run on-demand, but without requiring heavier container images.
- Backups can take an arbitrary amount of time to run
- For backups to be useful for disaster recovery, they need to be stored outside the context of the Operator itself.
- Need for object storage support, and robust MongoDB backup support.
- The newly launched Percona Backup for MongoDB is fully integrated into the operator and provides advanced backup capabilities for MongoDB and native support for object storage compatible with the S3 APIs.
- Similar constraints exist currently for restores as the constraints which PXC operates under.

Certificate Management

- Certificate management is complicated and not well-solved in the Kubernetes ecosystem.
- There is still no strong standard on how to achieve this.
- We assessed these four different methods:
  - Kubernetes Secrets API
  - OpenShift Certificate Manager
  - Cert-Operator by Red Hat OCP Group
  - Cert-Manager by JetStack
- We support both Kubernetes Secrets API and Cert-Manager.
- Starting with operator version 1.2.0 we now automatically generate Kubernetes Secrets if Cert-Manager isn’t available to configure TLS.

Current State

Simple PXC Cluster Deployment

- Takes about 2 minutes to deploy with default CR.
- 3-member PXC cluster.
- 3-member ProxySQL 2.0 Native Cluster.
- PXC + ProxySQL integration.
  - Uses new Galera integrations in ProxySQL 2.0.
  - Set up in read/write splitting mode.
  - ProxySQL is a component of automating failover.
- TLS enabled by default, using Cert-Manager or auto-generated certs in Kubernetes Secrets.
- Data-at-rest encryption coming in a future release.

Simple PSMDB Cluster Deployment

- Takes about 2 minutes to deploy with default CR.
- 3-member PSMDB replica set.
- Makes use of general replica set features to help with automating failover.
- Data-at-rest-encryption is fully supported in PSMDB and enabled by default.
- TLS enabled by default, using Cert-Manager or auto-generated certs in Kubernetes Secrets.

Automated Backups and Restores

- Backup and Restore use Percona XtraBackup or Percona Backup for MongoDB.
- Supports using a PVC for backups within the context of the Operator.
- Supports using external object storage with S3-compatible APIs.
  - Can build object storage into your Kubernetes environment with MinIO.
- Backups can occur on a schedule or on-demand.
- Restores trigger a cluster rebuild, utilizing SST or initial sync.
- PITR not currently supported, but coming in a future release.

Complex Kubernetes Features

- Full support for the use of Pod Disruption Budgets.
- Supports nodeSelector, Topology Key, Constraints, Tolerations, and Priority Classes.
- Via affinity/anti-affinity can be configured for Multi-AZ deployments.
- Via Constraints, Topology Key, or nodeSelector can be limited to specific host-types.
- Supports ConfigMaps for custom MySQL/MongoDB configuration.
- Supports local storage for higher IO performance via emptyDir and hostPath methods.
- Uses Kubernetes Secrets API for users.

Strong Integrations

- Both operators feature a Service Broker for OpenShift, and are certified for Red Hat OpenShift.
- Both operators are certified on VMware/Pivotal PKS.
- Both operators feature integration support for newly launched PMM2.
- We build on top of StatefulSet RollingUpdate to provide support for zero-downtime upgrades and zero-downtime configuration changes for both operators.

Demo

```
bash-5.0$
```

What’s Next?

Roadmap Sneak Peek (PXC)

- Automatic management of MySQL User
- Support for Data-at-Rest Encryption
- Auto-tuning MySQL configuration based on Pod resources
- Point-in-time-Recovery (PITR) features
- PXC 8.0

Roadmap Sneak Peek (PSMDB)

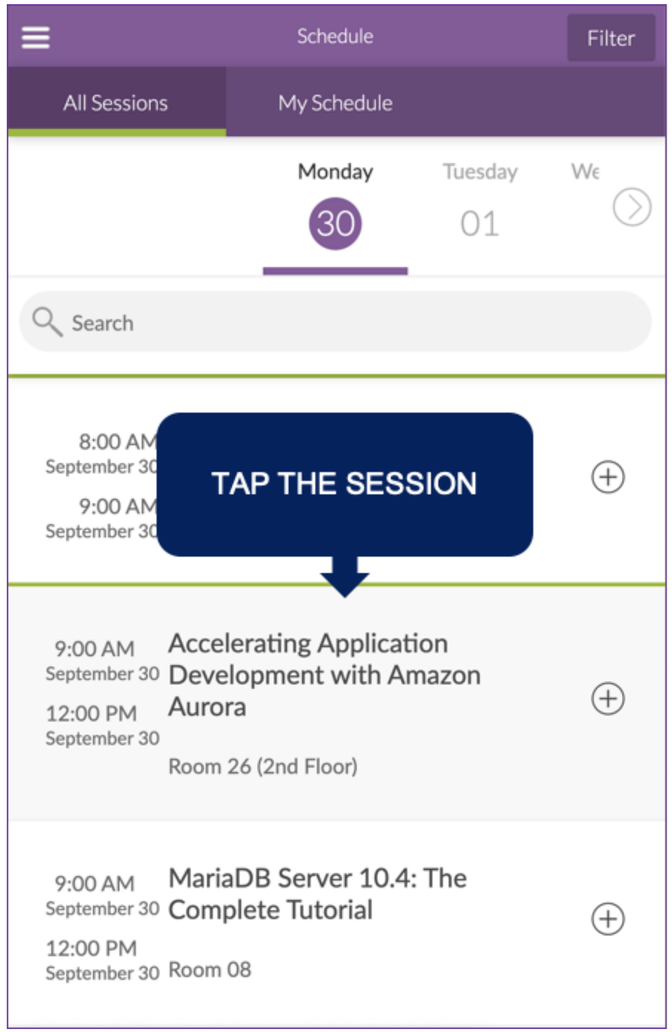
- Robust Sharding Support, with self-healing and management of all ReplSets which are part of the sharded cluster (big feature!)
- Automatic management of MongoDB users

