ProxySQL at Shopify

shopify

Who we are

Jordan Wheeler

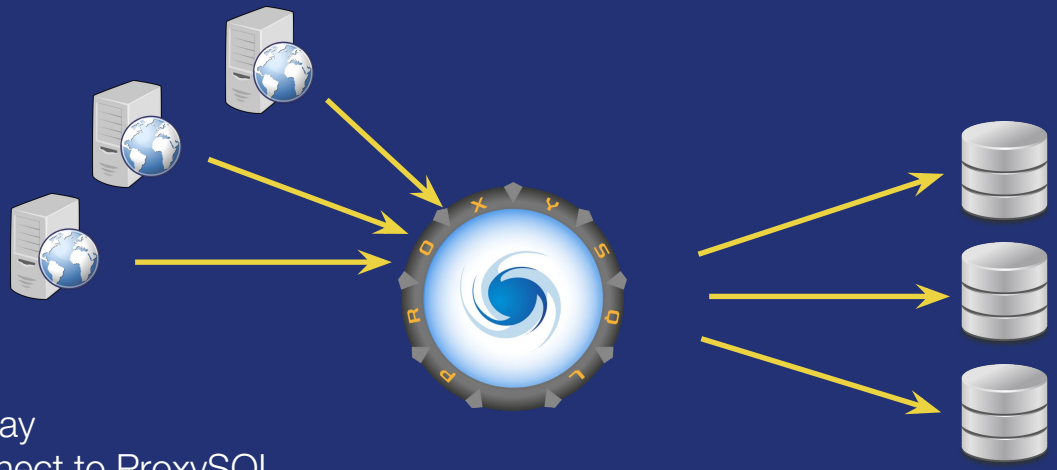
Senior Production Engineer @ Shopify

René Cannaò

Founder @ ProxySQL

shopify

ProxySQL Architecture Overview



Data gateway
Clients connect to ProxySQL
Requests are evaluated
Actions are performed

shopify

Some of the most interesting features:

- on-the-fly rewrite of queries
- caching reads outside the database server
- connection pooling and multiplexing
- complex query routing and read/write split
- load balancing
- real time statistics
- monitoring
- data masking
- multiple instances on same ports

Some of the most interesting features:

- high availability and scalability
- seamless failover
- firewall
- query throttling
- query timeout
- query mirroring
- runtime reconfiguration
- scheduler
- support for Galera/PXC and Group Replication

shopify

Some of the most interesting features:

- support for millions of users
- support for tens of thousands of database servers
- native ProxySQL Clustering Solution
- support for ClickHouse as a backend
- support for Aurora
- SSL support for frontend
- SSLv1.2
- native Support for Galera
- causal reads using GTID

Multiplexing:

Reduce the number of connections against mysqld (configurable)

Many clients connections (tens of thousands) can use few backend connections (few hundreds)

Tracks connection status (transactions, user variables, temporary tables, etc)

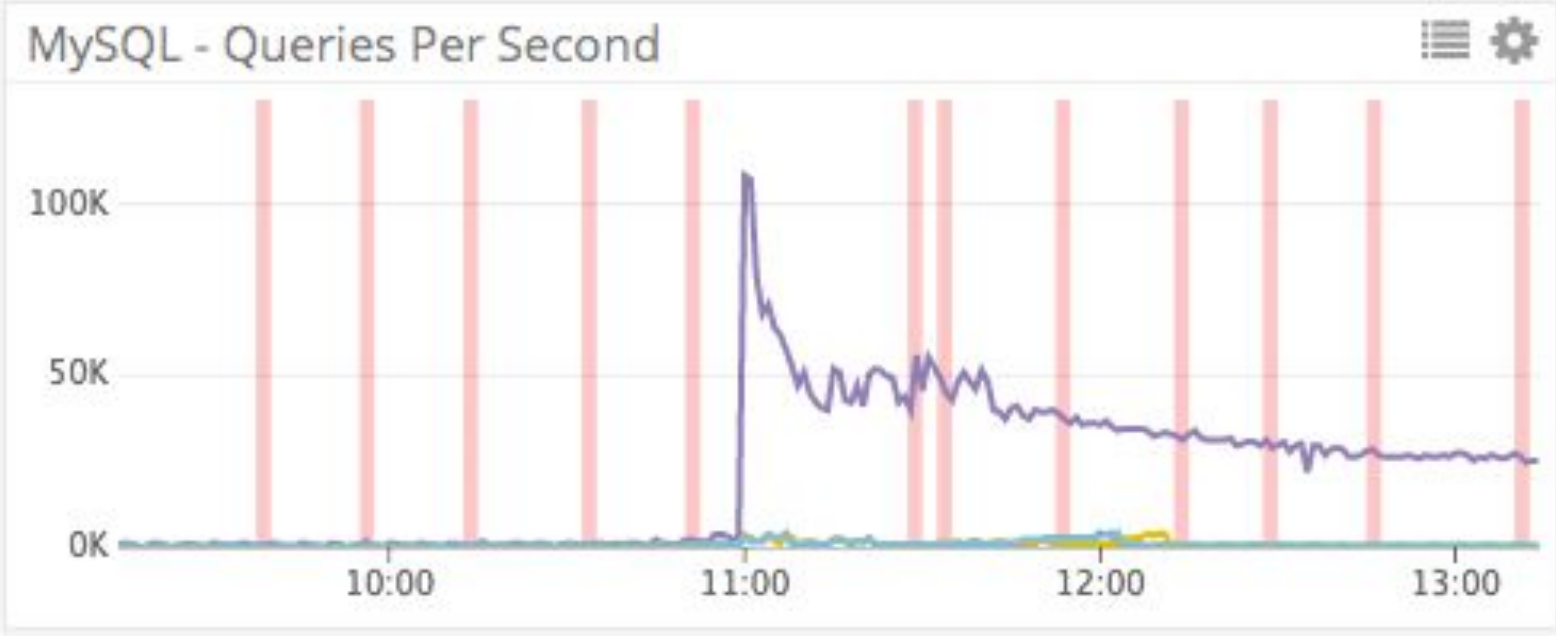
Order by waiting time

600K Active Merchants
80K RPS Peak
$26B GMV 2017
1000 Developers
40 Deploys/Day

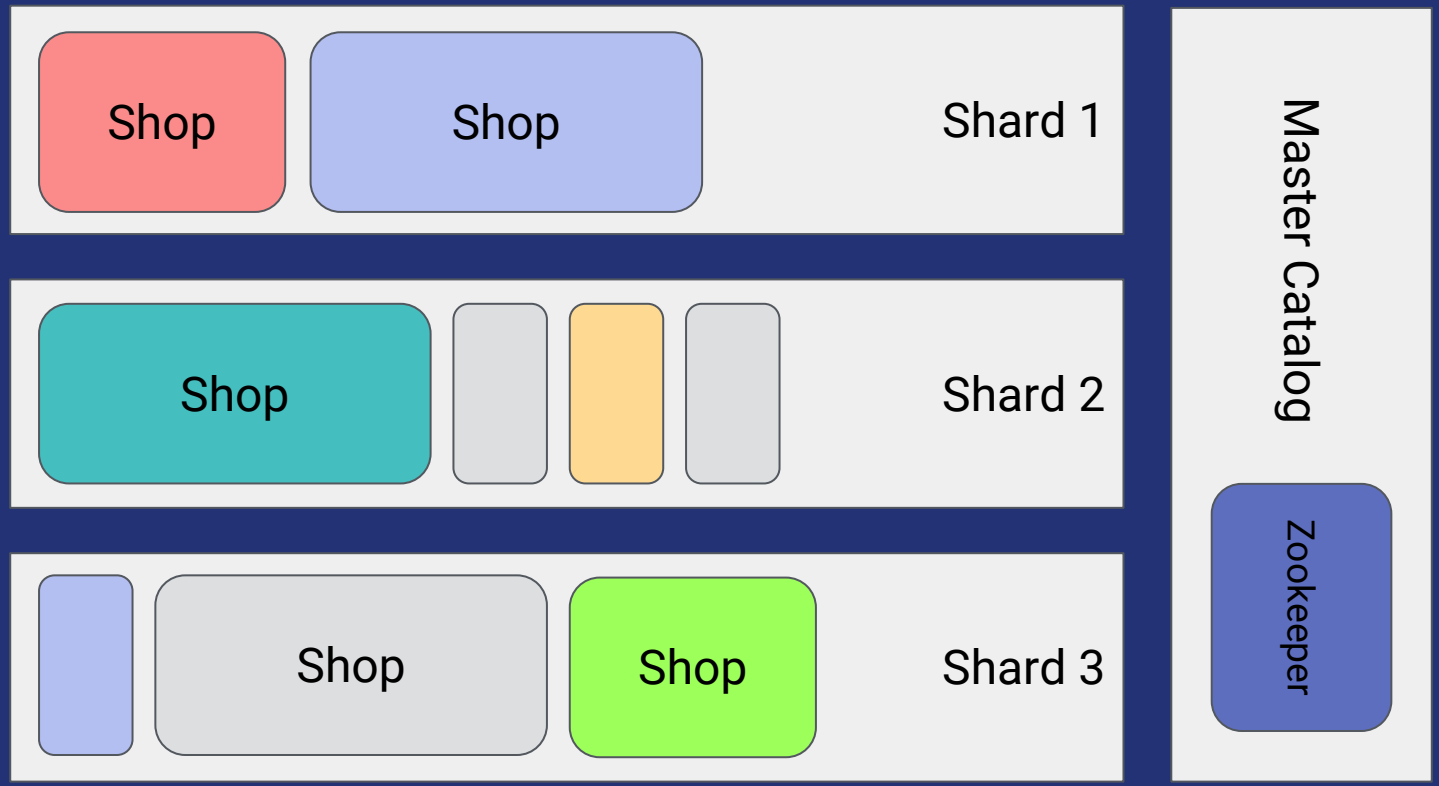
shopify


KYLIE
JENNER
kyliejenner • Follow
kyliejenner SURPRISE! The #WETSET is dropping on KylieCosmetics.com in 1 hour! @kyliecosmetics if you haven't tried it before.. it's about to be your new best friend!
Load more comments
elsyendypatty_ Fb
elsyendypatty_ Cb
syifaazzahraa__ lb
syifaazzahraa__ fb
syifaazzahraa__ cb
yungmanok Lb
htthuy03 Lbllb
saisharma9990 💝 💝 💟 💜 💜 🖤 💜 🖤
__marvielous Lb
bilnabilaahhaaajr_ Lb
990,395 likes
JUNE 22
Log in to like or comment.

