
Deploying PXC in Kubernetes / Openshift

Alexander Rubin, Percona

About me

- Working with MySQL for 10-15 years
 - Started at MySQL AB, Sun Microsystems, Oracle (MySQL Consulting)
 - Joined Percona in 2013

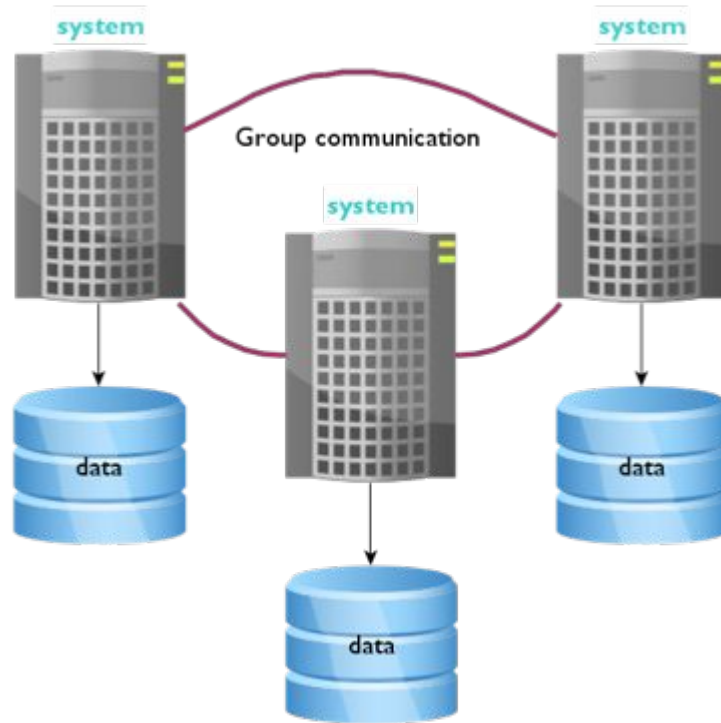
What is Kubernetes?

Kubernetes is an open source system for managing containerized applications across multiple hosts

