



Migrating MySQL to Kubernetes

"What fun!" -Nobody

A large, light orange graphic on the left side of the slide consists of a stylized letter 'A' and a stylized letter 'P' overlapping each other. The 'A' is formed by two diagonal lines meeting at a point at the top, and the 'P' is formed by a circular top and a vertical stem. The background is a solid, vibrant orange color.

Introductions

"Who is your daddy, and what does he do?"

Matthew Boehm

- Principal Architect (MySQL) @ Percona, 12 yrs
- Consultant in Professional Services
 - <Redacted>, Inc. – Fortune 100
 - <Redacted>, Corp. – Fortune 10
 - <Redacted>, LLC. – Fortune 200
 - <Redacted>, Inc. – Fortune 10
 - <Redacted>, Inc. – Fortune 100
- Lead Trainer, <https://percona.com/training>
 - Operations and Troubleshooting
 - Scaling and Optimizations
 - ProxySQL
 - Group Replication
 - Percona XtraDB Cluster





Gameplan

(ie: table of contents)

Gameplan

- What is Kubernetes?
- What is an Operator?
- Creating our MySQL Cluster
- Logical export/import of MySQL
- Physical export/import of MySQL

Quick Survey of Hands

- Who is still on MySQL 5.7?
 - What's holding you back from MySQL 8.0?
- Who here has...
 - ... heard of Percona Xtrabackup?
 - ... used Xtrabackup?
 - ... used something other than Xtrabackup?
- Ready for 8.4?

Just in Case You Didn't Know

- All Percona software is 100% free and open source!
- MySQL
- PostgreSQL
- MongoDB

Just in Case You Didn't Know 2

- Percona Xtrabackup is the leading tool for taking hot, online, physical backups.
- Quick Feature List
 - Hot, Online (ie: no locking/blocking)
 - Incremental / Differential
 - Direct-to-compressed
 - Streaming (ie: no local disk required)
 - Encrypted Tables Support

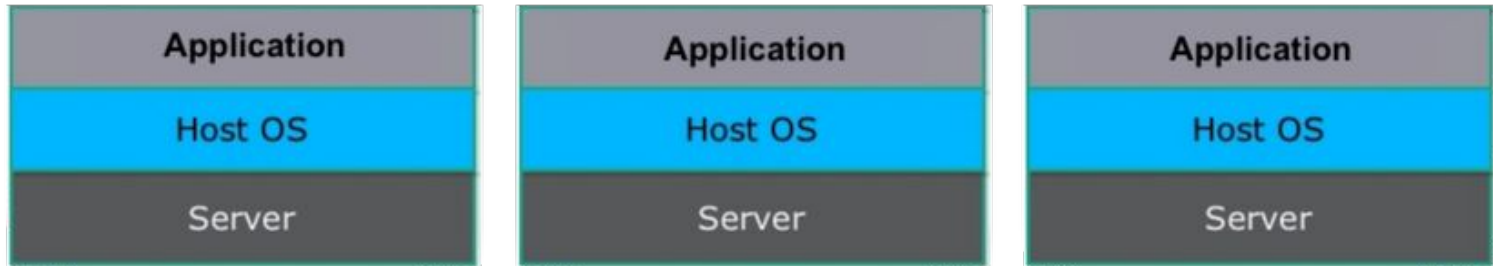


What is K8S?

