HA MySQL on Kubernetes using Percona XtraDB Cluster and ProxySQL

Deploy a High Available (HA) MySQL environment on Kubernetes using Percona XtraDB Cluster (PXC) along with ProxySQL for read/write splitting

Ananias Tsalouchidis
Senior MySQL DBA
Percona University, Istanbul
October, 2019

PERCONA

What is this presentation about

- Short introduction to Kubernetes, Percona XtraDB Cluster (PXC) and ProxySQL
- Hands-on deployment of a High Available (HA) MySQL environment on Kubernetes using Percona XtraDB Cluster (PXC) along with ProxySQL for read/write splitting.
- By the end of this session you should be able to get a better understanding of:
  - *What is Kubernetes, XtraDB Cluster and ProxySQL*
  - *How to deploy HA MySQL on Kubernetes using Percona XtraDB Cluster and ProxySQL*

© 2019 Percona

About me

- BSc and a MSc in computer science
- Almost 15 years of working experience as a systems and databases administrator
- Joined Percona as a Senior MySQL DBA on May 2017
- Passionate about databases and programming
- Located in Thessaloniki, Greece, with my wife and our two children

© 2019 Percona

About This Presentation

Kubernetes (K8s)

Percona XtraDB Cluster

ProxySQL

© 2019 Percona

Kubernetes

What is Kubernetes (K8s)

Open-source system for automating deployment, scaling and management of containerized applications.

The project was started by Google as an open-source next generation container scheduler. It grew out of a previous Google project, called Borg

© 2019 Percona

PERCONA

What Kubernetes does?

- Linux kernel for distributed systems
- Abstracts away the underlying hardware of the nodes and provides a uniform interface for applications to be deployed and consume a shared pool of resources
- Manages the lifecycle of containerized applications and services. It provides a predictable and consistent way to manage scalability and recovery from a failure in high-availability environments
- The goal of Kubernetes is to provide a platform for automating deployment, scaling, and operations of application containers across clusters of hosts

© 2019 Percona
PERCONA

Kubernetes implementations

- Openshift by RedHat
- Amazon Elastic Kubernetes Service (Amazon EKS)
- Google Kubernetes Engine (GKE)
- Azure Kubernetes Engine (Azure EKS engine)
- Pivotal Kubernetes Service (PKS)
- Rancher
- *Minikube (tool that allows to run K8s locally, into a VM)
- and more …

© 2019 Percona

Basic Kubernetes concepts

- **Cluster**: A collection of hosts that aggregate their available resources into a usable pool
- **Master(s)**: A collection of components that make up the control plane of Kubernetes. These components are responsible for all cluster decisions including both scheduling and responding to cluster events
- **Node**: A single host, physical or virtual, capable of running Pods. A node is managed by the master(s) and at a minimum runs both kubelet and kube-proxy to be considered as part of the cluster.
- **Pod**: Group of containers deployed together on the same node. For single containers, "pod" = "container"
- **Namespace**: A logical cluster or environment.