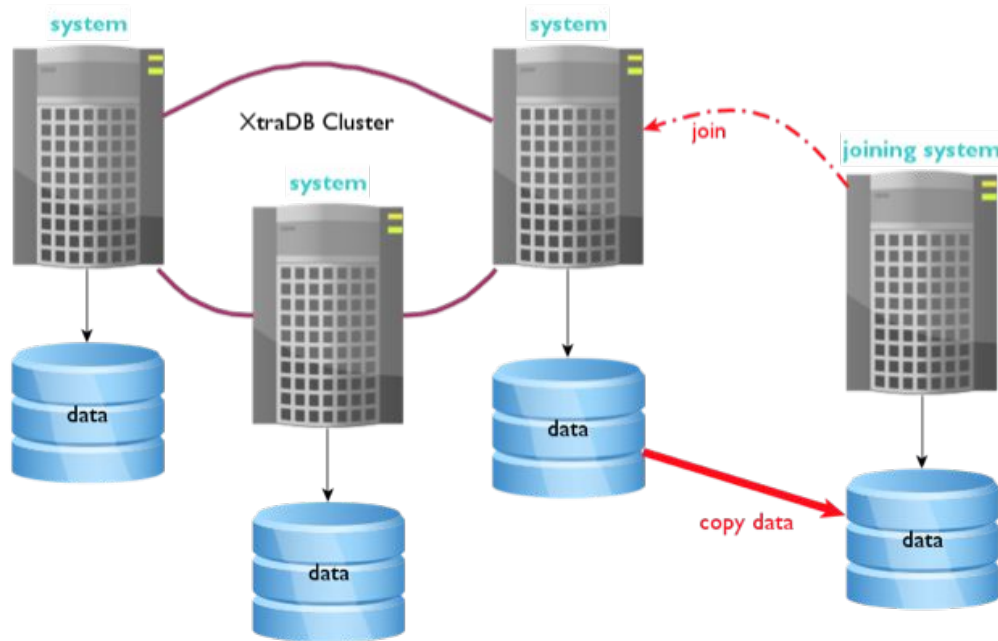


kubernetes

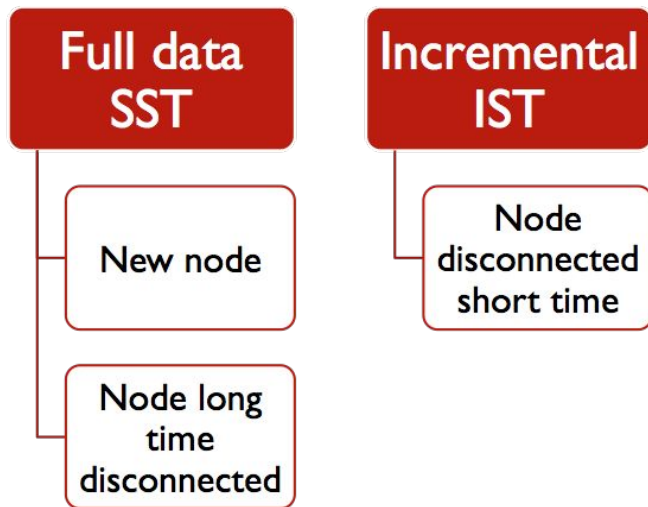
What is Percona XtraDB Cluster (PXC)?



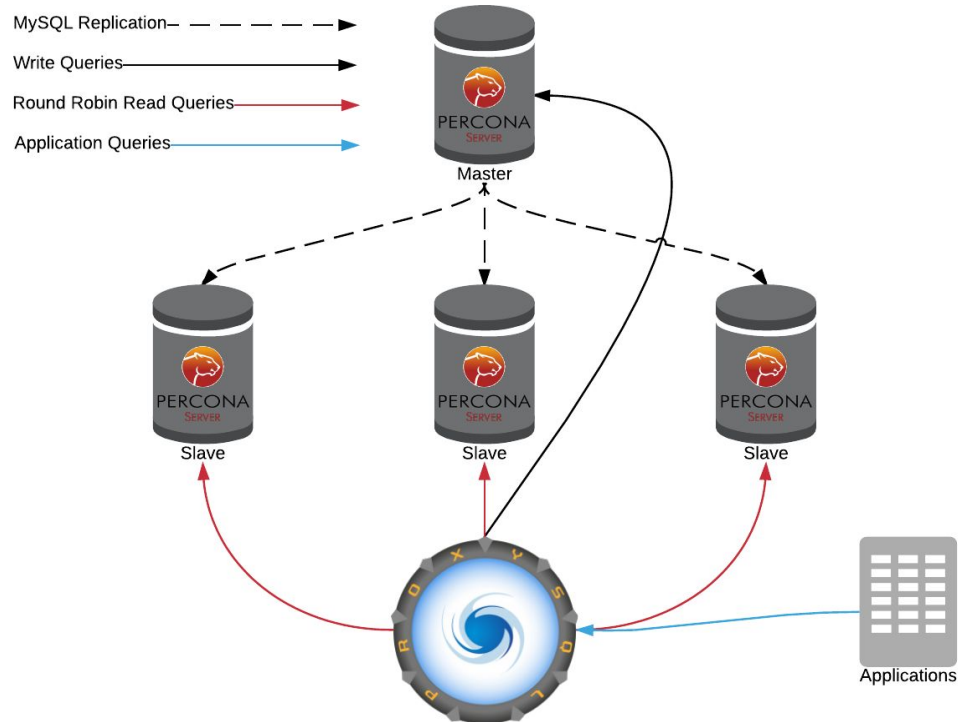
Why PXC? Auto provisioning...



PXC: Data transfer



What is ProxySQL?