Kubernetes has 8 letters in the middle

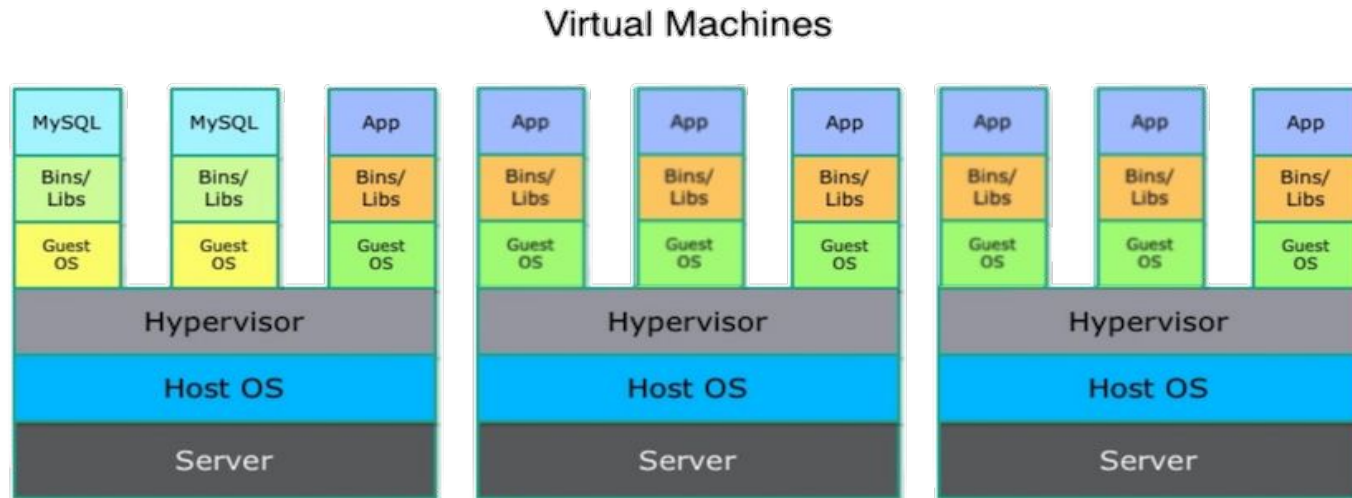
In the Beginning...

- ... there were physical servers
 - Operating Systems
 - Applications
- Scaling == Add more hardware
- Inefficient resource usage

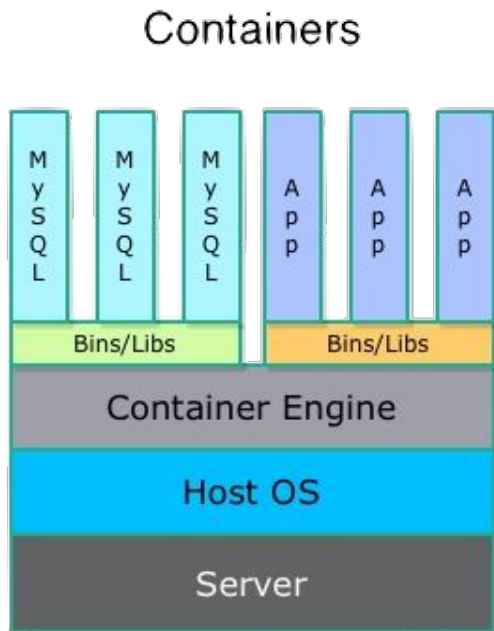


Virtual Machines

- Simulate physical machine
- Provide local file system
- Accessible over network
- Full/independent OS ("guest os")
- Virtualized device drivers
- Resource and memory management
- Requires a hypervisor



Containers (Docker)



- Opensource project built on LXC
- Portable container engine that can package applications and dependencies as "images"
- "git-like" capabilities for tracking versions of each container
- Build new container using others as base
- Ecosystem for sharing pre-build containers
- The "easy button"

Simple Enough for Single Use

- That's easy for 5-10 containers on a single host
 - ...but what about 50-100 containers...
 - ...on 20 different hosts!

You need an orchestrator!