© 2019 Percona

PERCONA

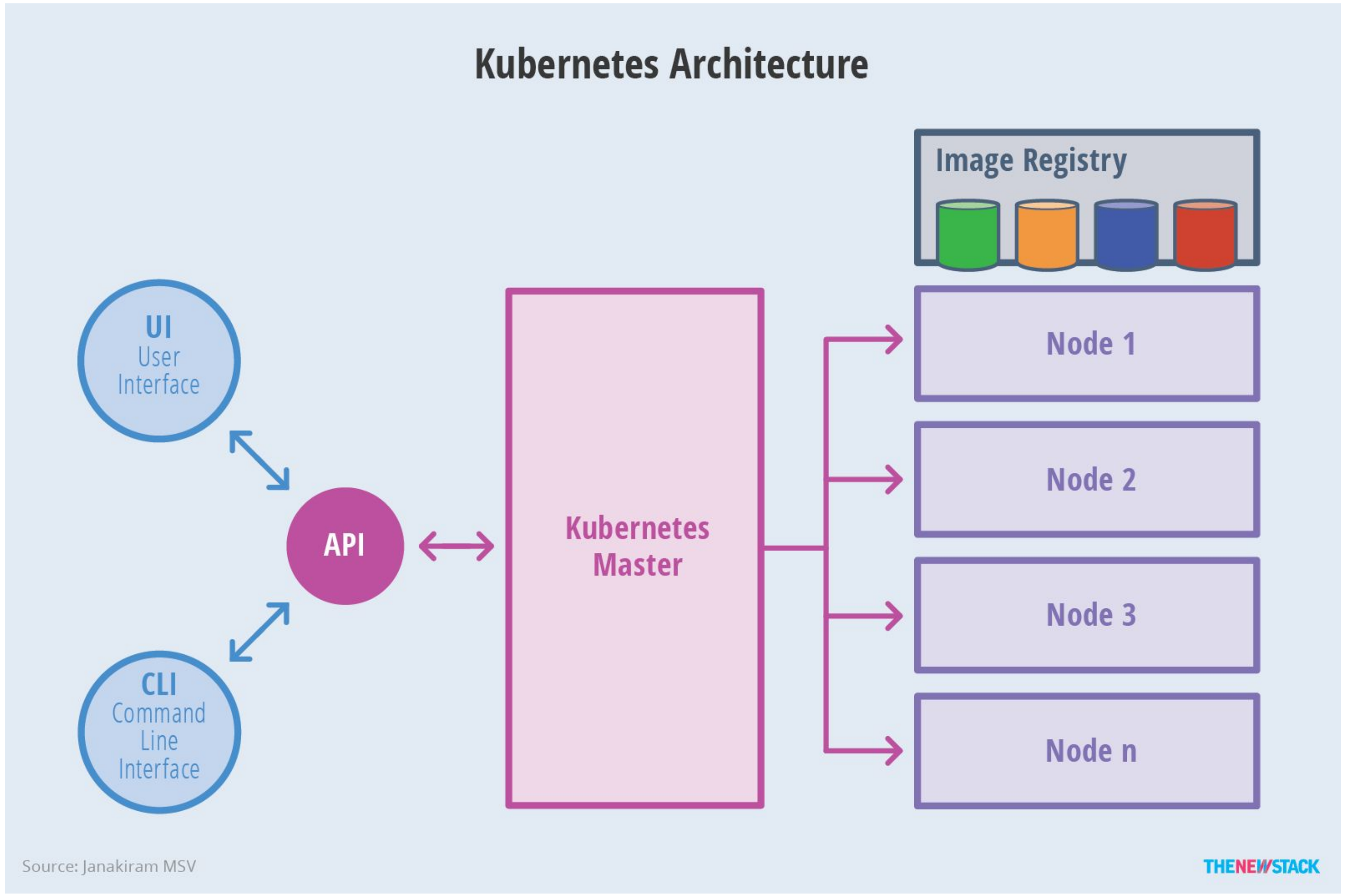
Kubernetes architecture

- **Master**

Acts as the primary control pane for Kubernetes. Masters are responsible at a minimum for running the API Server, scheduler and cluster controller

- **Nodes**

Nodes are the “workers” of the Kubernetes cluster. They run a minimal agent that manages the node itself and are tasked with executing workloads as designated by the master



© 2019 Percona

Master Components

- **Kube-apiserver**: Exposes the Kubernetes API. It is the front-end for the Kubernetes control plane. It is designed to scale horizontally – that is, it scales by deploying more instances
- **Etcd**: Consistent and highly-available key value store used as Kubernetes' backend store for all cluster data
- **Kube-scheduler**: Monitors newly created pods that have no node assigned and selects a node for them to run on
- **Kube-controller-manager**: Runs controllers . Each controller is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process
- **Cloud-control-scheduler**: Runs controllers that interact with the underlying cloud providers