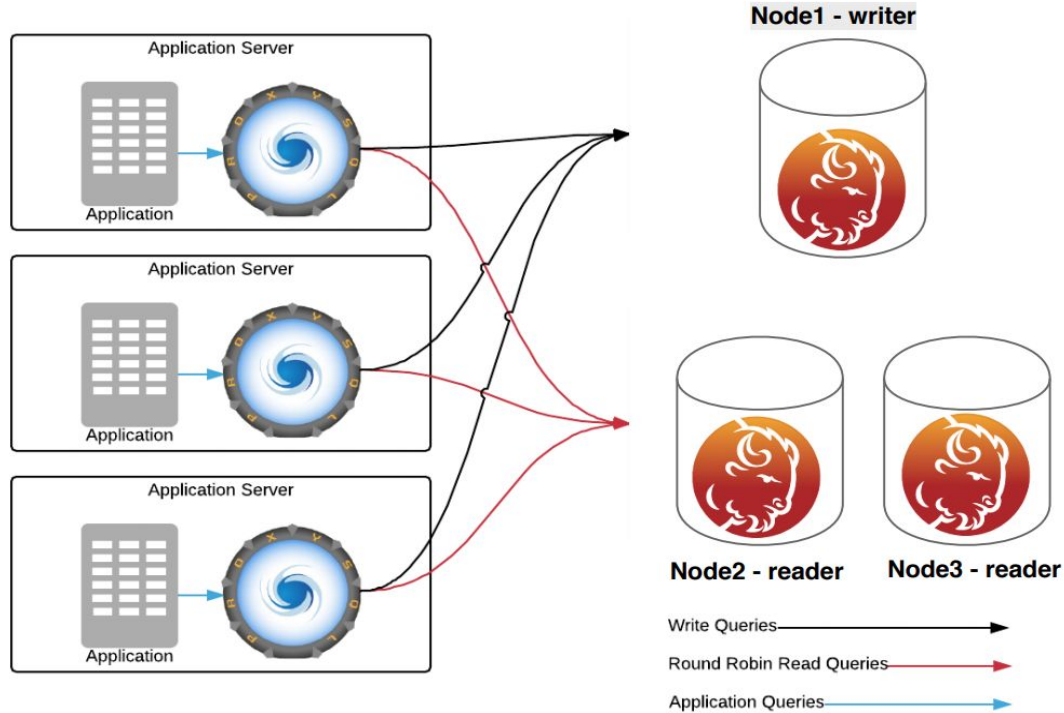
ProxySQL read/write split



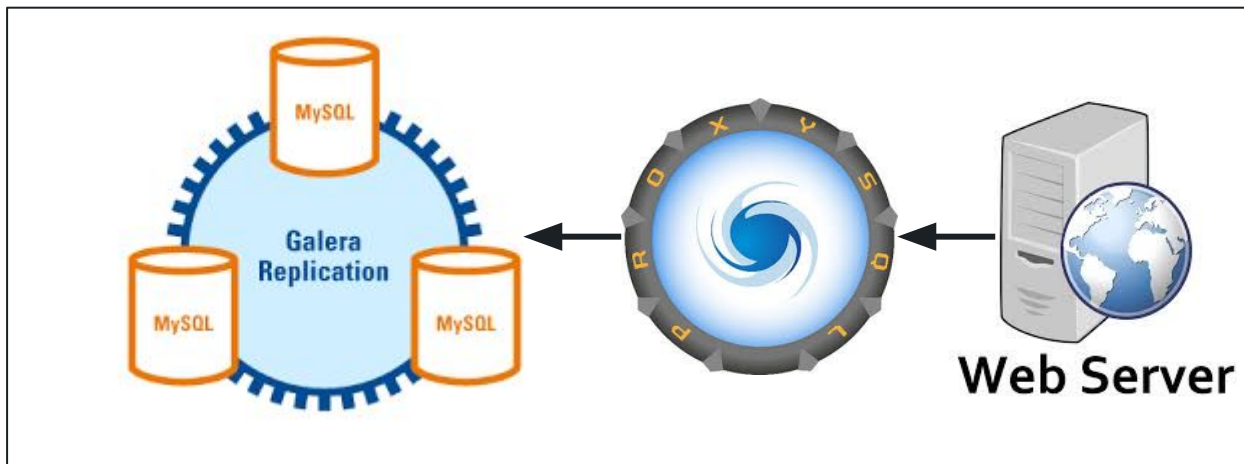
Query routing is one of the core features of ProxySQL.