Container Orchestration

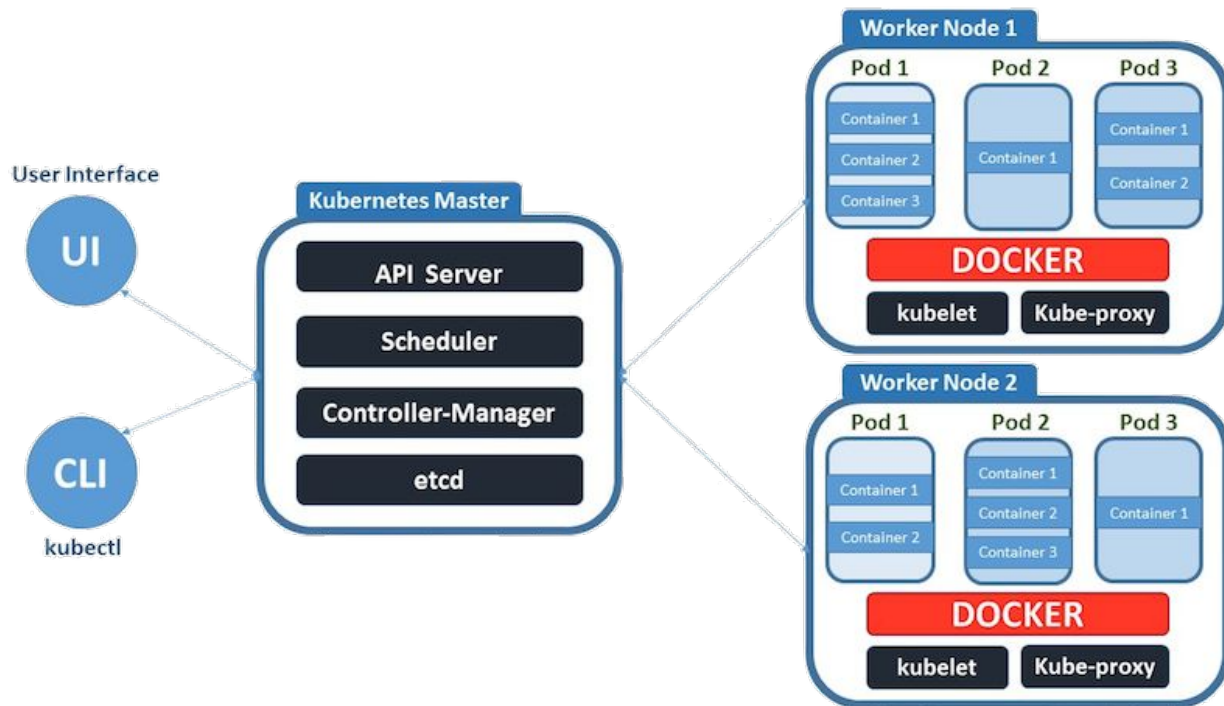
- Now you have to run hundreds of containers
 - across, potentially, hundreds of hosts
- Health checks on the containers
- Launching X copies for a particular container
- Scaling the number of containers up and down depending on load
- Performing rolling updates across containers
- Services in container X discovering services in Y

What is Kubernetes?

- Greek for "captain", or "navigator"
- Created by Google, 2014
 - Heavily influenced by Google's Borg system
- Written in Go
- 2015, Google partnered with the Linux Foundation to form the Cloud Native Computing Foundation (CNCF)
 - CNCF current maintainers
- A cluster, consisting of at least one master and multiple worker machines ("nodes").

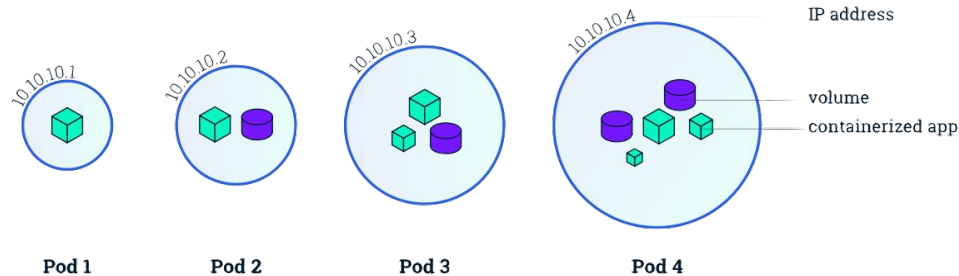
- <https://en.wikipedia.org/wiki/Kubernetes>

Architecture



Peas in a Pod

- A unit of deployment
 - If single containers are deployed, then you can generally replace the word "pod" with "container" and accurately understand the concept.
- A group of one or more containers, with shared storage/network, and a specification for how to run the containers.
- A pod's contents are always co-located and co-scheduled, and run in a shared context.
- Receives a unique IP to prevent port conflicts
 - Intra-Pod communication should use service discovery like Kube-DNS



Operators

"An Operator is a method of packaging, deploying and managing a Kubernetes application"

- Analogous to a *systemd* service, manages an application deployed on Kubernetes.
 - We will install the Percona XtraDB Cluster operator, which will manage N number of PXC clusters.
- Runs in a container