© 2019 Percona

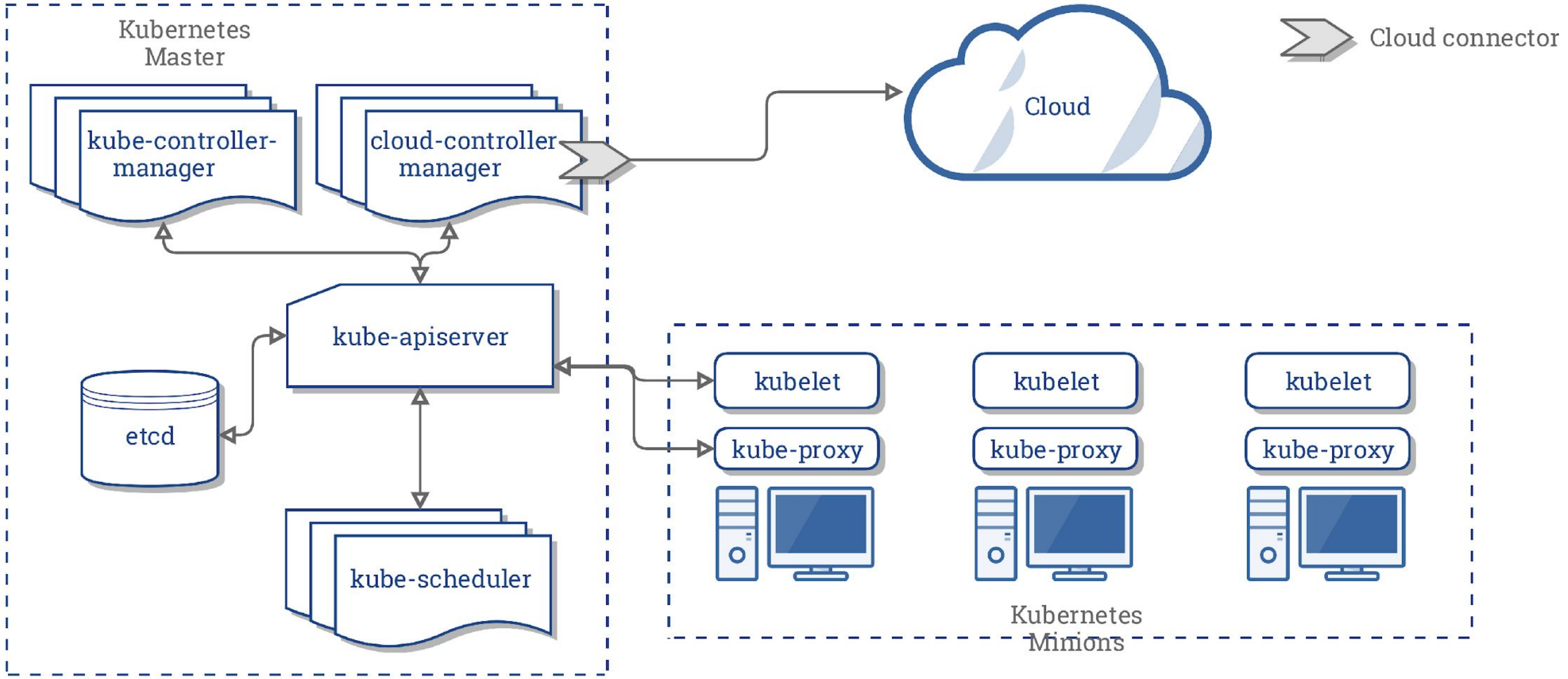
PERCONA

Node Components

- **Kubelet**: Agent that runs on each node in the cluster. It ensures that containers are running in a pod
- **Kube-proxy**: Network proxy that runs on each node in the cluster, implementing part of the Kubernetes Service concept. It maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster
- **Container runtime engine**: Software that is responsible for running containers

© 2019 Percona

Components all together

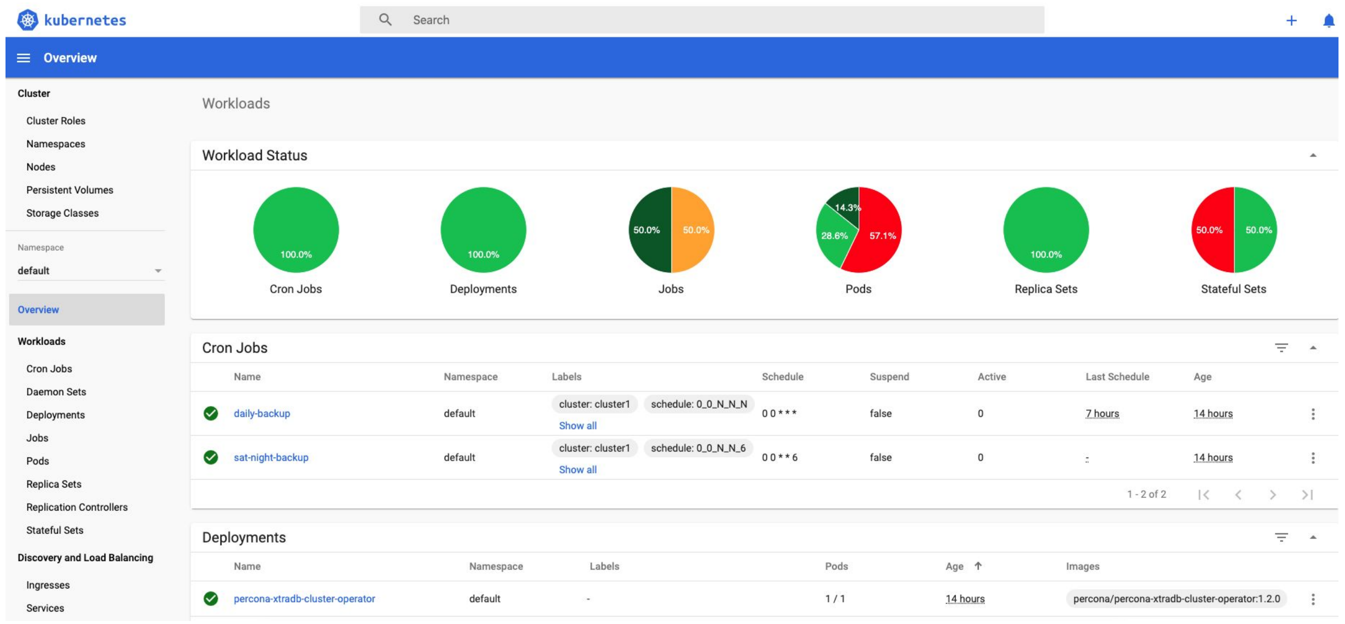


© 2019 Percona

PERCONA

Additional Services

- **Kube-dns**: Provides cluster wide DNS services i.e. <service>.<namespace>.svc.cluster.local
- **Integrated Prometheus endpoints**
- **Kube-dashboard:** General purpose web UI for Kubernetes



Cron Jobs

	Name	Namespace	Labels	Schedule	Suspend	Active	Last Schedule	Age
✓	daily-backup	default	cluster: cluster1 schedule: 0_0_N_N_N Show all	0 0 * * *	false	0	7 hours	14 hours
✓	sat-night-backup	default	cluster: cluster1 schedule: 0_0_N_N_6 Show all	0 0 * * 6	false	0	-	14 hours