Read/write split is perhaps one of the most common query routing use

ProxySQL read-write split



PXC + ProxySQL on Kubernetes / Openshift

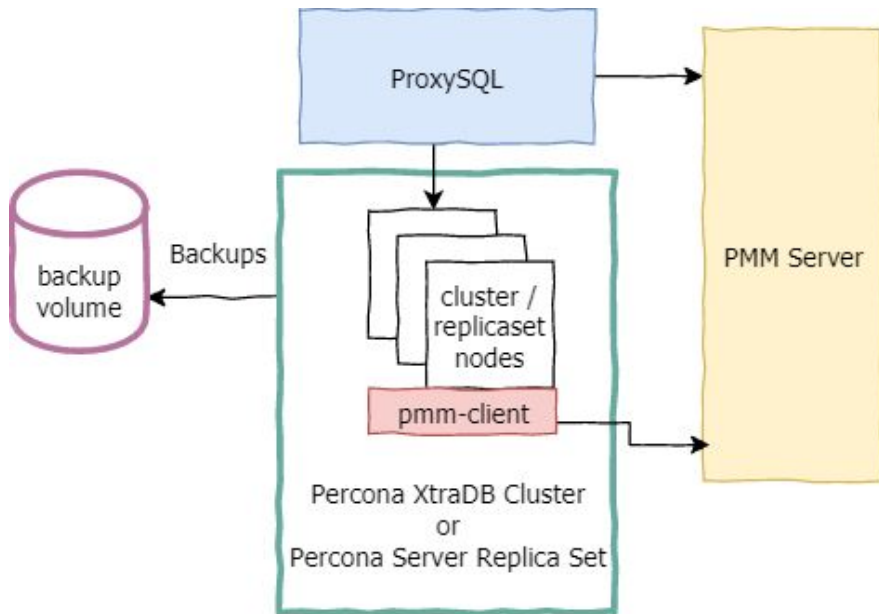


kubernetes

OpenShift Percona Labs

<https://github.com/Percona-Lab/percona-openshift>

`$ git clone git@github.com:Percona-Lab/percona-openshift.git`



Demo / Howto

```
$ ./deploy_all.sh
```

- secret "mysql-passwords" created
- configmap "mysql" created
- service "pxccluster1" created
- statefulset "pxcnodes" created
- statefulset "proxysql" created
- service "sql" created

- secret "mysql-passwords" created
-

```
$ cat secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysql-passwords
type: Opaque
data:
  root: cm9vdF9wYXNzd29yZA==
  xtrabackup: YmFja3VwX3Bhc3N3b3Jk
```

- secret "mysql-passwords" created
-

Use base64 to encode a password for secret.yaml :

- `echo -n 'securepassword' | base64`

Used base64 -d to decode a password from secret.yaml :

- `echo YmFja3VwX3Bhc3N3b3Jk | base64 -d`

- configmap "mysql" created
a way to pass custom config to MySQL (each PXC node)
-

```
$ cat mysql-configmap.yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: mysql
  labels:
    app: mysql
data:
  extra.cnf: |
    [mysqld]
    innodb-buffer-pool-size=512M
```

Demo / Howto

```
$ oc get all
```

NAME	DESIRED	CURRENT	AGE
statefulsets/proxysql	1	1	2m
statefulsets/pxcnode	3	3	2m

NAME	READY	STATUS	RESTARTS	AGE
po/proxysql-0	1/1	Running	0	2m
po/pxcnode-0	1/1	Running	0	2m
po/pxcnode-1	1/1	Running	0	1m
po/pxcnode-2	1/1	Running	0	1m

...