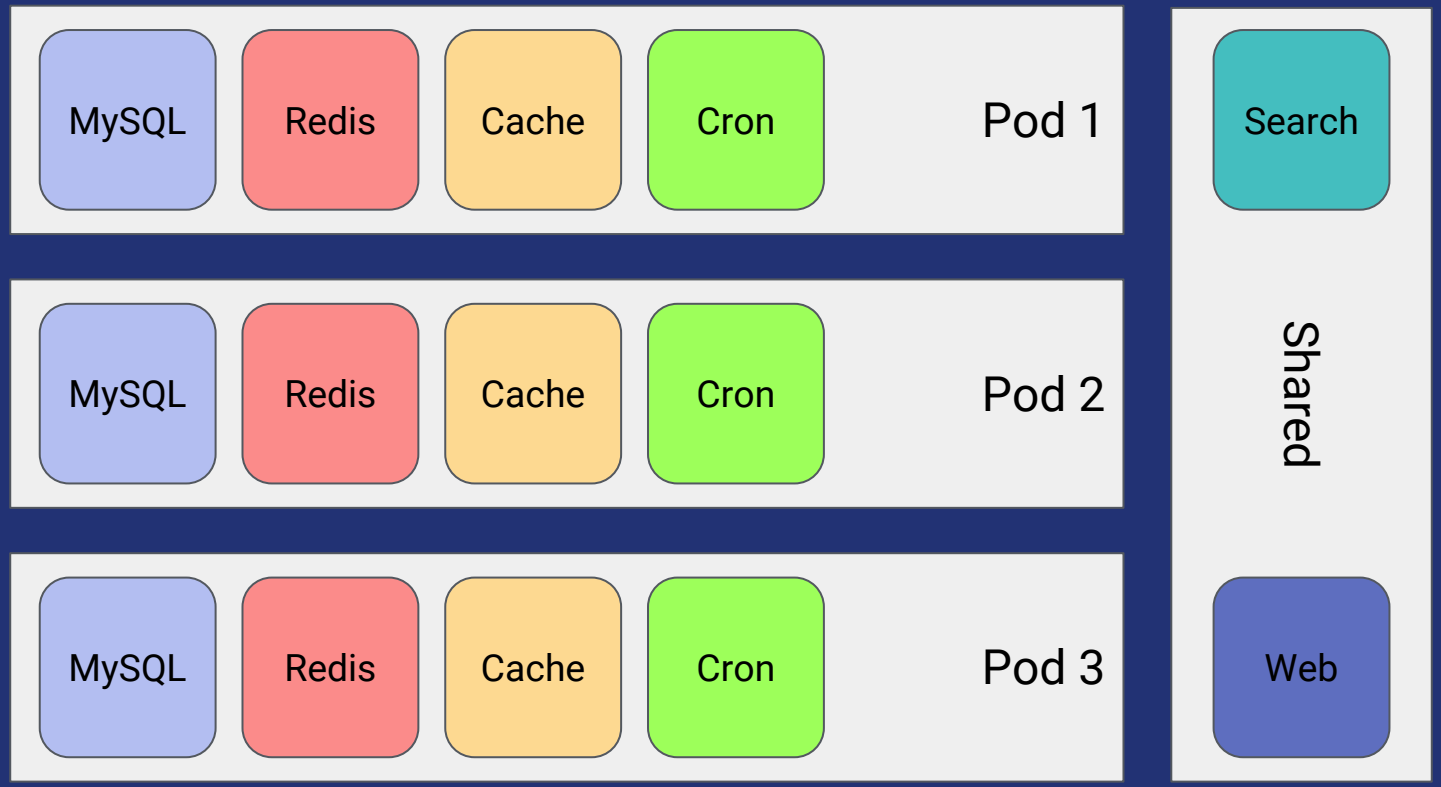
shopify



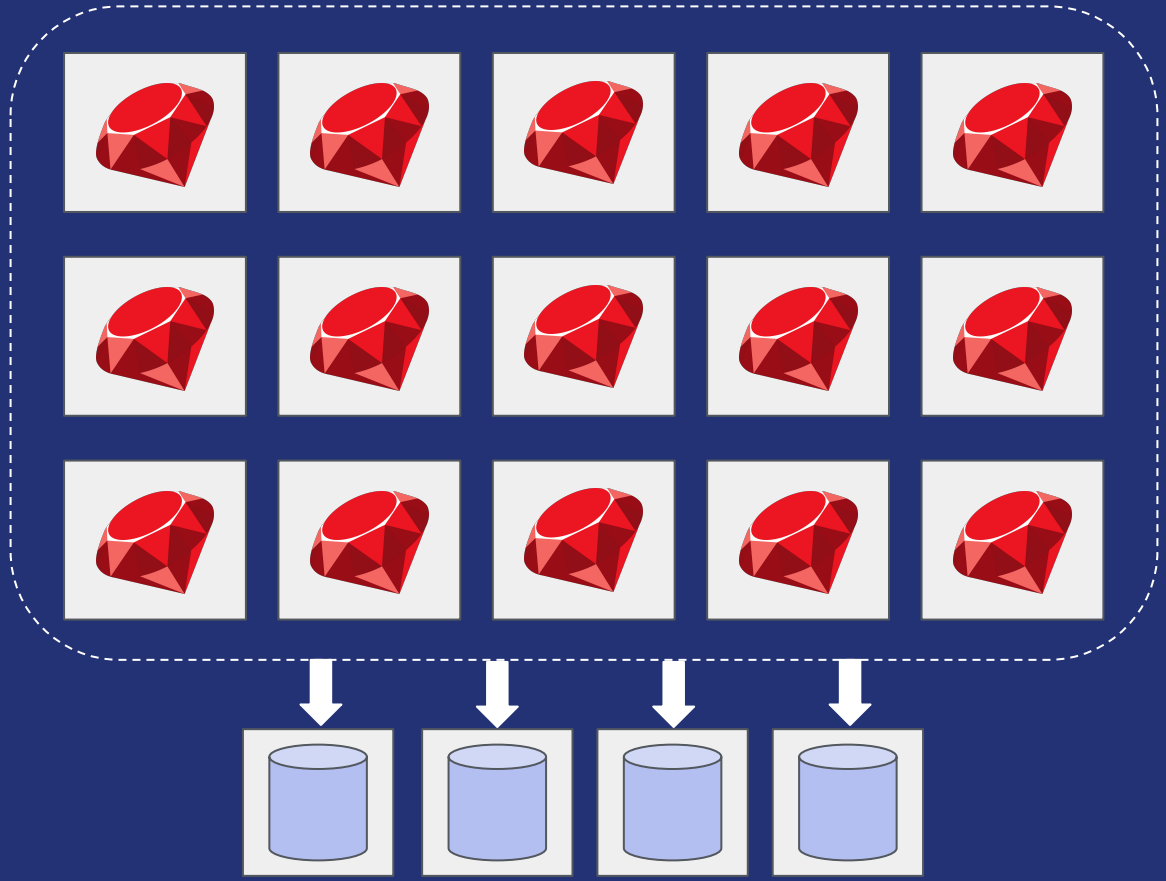
shopify


Shop
Shop
Shard 1
Shop
Shard 2
Shop
Shop
Shard 3
Master Catalog
Zookeeper

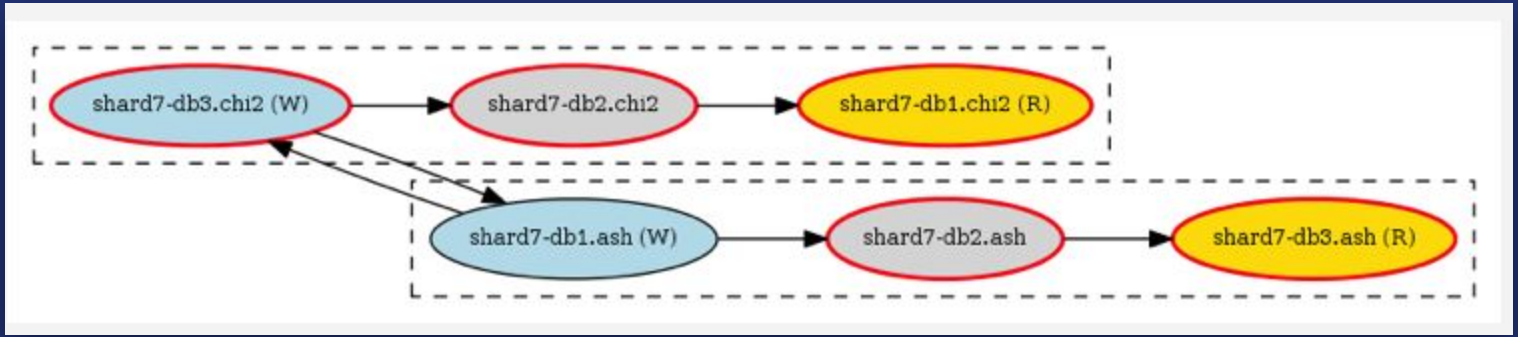
shopify


MySQL
Redis
Cache
Cron
Pod 1
MySQL
Redis
Cache
Cron
Pod 2
MySQL
Redis
Cache
Cron
Pod 3
Search
Shared
Web