1 - 2 of 2

Deployments

	Name	Namespace	Labels	Pods	Age	Images
✓	percona-xtradb-cluster-operator	default	-	1 / 1	14 hours	percona/percona-xtradb-cluster-operator:1.2.0

© 2019 Percona

Replication

Replication options and challenges

- **Traditional MySQL replication**
  - No automatic failover
  - No automatic provisioning (until ver. 5.7 / MySQL ver. 8 offers the clone plugin)
- **Group Replication**
  - Not mature enough
  - No automatic provisioning (until ver. 5.7 / MySQL ver. 8 offers the clone plugin)
- **Percona XtraDB Cluster** (see below)

© 2019 Percona
PERCONA

What is Percona XtraDB Cluster (PXC)

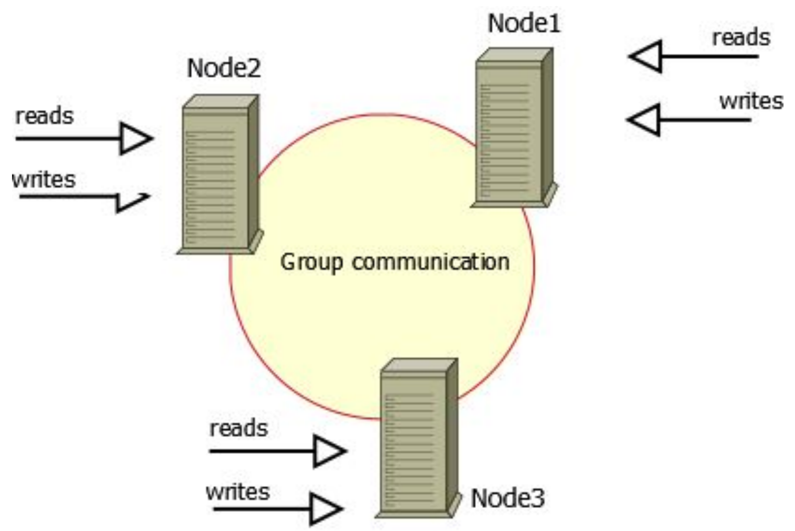
- PXC - multi-master solution. Integrates Percona Server and Percona XtraBackup with the Galera library
- Independent Percona product with a complete life cycle.

- ❑ Consistency
- ❑ Availability
- ❑ Partition Tolerance

Percona

XTRADB CLUSTER

Percona XtraDB Cluster is an active/active high availability and high scalability open source solution for MySQL clustering.



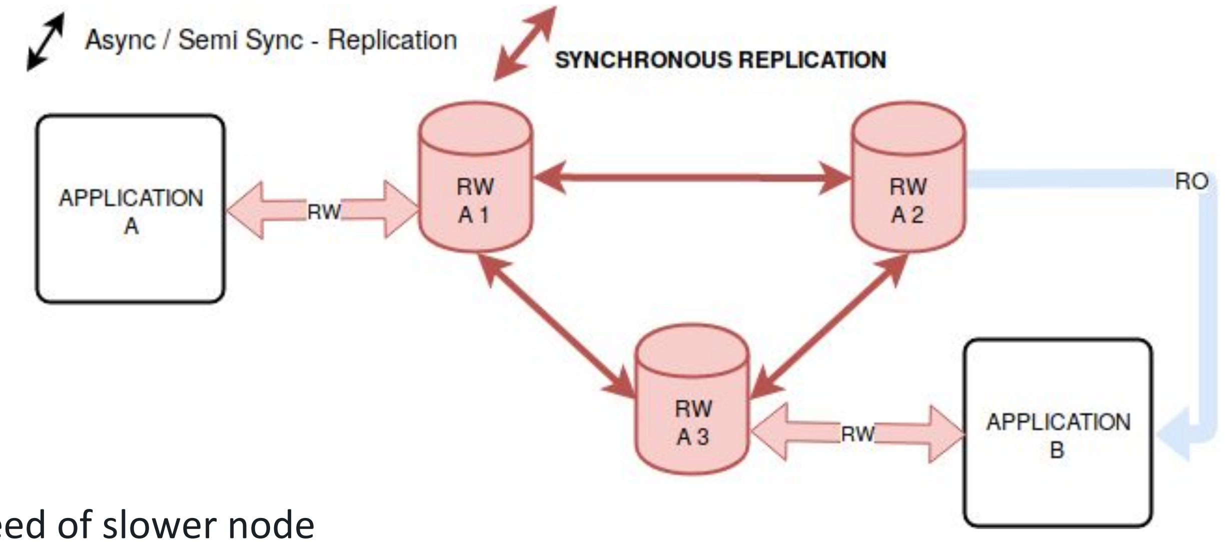
© 2019 Percona

PERCONA

What is Percona XtraDB Cluster (PXC)