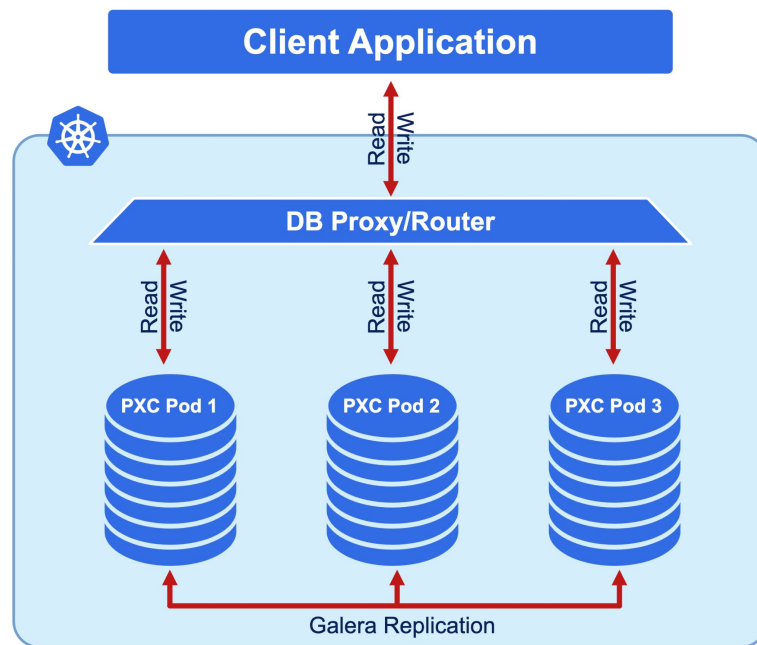


Percona Operator for MySQL

"Oh yea!" -KoolAid Guy

Operator Overview

- An "application" running in K8S
- Maintains a 3-node Percona XtraDB Cluster
- Handles scheduled backups to S3-compliant endpoint
- Choice between L4 or L7 frontend proxy to client applications



Get PXC Operator

```
# git clone -b v1.14.0 https://github.com/percona/percona-xtradb-cluster-operator  
# cd percona-xtradb-cluster-operator
```

Create PXC Namespace

- K8S supports multiple virtual clusters backed by the same physical cluster
 - Called namespaces
- Create and add a new namespace to K8S; set it as the default for future commands.
- A context is a specific K8S cluster
 - Can control multiple K8S clusters from same command

```
$ kubectl create namespace txlifest24  
$ kubectl config set-context $(kubectl config current-context) --namespace=txlifest24
```

Deploy PXC Resources

- Extend the standard set of resources for which Kubernetes "knows" about
- Create several new "resources" within K8S:
 - PerconaXtraDBCluster
 - PerconaXtraDBClusterBackup
 - PerconaXtraDBClusterRestore
- Also establishes API endpoints for use by the operator

```
-- Custom Resource Definition
# kubectl apply -f deploy/crd.yaml

# kubectl get [perconaxtradbclusters|pxc|pxcs|pxc-backups]
```

Create Roles

- K8S uses role-based access control (RBAC)
- Specifically defined roles and actions corresponding to them
- Allows actions to be performed on specific Kubernetes resources
 - Recall that above we created a new resource just for PXC
 - We need to create a role which can access those resources

```
# kubectl apply -f deploy/rbac.yaml
```


Start the Operator

- Time to deploy our PXC Operator within Kubernetes
 - This does not create an XtraDB Cluster; this only starts the operator
- The operator itself is deployed as a docker container, deployed within K8S.
- When we want to deploy actual PXC clusters (later on), we will interact with this operator.

```
# kubectl apply -f deploy/operator.yaml
```

```
# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
percona-xtradb-cluster-operator-6f65855d75-9cnnr	1/1	Running	0	2m27s

Administrative Users Management in K8S

- Add some Percona XtraDB Cluster administrative users and their passwords (secrets) to Kubernetes
- These secrets will be used by the operator when deploying an actual PXC cluster
- They will become regular MySQL users (ie: CREATE USER ...; GRANT ...;)
 - The cluster will be initialized and these user accounts created automatically
 - To change the passwords, edit the data section of `deploy/secrets.yaml`

```
$ kubectl apply -f deploy/secrets.yaml
```

Encryption / SSL / TLS

- Kubernetes doesn't handle on its own
- Cert-Manager: K8S certificate management controller
 - Community-driven, open source
- Configuration beyond scope of tutorial
 - See our official documentation
 - Will use pre-generated SSL/TLS certs

```
$ kubectl apply -f deploy/ssl-secrets.yaml
```

Create a Percona XtraDB Cluster

- Anti-affinity creates rules for deploying PXC Pods
 - Restrict by hostname (default)
 - ... by rack, ... by zone/region
- Switch to using ProxySQL

```
# [vi|nano] deploy/cr.yaml
--
-- Around line 167, disable haproxy
haproxy:
  enabled: false
--
-- Enable ProxySQL, decrease size to 1, enable public exposure
proxysql:
  enabled: true
  size: 1
  ...
  expose:
    enabled: true
    type: LoadBalancer
```

Create a Percona XtraDB Cluster (cont.)

- Deploy! Deploy! Deploy!