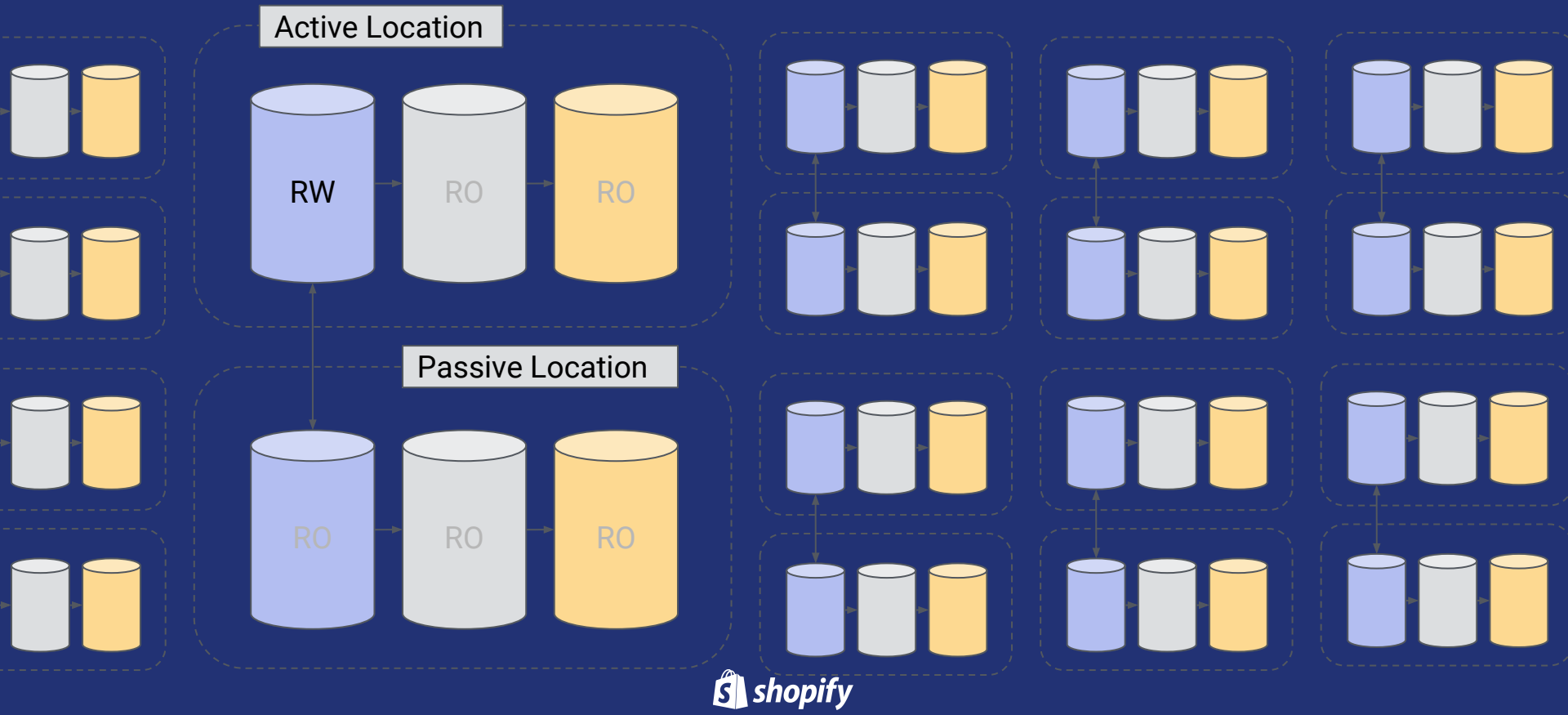
shopify

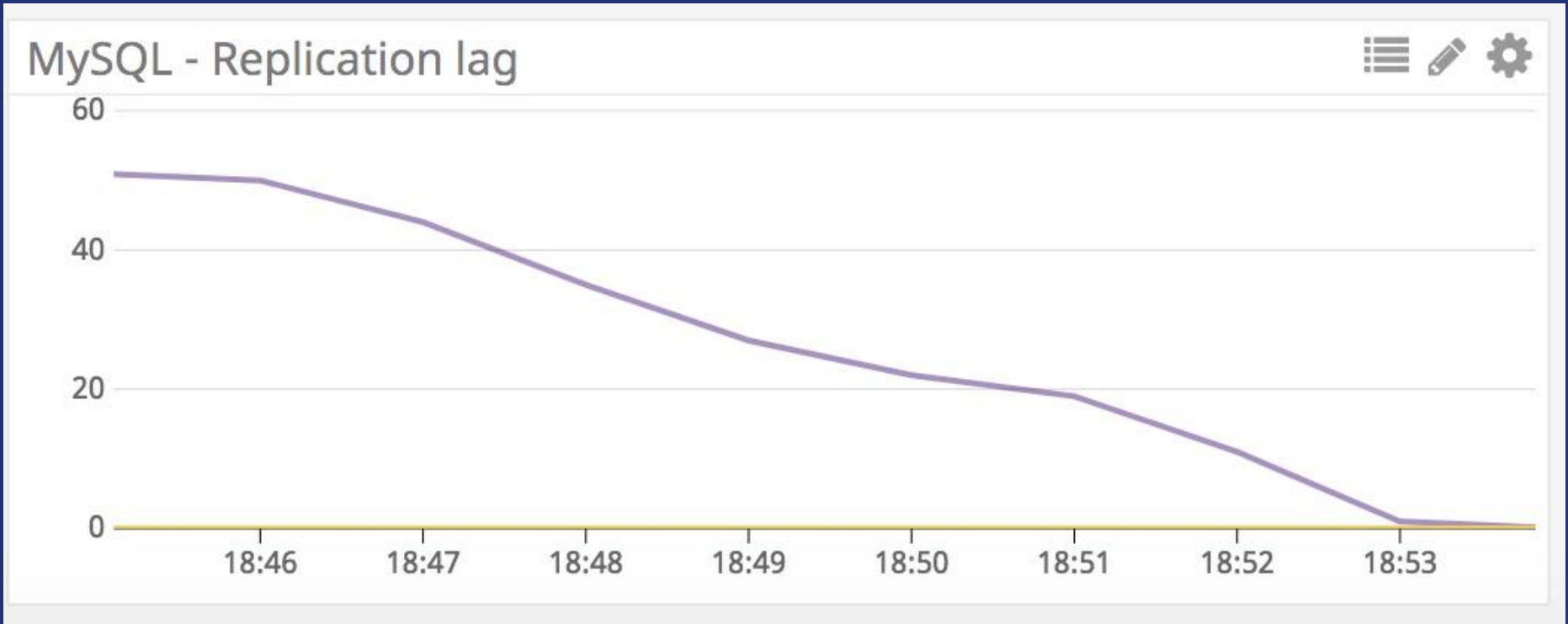


shopify


shard7-db3.chi2 (W)
shard7-db2.chi2
shard7-db1.chi2 (R)
shard7-db1.ash (W)
shard7-db2.ash
shard7-db3.ash (R)