- Write to any node(1)
- Read from any node

CURRENT(2) AND CONSISTENT

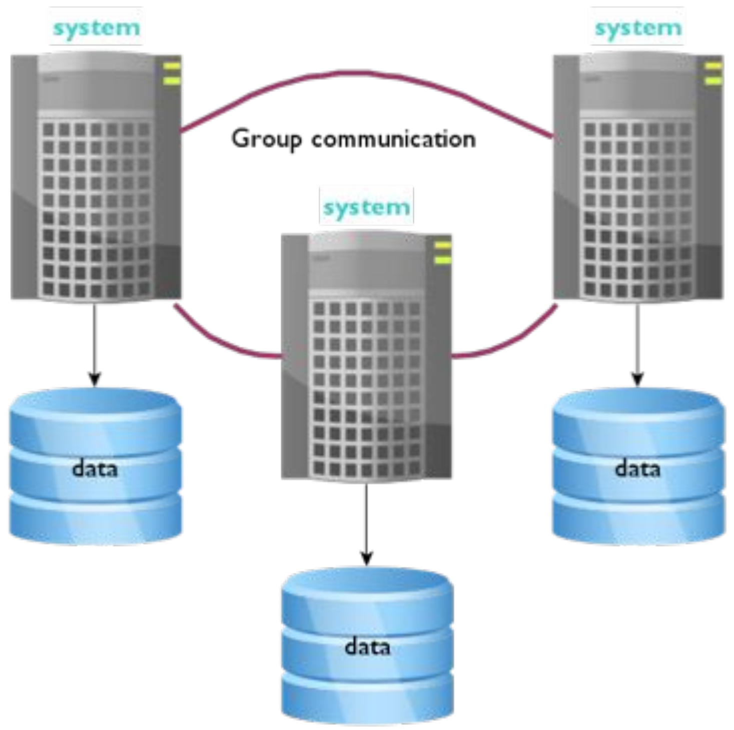


1 Subject to deadlocks, first committer wins, speed of slower node
2 Subject to <fc_limit> number of write sets.

© 2019 Percona

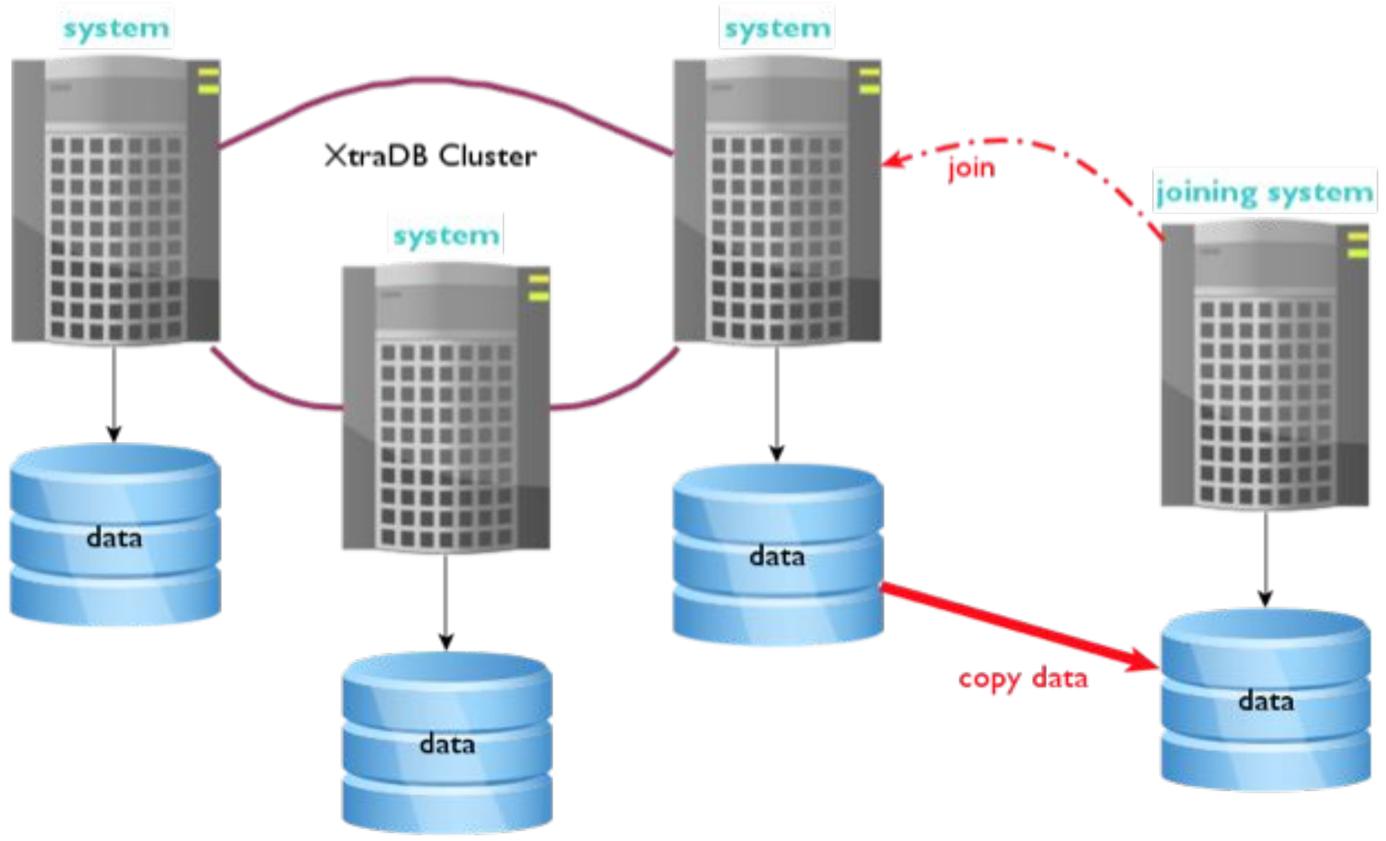
PERCONA

What is Percona XtraDB Cluster (PXC)