```
$ kubectl apply -f deploy/cr.yaml
```

```
$ kubectl get pods [-w]
```

NAME	READY	STATUS	RESTARTS	AGE
cluster1-proxysql-0	3/3	Running	0	20m
cluster1-pxc-0	3/3	Running	0	20m
cluster1-pxc-1	3/3	Running	0	19m
cluster1-pxc-2	3/3	Running	0	19m
percona-xtradb-cluster-operator-889686bfc-9d91g	1/1	Running	0	2h

Connect to MySQL within K8S

- We've created our PXC cluster! Let's check connectivity.
- Launch a docker container inside our K8S cluster with Percona MySQL 8.0 client
 - May take a few minutes to run, with no output, while the container is downloaded.

```
$ kubectl run -it --rm percona-client --image=percona:8.0 -- bash

$ mysql -h cluster1-proxysql -uroot -proot_password
$ mysql -h cluster1-pxc-0.cluster1-pxc.txlinfest24.svc.cluster.local -uroot -proot_password

mysql> SHOW GLOBAL STATUS LIKE 'wsrep_cluster_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 3 |
+-----+-----+

mysql> SHOW GLOBAL STATUS LIKE 'wsrep_cluster_status';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_status | Primary |
+-----+-----+
```

Connect to MySQL Externally

- We exposed ProxySQL with a public IP
 - Yes, bad idea. Use SecurityGroups to limit traffic.

```
$ kubectl get service cluster1-proxysql
NAME                                TYPE                CLUSTER-IP          EXTERNAL-IP
PORT(S)                             AGE
cluster1-proxysql                  LoadBalancer        172.20.102.246
abfa73b8d36464f5ba309742b0cf1bc9-1612698168.us-east-2.elb.amazonaws.com 3306:32339/TCP,33062:30608/TCP
164m

$ mysql -h abfa73b8d36464f5ba309742b0cf1bc9-1612698168.us-east-2.elb.amazonaws.com \
-u root -proot_password
```



Logical Migration

"Dump and load!"

Logical Backup

- Logical backups..
 - .. store data in a generic, storage-independent format
 - .. can be SQL `INSERT` statements, CSV, or tab-delimited
 - .. are the slowest to backup, and slowest to restore
 - .. gain some flexibility in single-row, or single-table restore

Mydumper

- <https://github.com/mydumper/mydumper>
- Logical dump/export tool for MySQL
- Multi-threaded, up to 10x faster than `mysqldump`
- Compression, and encryption on-the-fly
- Dump process saves each table into own file
- Works on local, or remote MySQL (including RDS)

Mydumper in Action

```
$ wget https://github.com/mydumper/mydumper/releases/download/v0.16.1-3/mydumper_0.16.1-3.jammy_amd64.deb
$ sudo apt install ./mydumper_0.16.1-3.jammy_amd64.deb

$ mydumper --verbose 3 --threads 4 --defaults-file /etc/mysql/my.cnf \
  --regex '^(?!mysql\.)' -h 127.0.0.1 --hex-blob --outputdir lifestdump

** Message: 17:53:41.034: MyDumper backup version: 0.16.1-3
** Message: 17:53:41.035: Using 4 dumper threads
** Message: 17:53:41.058: Connected to Percona 8.0.36
** Message: 17:53:41.059: Started dump at: 2024-04-11 17:53:41
...
** Message: 17:53:41.749: Thread 4: dumping data for `imdb`.`user_friends` into
lifestdump/imdb.user_friends.00000.sql| Remaining jobs in this table: 1 All remaining jobs: 43
...
** Message: 17:55:36.599: Main connection closed
** Message: 17:55:36.602: Finished dump at: 2024-04-11 17:55:36