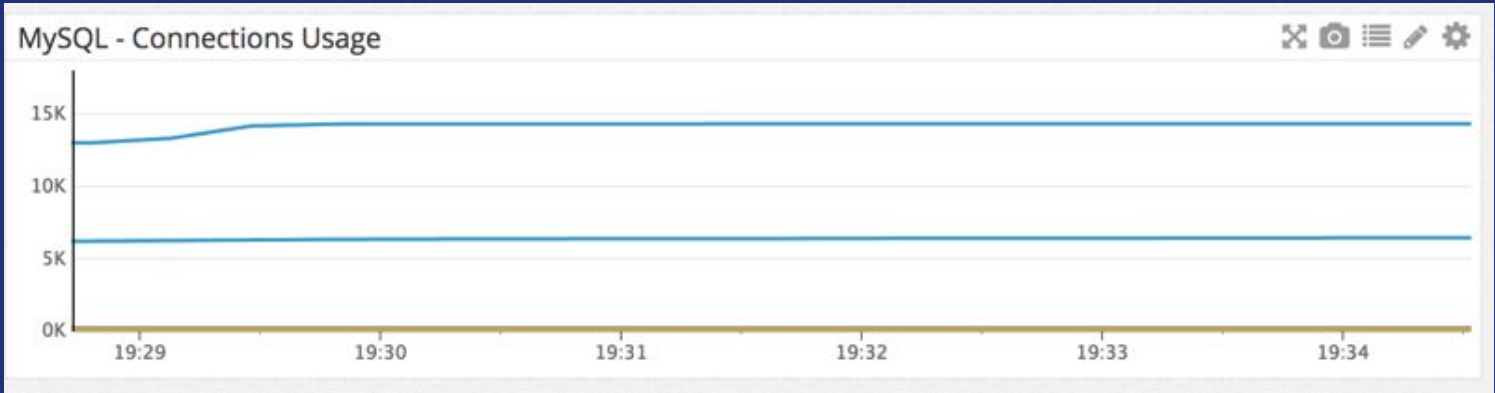
shopify


Active Location
RW
RO
RO
Passive Location
RO
RO
RO
shopify

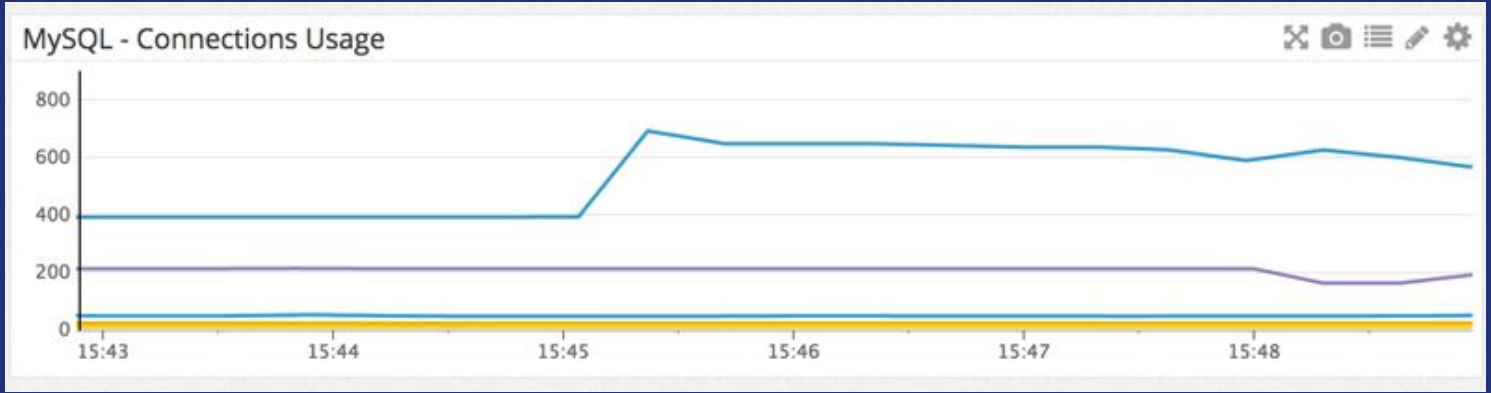

MySQL - Replication lag
60
40
20
0
18:46
18:47
18:48
18:49
18:50
18:51
18:52
18:53

shopify

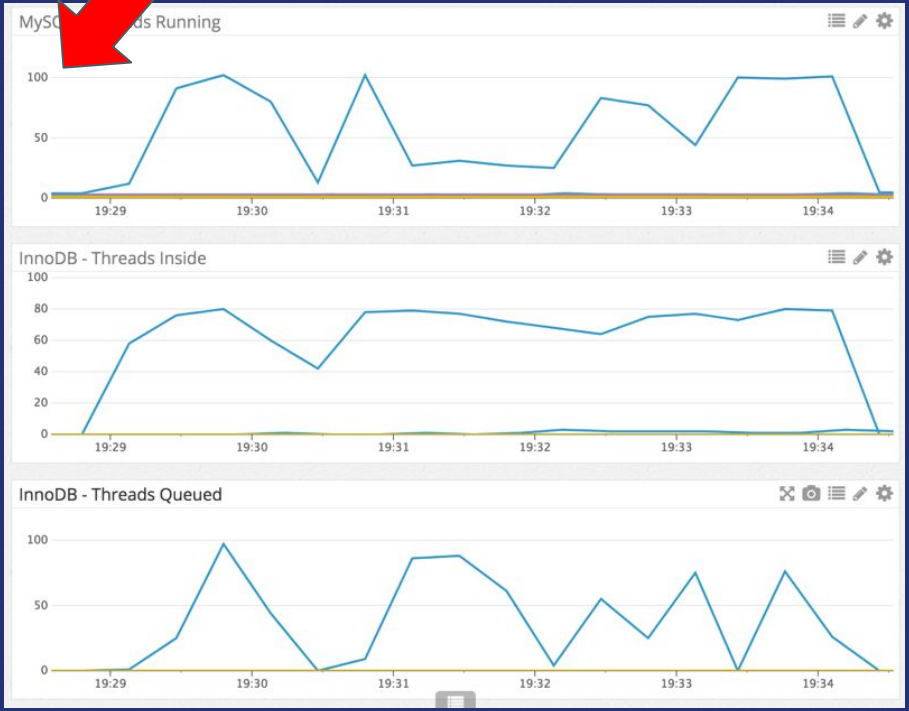
No ProxySQL:



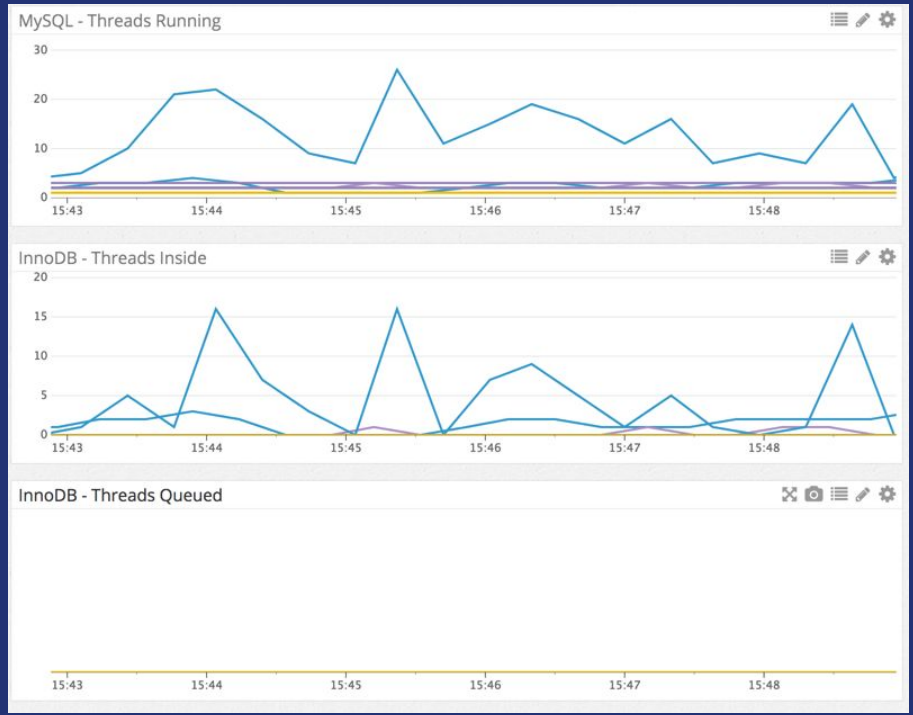
ProxySQL:



No ProxySQL:

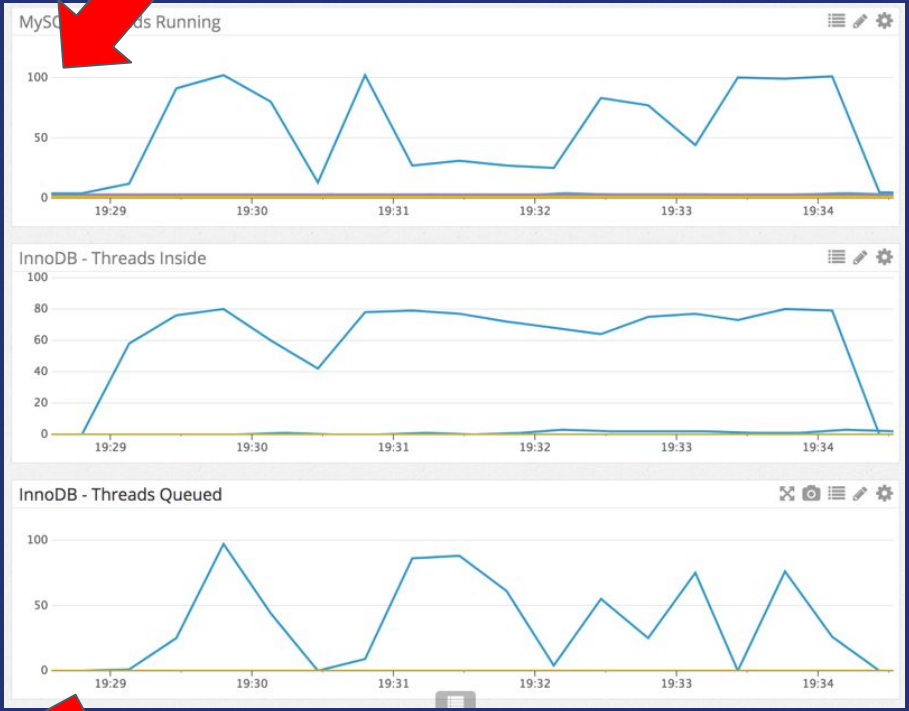


ProxySQL:

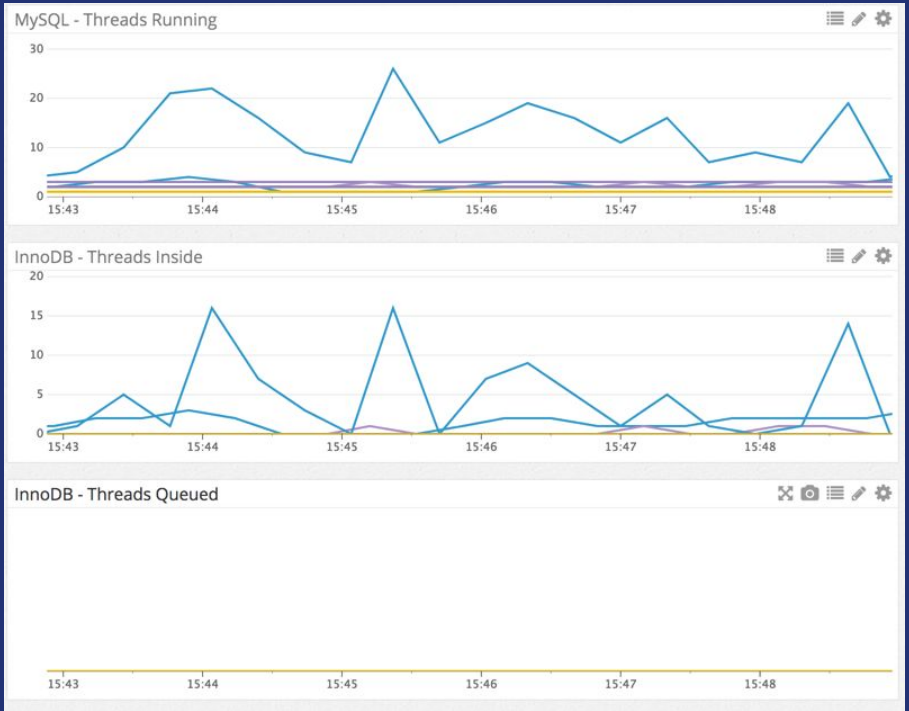


shopify

No ProxySQL:

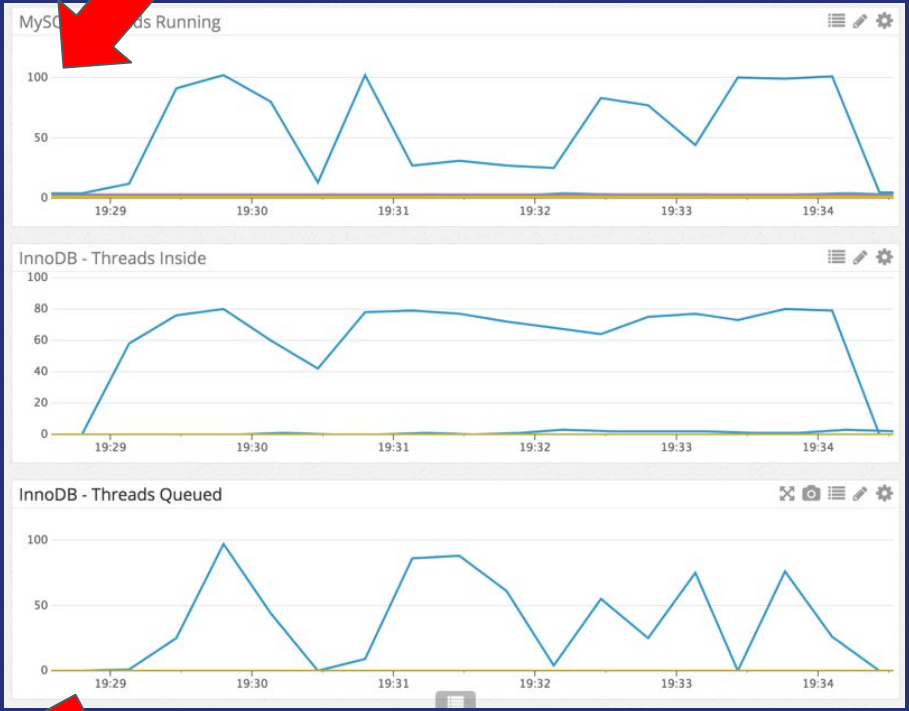


ProxySQL:

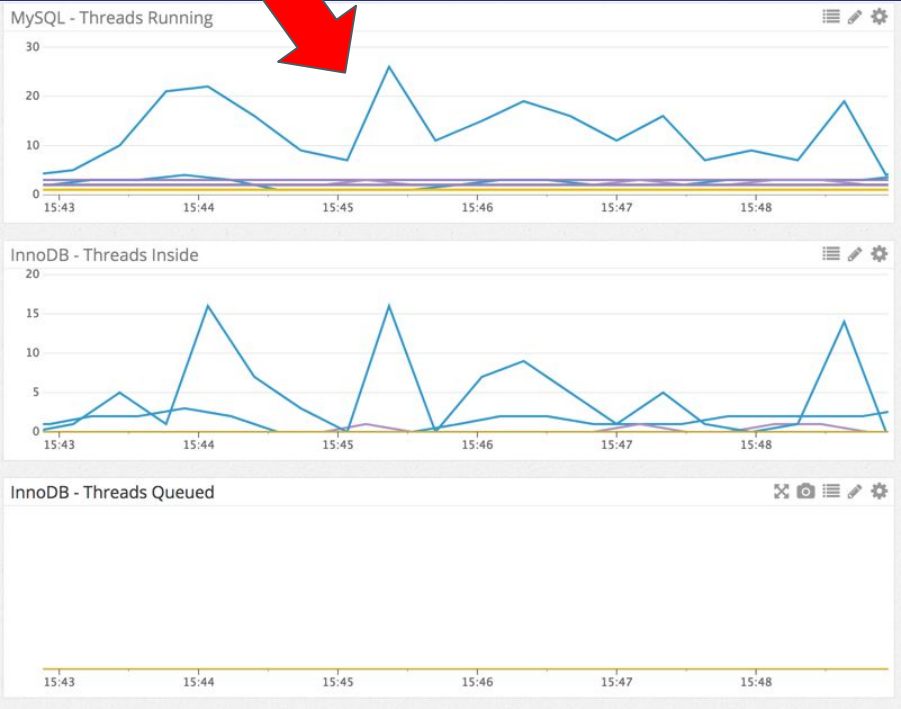


shopify

No ProxySQL:

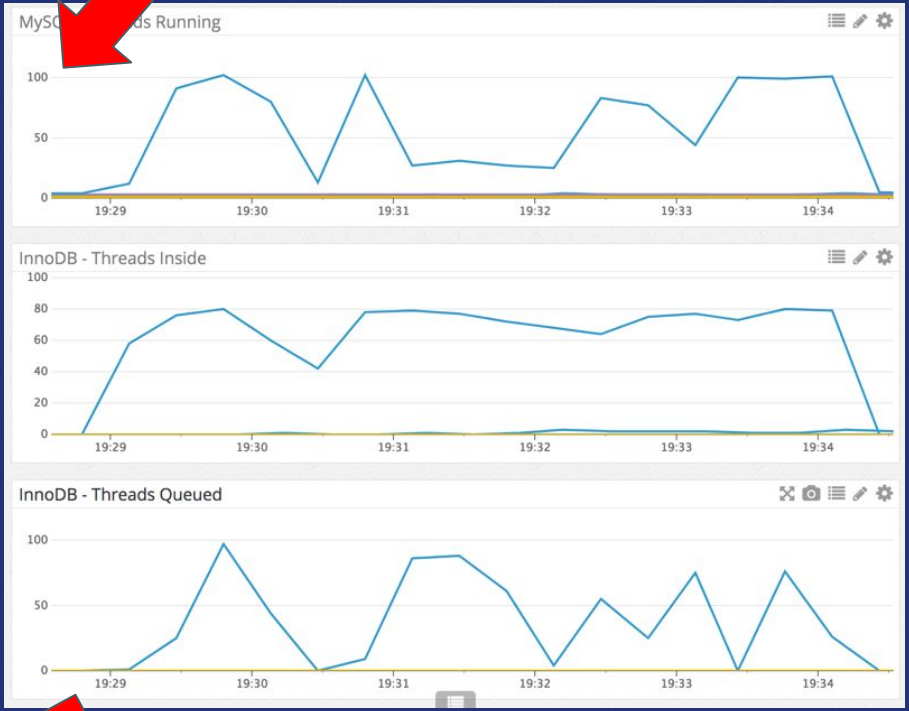


ProxySQL:

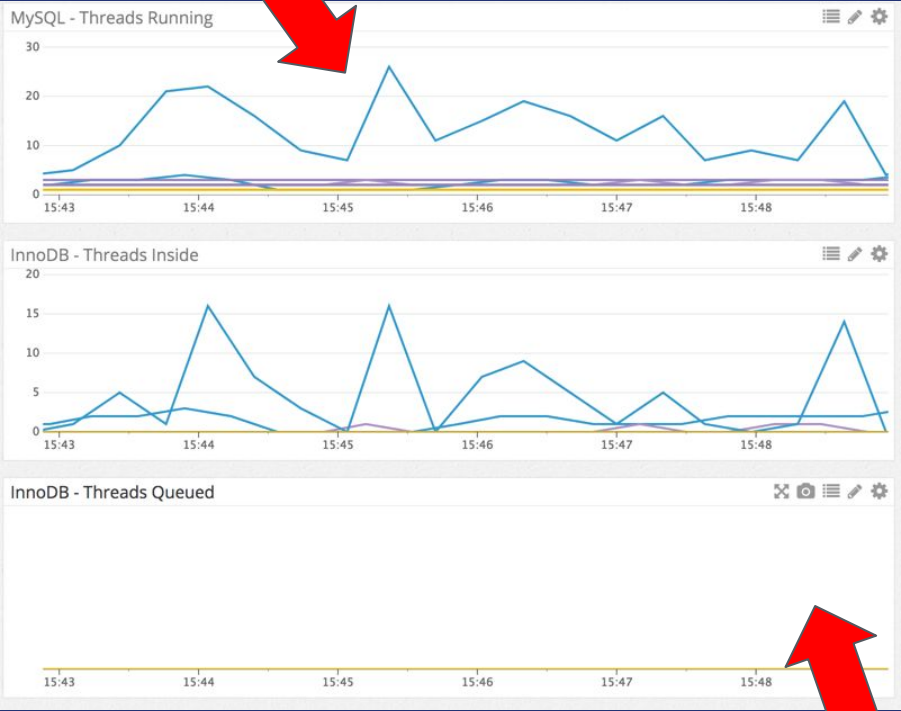


shopify

No ProxySQL:



ProxySQL:



shopify

No ProxySQL:

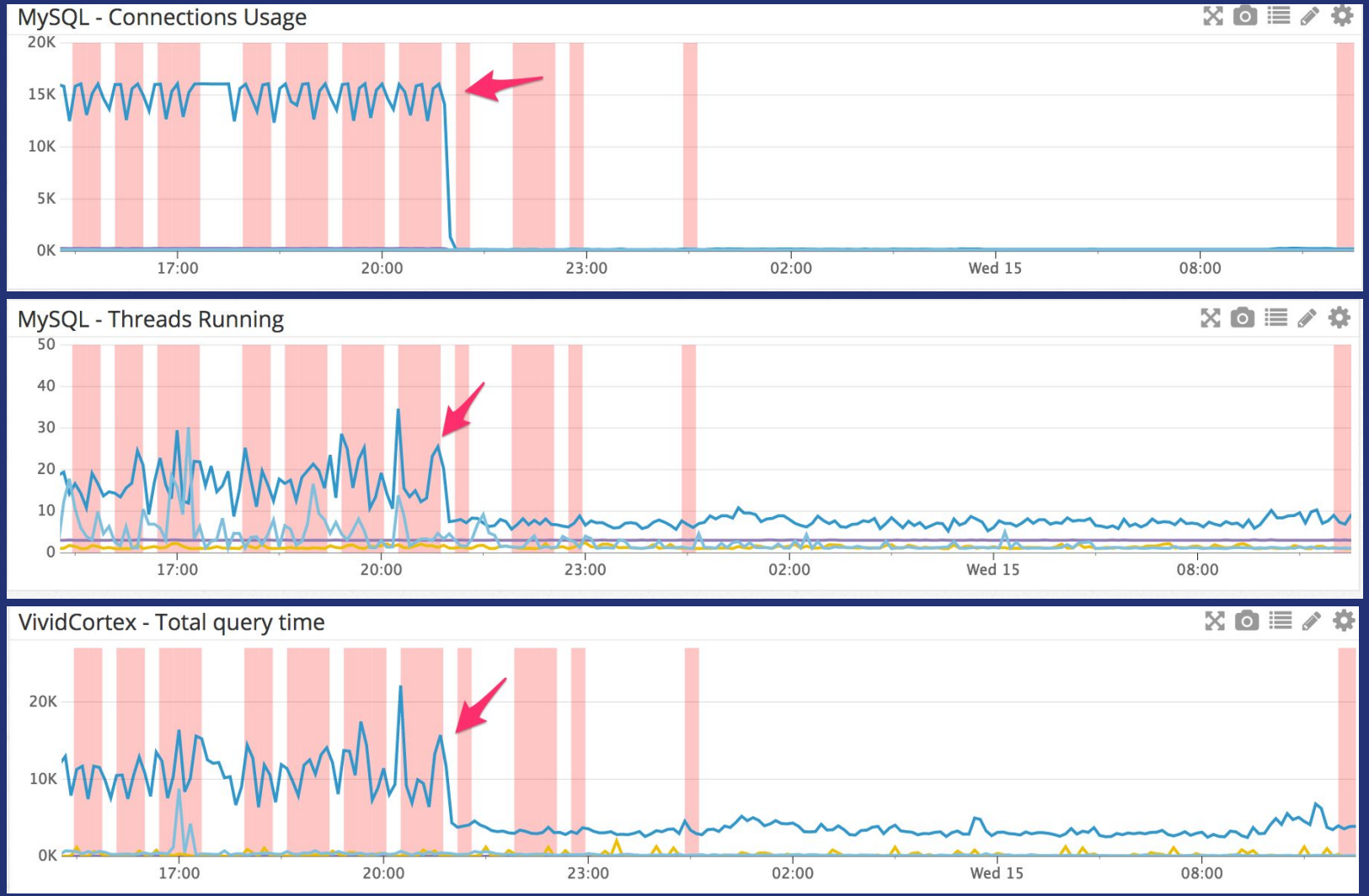
ProxySQL:

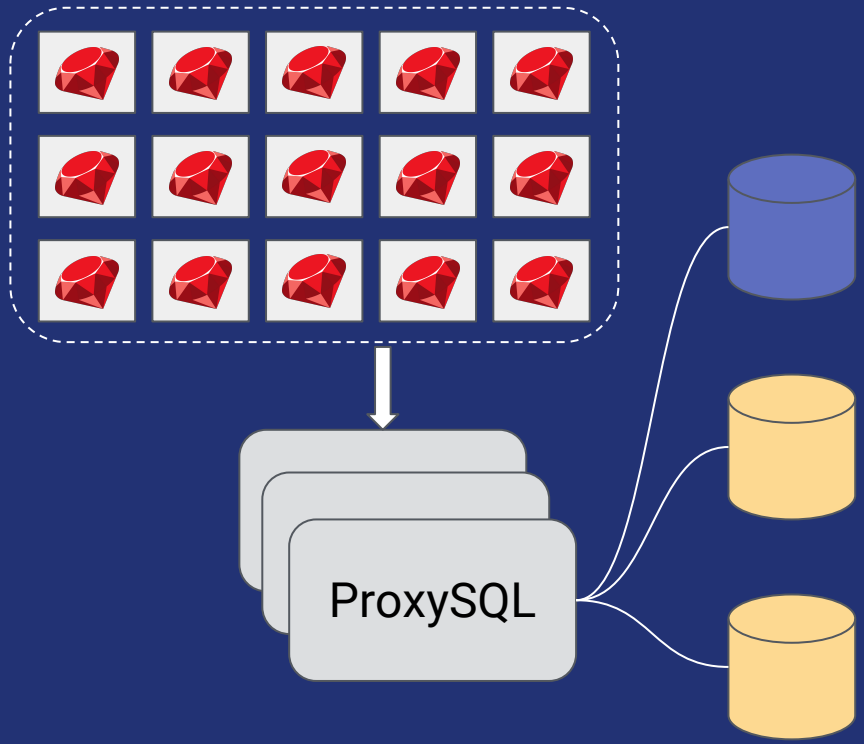
shopify

No ProxySQL:

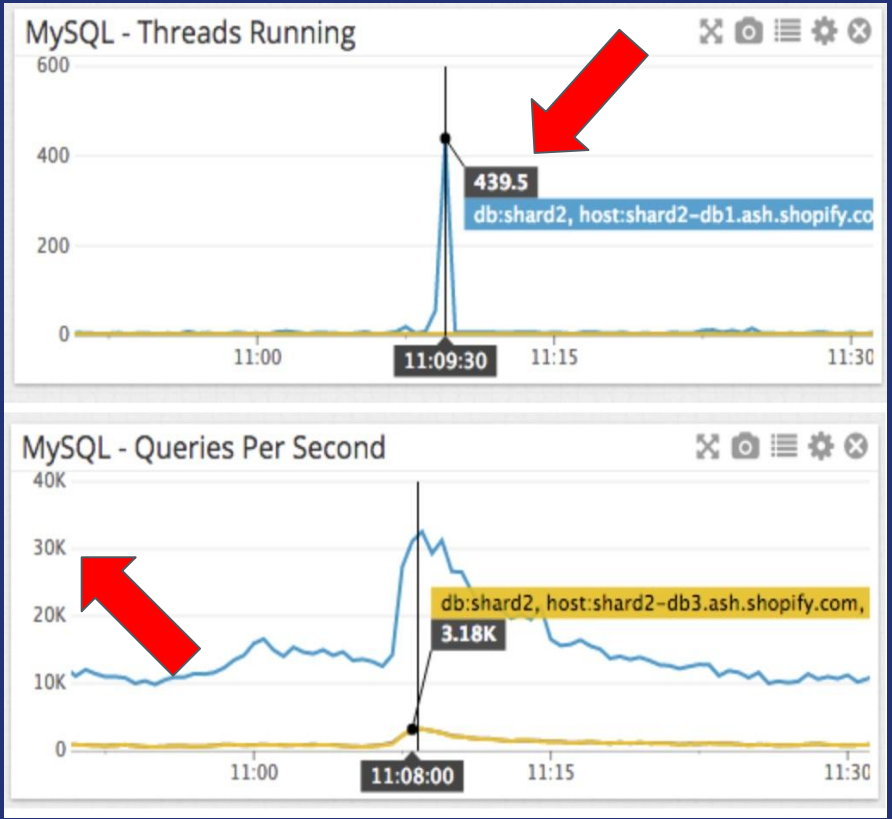
ProxySQL:

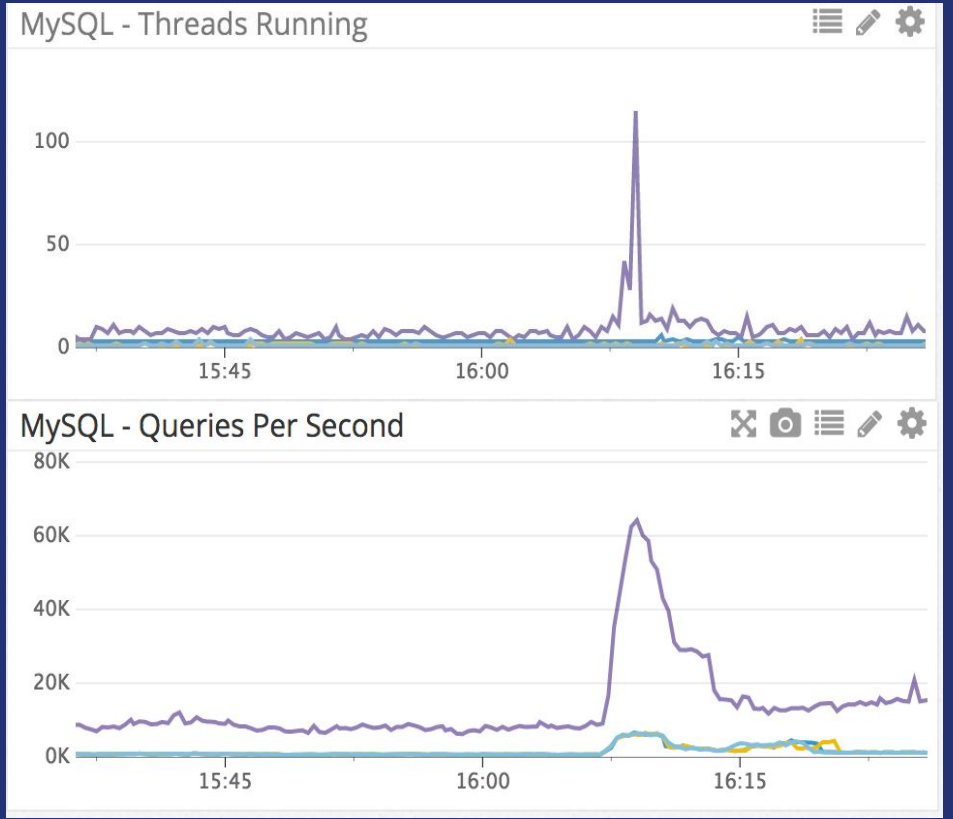
shopify


MySQL - Connections Usage
20K
15K
10K
5K
0K
17:00
20:00
23:00
02:00
Wed 15
08:00
MySQL - Threads Running
50
40
30
20
10
0
17:00
20:00
23:00
02:00
Wed 15
08:00
VividCortex - Total query time
20K
10K
0K
17:00
20:00
23:00
02:00
Wed 15
08:00

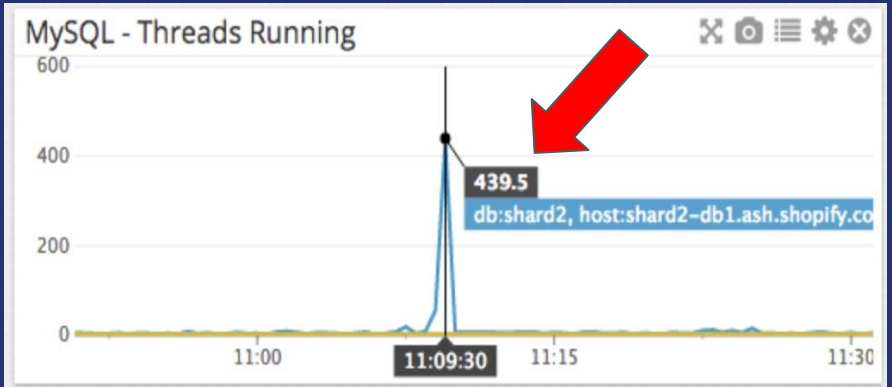


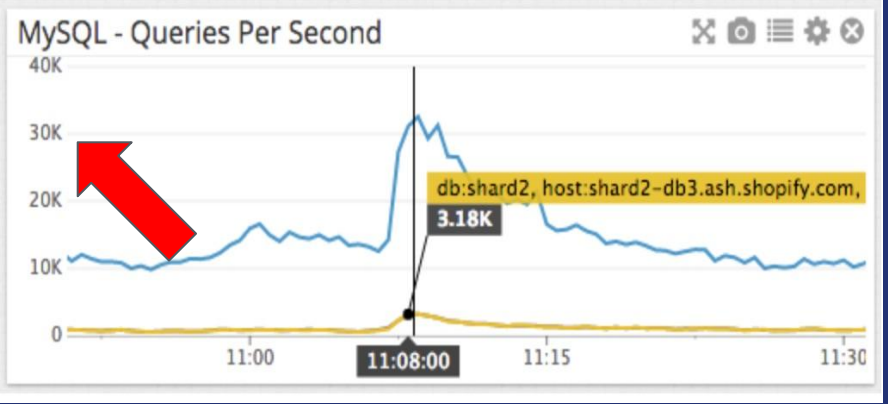
shopify

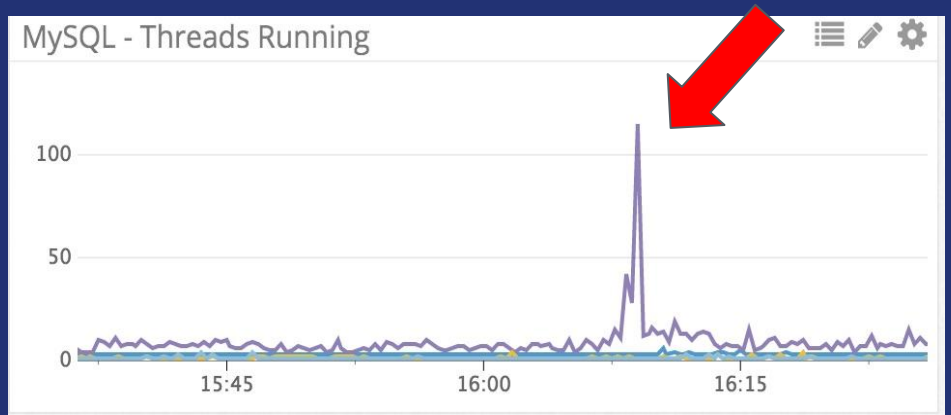

MySQL - Threads Running
600
400
200
0
439.5
db:shard2, host:shard2-db1.ash.shopify.co
11:00
11:09:30
11:15
11:30
MySQL - Queries Per Second
40K
30K
20K
10K
0
db:shard2, host:shard2-db3.ash.shopify.com,
3.18K
11:00
11:08:00
11:15
11:30

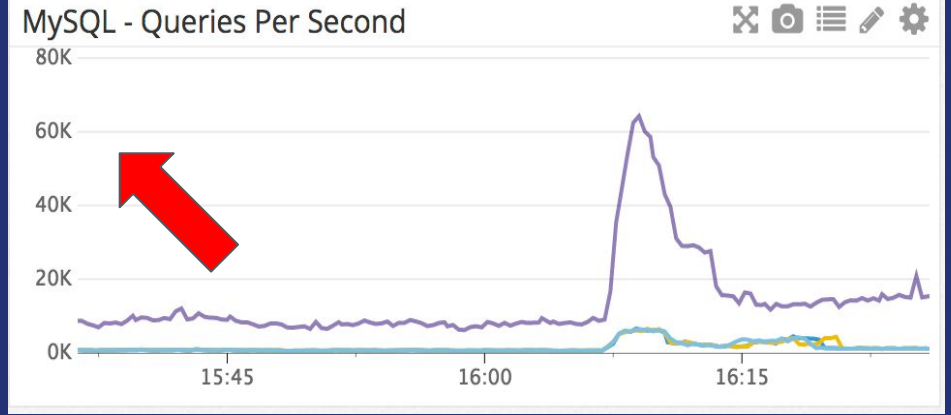

MySQL - Threads Running
100
50
0
15:45
16:00
16:15
MySQL - Queries Per Second
80K
60K
40K
20K
0K
15:45
16:00
16:15

shopify


MySQL - Threads Running
600
400
200
0
439.5
db:shard2, host:shard2-db1.ash.shopify.co
11:00
11:09:30
11:15
11:30