Demo / Howto

```
$ oc get all
```

```
...
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
svc/pxcccluster1	ClusterIP	None	<none>	3306/TCP	6m
svc/sql	ClusterIP	172.30.96.210	<none>	3306/TCP,6032/TCP	6m

Demo / Howto

```
$ oc get all
```

```
...
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
svc/pxcccluster1	ClusterIP	None	<none>	3306/TCP	6m
svc/sql	ClusterIP	172.30.96.210	<none>	3306/TCP,6032/TCP	6m

Demo / Howto - pods



```
$ oc get pods
```



ProxySQL



PERCONA
XtraDB Cluster

NAME

READY

STATUS

RESTARTS

proxysql-0

1/1

Running

0

pxcnode-0

1/1

Running

0

pxcnode-1

1/1

Running

0

pxcnode-2

1/1

Running

0

Demo / Howto

```
$ oc get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
pxcccluster1	ClusterIP	None	<none>	3306/TCP
sql	ClusterIP	172.30.96.210	<none>	3306/TCP,6032/TCP



ProxySQL

MySQL Port
(for app)

Proxy Admin Port

Demo / Howto

```
$ mysql -h 172.30.96.210 -e 'select @@hostname'
```

```
+-----+
```

```
| @@hostname |
```

```
+-----+
```

```
| pxcnode-1 |
```

```
+-----+
```

```
mysql -h 172.30.96.210 -e 'select @@hostname'
```

```
+-----+
```

```
| @@hostname |
```

```
+-----+
```

```
| pxcnode-2 |
```

```
+-----+
```

Demo / Howto

```
$ mysql -h 172.30.96.210 -e 'select @@hostname for update'
```

```
+-----+
```

```
| @@hostname |
```

```
+-----+
```

```
| pxcnode-0 |
```

```
+-----+
```

```
mysql -h 172.30.96.210 -e 'select @@hostname for update'
```

```
+-----+
```

```
| @@hostname |
```

```
+-----+
```

```
| pxcnode-0 |
```

```
+-----+
```

Demo / Howto - test

```
$ for i in {1..600}; do mysql -h `oc get svc/sql -o=yaml|grep  
clusterIP|cut -d':' -f 2` -s -Nbe 'select concat("reader: ",  
@@hostname); select concat("-- writer: ", @@hostname) for update';  
sleep 1; done
```

```
reader: pxcnode-1
```

```
-- writer: pxcnode-0
```

```
reader: pxcnode-2
```

```
-- writer: pxcnode-0
```

```
reader: pxcnode-1
```

```
-- writer: pxcnode-0
```

```
reader: pxcnode-1
```

```
-- writer: pxcnode-0
```

Demo / Howto - test

```
$ docker ps
```

```
...
```

```
79e2918d072c          perconalab/pxc-openshift mysql_pxcnode-0
```

```
...
```

```
$ docker kill 79e2918d072c
```

Demo / Howto - test

...

```
reader: pxcnode-1
-- writer: pxcnode-0
reader: pxcnode-2
-- writer: pxcnode-0
reader: pxcnode-1
-- writer: pxcnode-0
reader: pxcnode-1
-- writer: pxcnode-0
```

Failover - demo

Backup

- Openshift/Kubernetes is a different mindset
 - No cronjob inside container
 - Provides its own cronjob

```
$ cat xtrabackup-job.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: xtrabackup-job
spec:
```

Monitoring

Percona Monitoring and Management platform integration

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

Next Steps - helm

```
$ helm install --name cluster1 . -f values.yaml
```

By default will deploy proxysql in from of nodes and pmm-client on each node

Install Helm in OpenShift

For the details see <https://blog.openshift.com/getting-started-helm-openshift/>

Conclusion

- Openshift provides easy deployment for MySQL (Percona XtraDB Cluster)
 - Great for deployment and testing
- ProxySQL provides the cluster endpoint
 - Implements read-write split