$ ls lifestdump/
imdb-schema-create.sql          sakila.actor.00000.sql
imdb.aka_name-schema.sql       sakila.actor_info-schema-view.sql
imdb.aka_name.00000.sql        sakila.actor_info-schema.sql
imdb.aka_title-schema.sql      sakila.address-schema.sql
...
```

Myloader

- Companion application
- Parallel, multi-threaded, compression, encryption, consistent, etc
- Local, or remote MySQL (including RDS)

```
$ myloader --threads 2 --verbose 3 -h blahblah-1472204115.us-east-2.elb.amazonaws.com \  
  -o -d lifestdump/ --max-threads-for-schema-creation 1 --skip-definer -u root -p root_password  
  
** Message: 00:32:37.847: Using 2 loader threads  
** Message: 00:32:37.913: Connection via default library settings using password:  
  Host: a76a43282b5d346d19f2b125b2a6832b-1472204115.us-east-2.elb.amazonaws.com  
  User: root  
** Message: 00:32:38.015: Initializing initialize_worker_schema  
** Message: 00:32:38.129: S-Thread 3: Starting import  
** Message: 00:32:38.146: Reading metadata: metadata  
** Message: 00:32:38.149: Intermediate queue: Sending END job  
** Message: 00:32:38.150: Thread 3: restoring create database on `world` from world-schema-create.sql.  
Tables 0 of 154 completed  
...
```

Good for Large Datasets

- Mydumper/myloader do work with multi-TB datasets
 - Yes, it will take a while
 - Ex: 2.4TB ~ 14hrs to dump; 21hrs to import
- Sometimes, this is only option
 - Hosted MySQL (eg: RDS) do not support physical backups



Physical Migration

Boo-ya!

Percona Xtrabackup

- Non-blocking, hot backup solution for InnoDB, XtraDB, MyRocks, and MyISAM
 - Supports MySQL, Percona Server, MariaDB (up to MariaDB 10.2)
- Open-source, free (GPL)

- Compressed backups
- Partial backups
- PITR support
- Incremental backups
- Streaming backups
- Parallel backups

- Encrypted backups
- Export individual tables
- LOCK TABLES FOR BACKUP (Percona MySQL feature)
- Native S3 destination support

Prepare the Source

- Create several utility users needed by the Operator
- Update passwords to match those in K8S secrets (or redeploy secrets)

```
mysql> CREATE USER 'monitor'@'%' IDENTIFIED BY 'monitory' WITH MAX_USER_CONNECTIONS 100;
GRANT SELECT, PROCESS, SUPER, REPLICATION CLIENT, RELOAD ON *.* TO 'monitor'@'%';
GRANT SERVICE_CONNECTION_ADMIN ON *.* TO 'monitor'@'%';

CREATE USER 'operator'@'%' IDENTIFIED BY 'operatoradmin';
GRANT ALL ON *.* TO 'operator'@'%';

CREATE USER 'xtrabackup'@'%' IDENTIFIED BY 'backup_password';
GRANT ALL ON *.* TO 'xtrabackup'@'%';

CREATE USER 'replication'@'%' IDENTIFIED BY 'repl_password';
GRANT REPLICATION SLAVE ON *.* TO 'replication'@'%';

SET PASSWORD FOR 'root'@'localhost' = 'root_password';
```


Take and Store the Backup

- Create a physical backup, stored in S3

```
$ xtrabackup --backup --stream=xbstream --target-dir=/tmp/backups/ --extra-lsdir=/tmp/backups/
--password=root_password | \
xbcloud put \
--storage=s3 \
--s3-endpoint='s3.amazonaws.com' \
--s3-access-key='XXCFFF' \
--s3-secret-key='XXCCGG' \
--s3-bucket='percona-training' \
--parallel=10 "lifest24-1"
...
2024-04-12T02:00:48.762309-00:00 2 [Note] [MY-011825] [Xtrabackup] Streaming ./ibdata1
240412 02:00:48 xbcloud: Successfully connected.
2024-04-12T02:00:49.482340-00:00 1 [Note] [MY-011825] [Xtrabackup] >> log scanned up to (5950799426)
2024-04-12T02:00:49.753656-00:00 2 [Note] [MY-011825] [Xtrabackup] Done: Streaming ./ibdata1
...
240412 02:00:50 xbcloud: [0] successfully uploaded chunk: lifest24-1/ibdata1.0000000000000008, size: 21
2024-04-12T02:01:26.604330-00:00 0 [Note] [MY-011825] [Xtrabackup] Transaction log of lsn (5950799426) to
(5950799436) was copied.
2024-04-12T02:01:26.822375-00:00 0 [Note] [MY-011825] [Xtrabackup] completed OK!
240412 02:01:27 xbcloud: Upload completed.
```

Setup Restore Operator Task

- Operator supports restore from S3