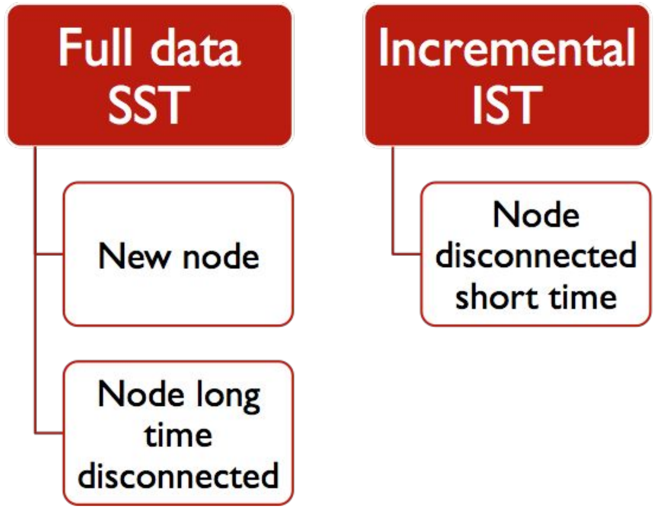
© 2019 Percona

What is Percona XtraDB Cluster (PXC)



© 2019 Percona

What is Percona XtraDB Cluster (PXC)



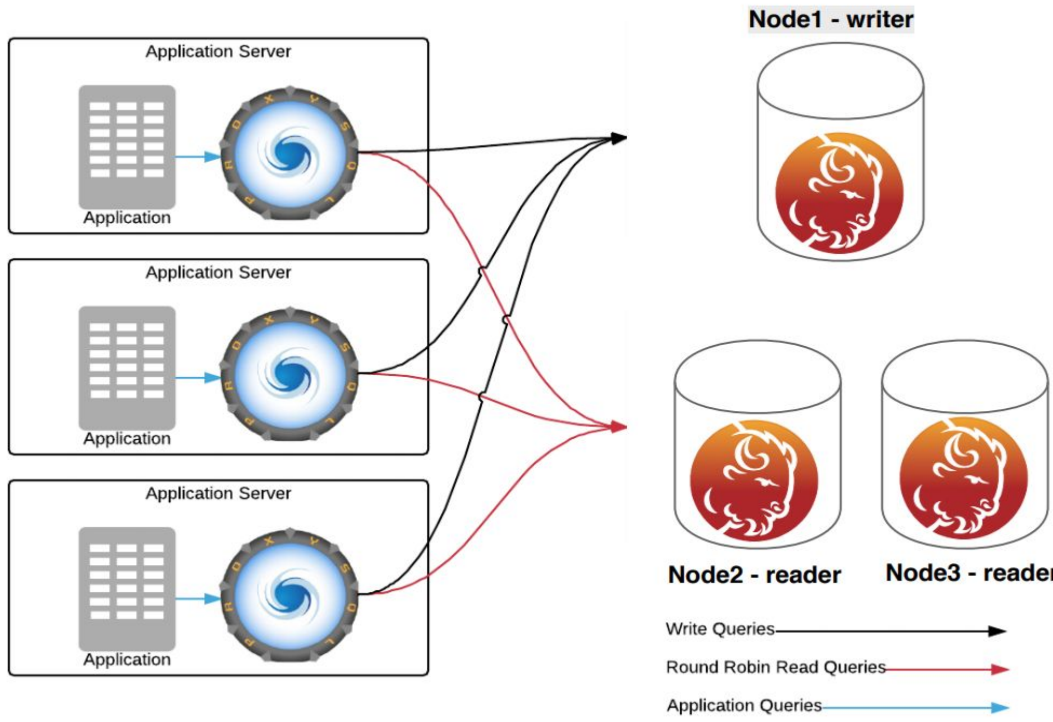
*time refers actually to the amount of data that can been written to the cluster. The bigger the *gcache* is the longer a node can stay offline without an SST

© 2019 Percona

Traffic routing

What is ProxySQL

- High-performance MySQL proxy. Not just a TCP forwarder. It "understands" SQL
- Supports Failover
- Query routing (one of its core features)
  - Read/write split
- Can be used as a firewall
- Supports query caching
- Open source