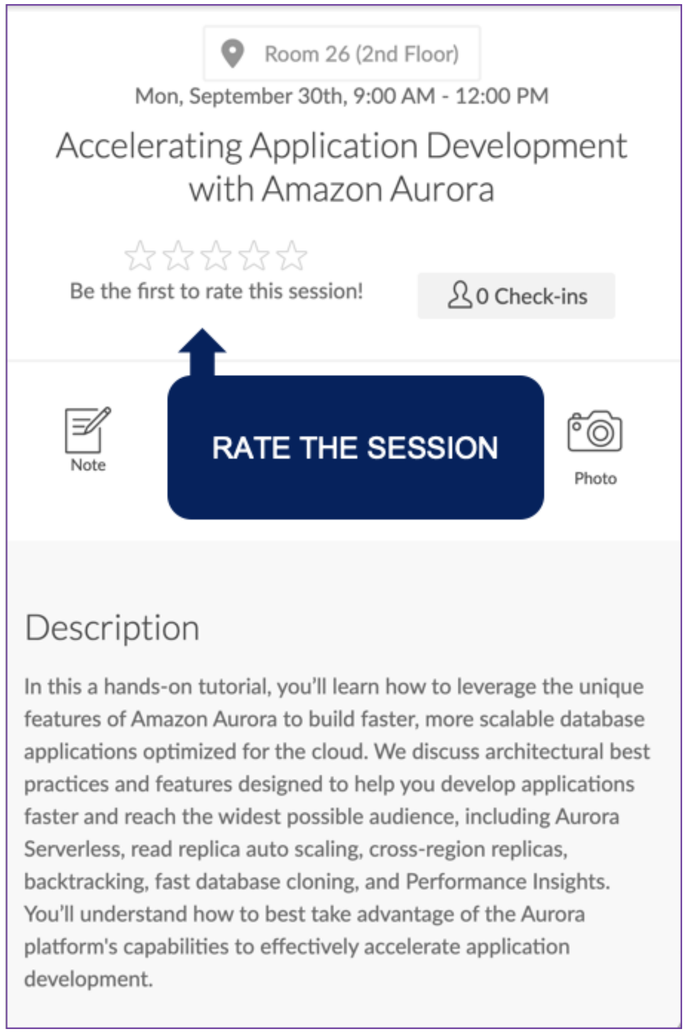
Other Things Coming

- Operator for Percona Distribution for PostgreSQL.
- Simplified CLI tool and accompanying API for managing databases across multiple Kubernetes environments using our operators

Questions?

Rate My Session



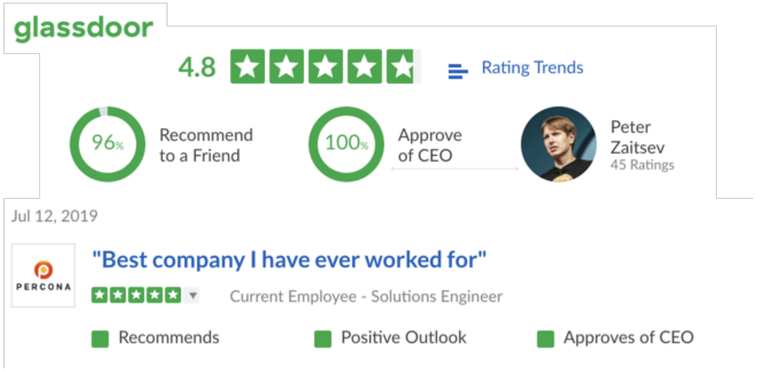


We’re Hiring

Percona’s open source database experts are true superheroes, improving database performance for customers across the globe.

Our staff live in nearly 30 different countries around the world, and most work remotely from home.

Discover what it means to have a Percona career with the smartest people in the database performance industries, solving the most challenging problems our customers come across.



Thank you Sponsors!

Gold

Silver

Community

Thank You