```
-- Store S3 credentials into K8S secrets
$ kubectl apply -f deploy/backup/backup-secret-s3.yaml
secret/cluster1-backup-s3 created

-- Create restore job
$ cat deploy/backup/restore.yaml
apiVersion: pxc.percona.com/v1
kind: PerconaXtraDBClusterRestore
metadata:
  name: restore-linfest24-1
spec:
  pxcCluster: cluster1
  backupSource:
    destination: s3://percona-training/linfest24-1
    s3:
      bucket: percona-training
      credentialsSecret: cluster1-backup-s3
      region: us-west-2
      endpointUrl: https://s3.us-west-2.amazonaws.com
```

High-Level Restore

- Stop existing cluster; delete pods
- Create restore pod
 - Attach PV/PVC
 - Download backup, prepare
- Create new PXC pod 0, using PV from restore
- Create new PXC pod 1, clone PV
- Create new PXC pod 2, clone PV
- Update proxy config

Launch Restore

```
$ kubectl apply -f deploy/backup/restore.yaml
perconaxtradbclusterrestore.pxc.percona.com/restore-linfest24-9 created

$ kubectl describe job restore-job-restore-linfest24-9-cluster1
Name:          restore-job-restore-linfest24-9-cluster1
Namespace:    txlinfest24
...
Events:
  Type            Reason             Age   From              Message
  ----            -
  Normal          SuccessfulCreate   10m   job-controller    Created pod:
restore-job-restore-linfest24-9-cluster1-8w1bk

-- Watch restore process
$ kubectl logs restore-job-restore-linfest24-9-cluster1-8w1bk -f
...
+ xcloud get --parallel=4 --curl-retriable-errors=7 s3://percona-training/linfest24-1
240412 04:39:00 xcloud: Successfully connected.
240412 04:39:00 xcloud: [0] Downloading linfest24-1/xtrabackup_tablespace.00000000000000000000.
240412 04:39:00 xcloud: [0] Downloading linfest24-1/world/countrylanguage.ibd.00000000000000000000.
...
240412 04:40:01 xcloud: Download completed.
+ xtrabackup --use-memory=100MB --prepare
...
```

Restore Continued

```
...
2024-04-12T04:40:03.149397-00:00 1 [Note] [MY-011825] [Xtrabackup] Moving ./mysql.ibd to
/datadir/mysql.ibd
2024-04-12T04:40:03.149413-00:00 1 [Note] [MY-011825] [Xtrabackup] Done: Moving file ./mysql.ibd to
/datadir/mysql.ibd
2024-04-12T04:40:03.240849-00:00 0 [Note] [MY-011825] [Xtrabackup] completed OK!
```

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
cluster1-proxysql-0	3/3	Running	0	52s
cluster1-pxc-0	3/3	Running	0	52s
cluster1-pxc-1	0/3	Init:0/1	0	12s
percona-xtradb-cluster-operator-5c496dd5c4-clv27	1/1	Running	0	3h37m
restore-job-restore-linfest24-9-cluster1-8wlbk	0/1	Completed	0	2m7s

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
cluster1-proxysql-0	3/3	Running	0	14m
cluster1-pxc-0	3/3	Running	0	14m
cluster1-pxc-1	3/3	Running	0	13m
cluster1-pxc-2	3/3	Running	0	12m



Conclusion

"That's all folks!"

Last Nutshell

- Percona Kubernetes Operator
 - Percona XtraDB Cluster
 - Percona MySQL
 - Percona MongoDB
 - Percona PostgreSQL
- Fully supported in GKE, EKS, AKS, RedHat OpenShift Certified, Generic K8S
- Built-in proxy support for read/write splitting
- Backup scheduler to PVC/S3
- Point-in-time backup / recovery



Questions?

PERCONA

Databases run better with Percona



<https://forums.percona.com/>

<https://percona.com/training>

Thank You!