© 2019 Percona

Kubernetes operator

Kubernetes operators

- A Kubernetes Operator functions as the brain of Kubernetes. It is used to add or remove containers from the environment and it follows a standard and repeatable process to manage those changes
- Controller which provides automation with the ability to create objects, such as a pods, to listen for a specific event and then perform a task
- Extend the capabilities of Kubernetes, enabling it to manage a variety of applications. As applications become increasingly complex, automating issues like scalability and failure recovery become more complex
- One of the main advantages of a Kubernetes Operator is that it performs these tasks in a consistent and reliable manner by following a standard set of ‘rules’

© 2019 Percona
PERCONA

Percona Kubernetes Operator for XtraDB Cluster

- Percona provides Percona Kubernetes Operators for XtraDB Cluster. This Operator makes it easy to implement a high available, self-healing, and scalable MySQL environment using PXC and ProxySQL
- The Percona Kubernetes Operator for XtraDB Cluster provides a consistent and repeatable way to deploy an HA open source MySQL environment. The operator can recover a PXC instance (Pod) not a physical node.
- The Operator supports monitoring (PMM) and backups as well

https://www.percona.com/doc/kubernetes-operator-for-pxc/index.html

© 2019 Percona

PERCONA

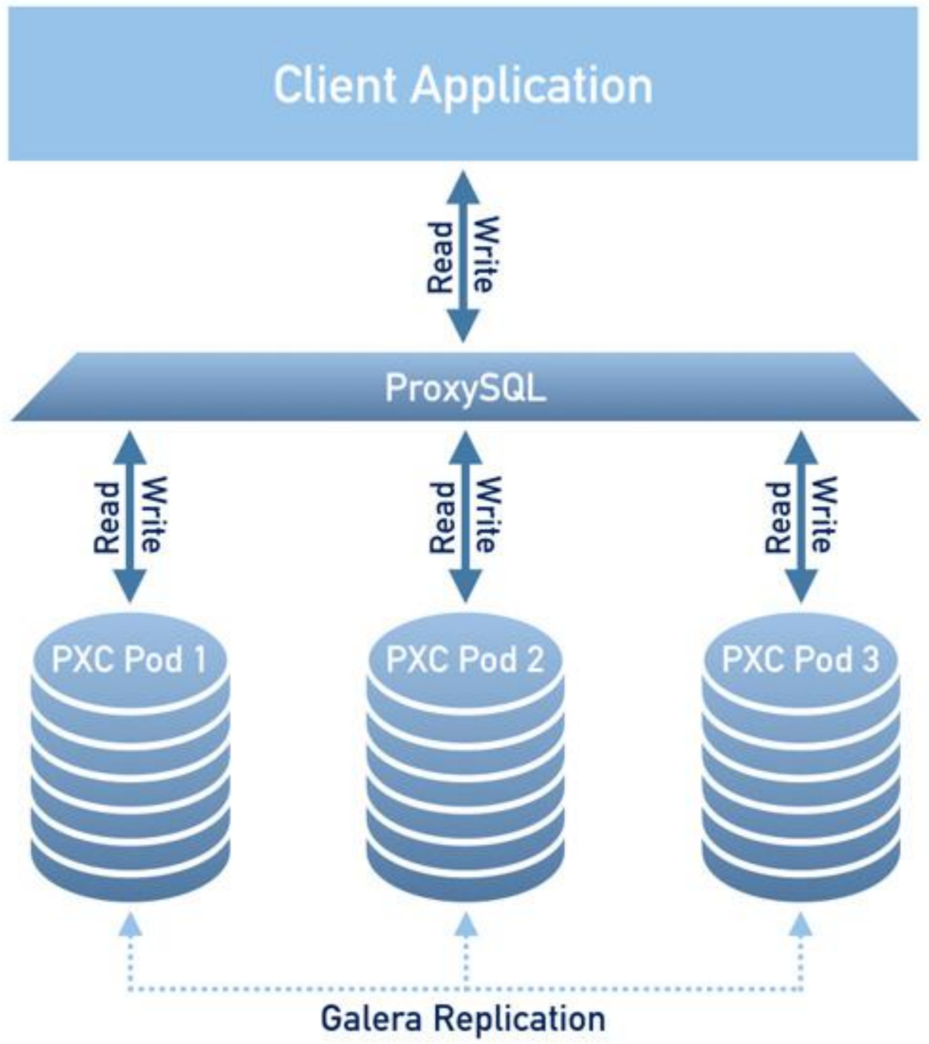
Kubernetes + PXC + ProxySQL all together

Our operator supports advanced affinity rules.

To provide high availability, the operator uses node affinity to run PXC instances on separate worker nodes if possible. If a node fails, the pod is re-created on another node.

If needed, ProxySQL can be ran on the same physical node with application Pod.

- **Query caching closer to the application**
- **Less network overhead**



© 2019 Percona

Hands-On

Hands on

Install minikube to locally run K8s:

- brew install minikube

Set memory resources:

- minikube config set memory 4096

  * *you should remember that if you allocate to your Pods more resources in total, PODS that do not fit won't run*

Start minikube:

- minikube start

Optionally, start the dashboard:

- minikube dashboard

© 2019 Percona

PERCONA

Hands on

Clone operator:

- git clone -b release-1.2.0 https://github.com/percona/percona-xtradb-cluster-operator
- cd percona-xtradb-cluster-operator

Deploy operator:

- kubectl apply -f deploy/bundle.yaml

Edit params to allow local run (deploy/cr.yaml):

- Reduce memory and CPU resources for ProxySQL and PXC nodes
  - resources.requests.memory
  - resources.requests.cpu
- Reduce the spec.proxysql.size from 3 to 1
- Set affinity.antiAffinityTopologyKey key to "none"
- Switch allowUnsafeConfigurations key to true

© 2019 Percona

Hands on

Apply cr.yaml:

- kubectl apply -f deploy/cr.yaml

Verify PODs have been created:

- kubectl get pods

```
atsaloux@macbook-pro:~/percona-xtradb-cluster-operator$ kubectl get pods
NAME                                    READY   STATUS    RESTARTS   AGE
cluster1-proxysql-0                           3/3     Running   0          34m
cluster1-pxc-0                              1/1     Running   0          34m
cluster1-pxc-1                              1/1     Running   0          31m
cluster1-pxc-2                              1/1     Running   0          31m
percona-xtradb-cluster-operator-74bcbd9df5-fbmsc   1/1     Running   0          34m
```

© 2019 Percona

Hands on

Get and decode the passwords:

- kubectl get secret my-cluster-secrets -o yaml
- echo '<HASH>' | base64 --decode

Create Pod to login to MySQL. This Pod will be terminated when you logout:

- kubectl run -i *--rm* --tty percona-client --image=percona:5.7 *--restart=Never* -- bash -il

Login to ProxySQL:

- mysql -h cluster1-proxysql -P 3306 -p -u root

Run two loops … simple select and select … for update. Ohhh. Really interesting

- Note that the Percona PXC operator installs ProxySQL 2.0 from the Percona repo. ProxySQL 2.0 includes native support for Galera clusters so we don't have to enable the scheduler as we did in version 1.x

© 2019 Percona

Hands on

Delete a PXC Pod

- kubectl delete pod cluster1-pxc-1

K8s will respawn another Pod

Check logs on another PXC node. An SST is in progress for the new node

- kubectl logs cluster1-pxc-0

© 2019 Percona

Hands on

Increase number of PXC Pods - Scale up:

- vim deploy/cr.yaml and save (adjust the size for pxc, pxc.size from 3 to 5)
- apply the new configuration: kubectl apply -f deploy/cr.yaml

Verify Pods have been created:

- kubectl get pods

atsaloux@macbook-pro:~$ **kubectl get pods**

```
atsaloux@macbook-pro:~/percona-xtradb-cluster-operator$ kubectl get pods
NAME                                              READY   STATUS    RESTARTS   AGE
cluster1-proxysql-0                               3/3     Running   0          24m
cluster1-pxc-0                                    1/1     Running   0          24m
cluster1-pxc-1                                    1/1     Running   0          22m
cluster1-pxc-2                                    1/1     Running   0          21m
cluster1-pxc-3                                    1/1     Running   0          64s
cluster1-pxc-4                                    0/1     Running   0          9s
percona-xtradb-cluster-operator-74bcbd9df5-fbmsc  1/1     Running   0          24m
```

© 2019 Percona

Hands on

Create again a Pod to login to MySQL (if the previous one was terminated):

- kubectl run -i *--rm* --tty percona-client --image=percona:5.7 *--restart=Never* -- bash -il

Login to ProxySQL again:

- mysql -h cluster1-proxysql -P 3306 -p -u root

Run two loops … simple select and select … for update.

Did you notice that? The new Pods are already being used!

© 2019 Percona

Hands on

How the operator knows if a node is healthy/ready?

- kubectl describe pod cluster1-pxc-1
- kubectl exec -it cluster1-pxc-0 -- /bin/bash
- => /usr/bin/clustercheck.sh

What happens when PXC or ProxySQL nodes are being added ?

- kubectl describe pod cluster1-proxysql-0
- kubectl exec -it cluster1-proxysql-0 -- /bin/bash
- => /usr/bin/add_proxysql_nodes.sh

© 2019 Percona

Conclusions

Conclusions

- With Kubernetes you can use the same orchestration tool and command-line interfaces to deploy your containerized apps in a consistent and reliable way across multiple, maybe different, systems
- Old hardware may be used as well. You may reduce the overall cost of cloud subscriptions
- Everything is ephemeral. Architecture should allow you to run in case something is broken. When you speak about databases you need reliable architecture which cannot just stop working. Kubernetes provides automatic scaling and failure detection
- Operators can automate few things
- Kubernetes is not only for production. It allows DEVs to quickly bring up a cluster or a stack giving them the option to quickly test things before pushing to production. It’s an option instead of running ansible scripts, local vagrant etc .You can aggregate HW resources and create a resource pool where you can deploy and test your apps.
- Project is mature enough nowadays and many companies are moving dbs to k8s. There is some work still to be done though.

© 2019 Percona
PERCONA

Champions of Unbiased
Open Source Database Solutions