MySQL - Queries Per Second
40K
30K
20K
10K
0
db:shard2, host:shard2-db3.ash.shopify.com,
3.18K
11:00
11:08:00
11:15
11:30


MySQL - Threads Running
100
50
0
15:45
16:00
16:15


MySQL - Queries Per Second
80K
60K
40K
20K
0K
15:45
16:00
16:15

shopify


[redacted] [redacted] 11:14 AM
fucking 💯 job friends
that was definitely the smoothest flash sale at that volume i've ever seen

shopify

Cost of MySQL’s one-thread-per-connection

Too many software threads per hardware thread
CPU registries save/restore and context switching
Mutexes/locks contentions
CPU cache almost useless
High cost for access to memory
Avoid having a central bottleneck

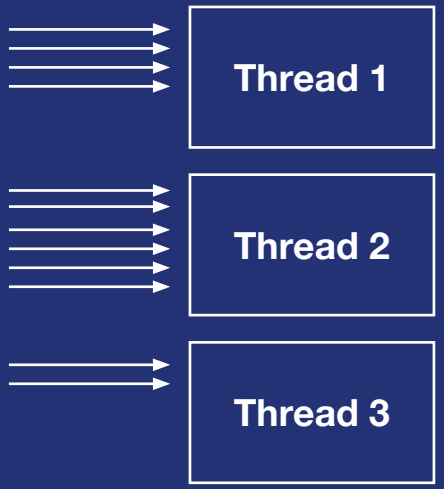
Thread pool in MySQL

shopify

Thread pool in MySQL

Threads in ProxySQL are known as "MySQL Threads"
Fixed number of worker threads (configurable)
All threads listen on the same port(s)
Client connections are not shared between threads
All threads perform their own network I/O
Uses "poll()"... (does it scale?)

Threads never share client connections



Pros:
Thread contention is reduced
No need for synchronization
Each thread calls "poll()"

Cons:
Possibly imbalanced load

shopify

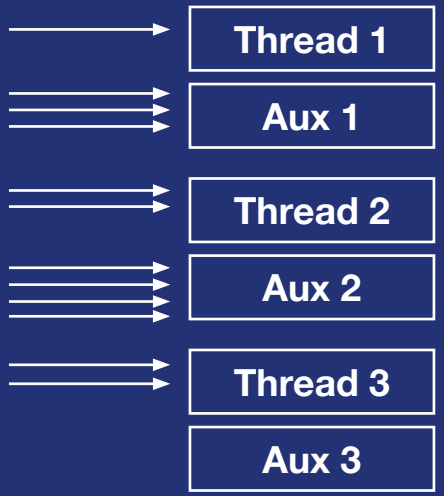
poll() vs. epoll()

"poll()" is O(N)
"epoll()" is O(1)
"epoll()" scales better than "poll()"

Why does ProxySQL use "poll()"?
- It is faster than "epoll()" for fewer connections (~1000)
- Performance degrades when there are a lot of connections

shopify

ProxySQL Auxiliary Threads

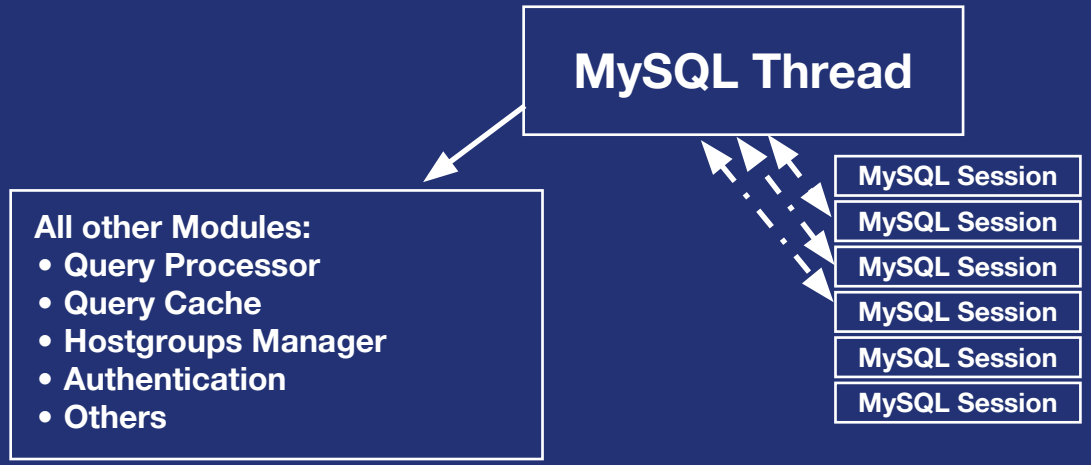


Each worker thread has an auxiliary thread
Worker thread uses "poll()"
Auxiliary thread uses "epoll()"
Worker thread passes idle connections to auxiliary thread
When a connections becomes active auxiliary thread passes connection to the worker thread

Solution scales to 1 million connections

shopify

MySQL Thread Overview



For low contention, threads independently:

- Track internal metrics
- Store values for mysql-XXX variables
- Store a **copy** of the defined query rules

shopify

Contention on MyHGM

MyHGM is a shared resource so it can cause **contention** when accessed by **MySQL Threads**

shopify

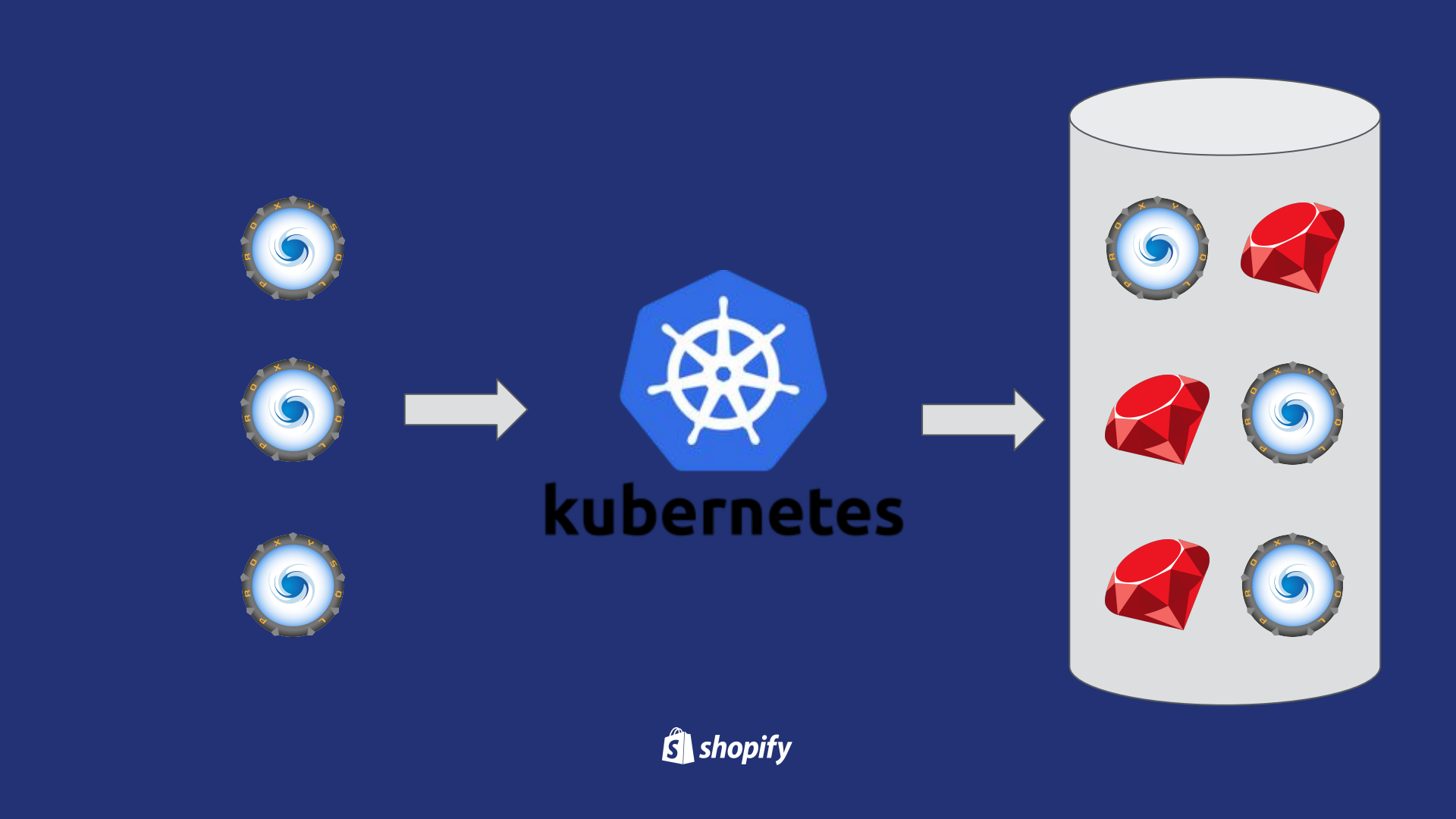
Thread Connection Cache

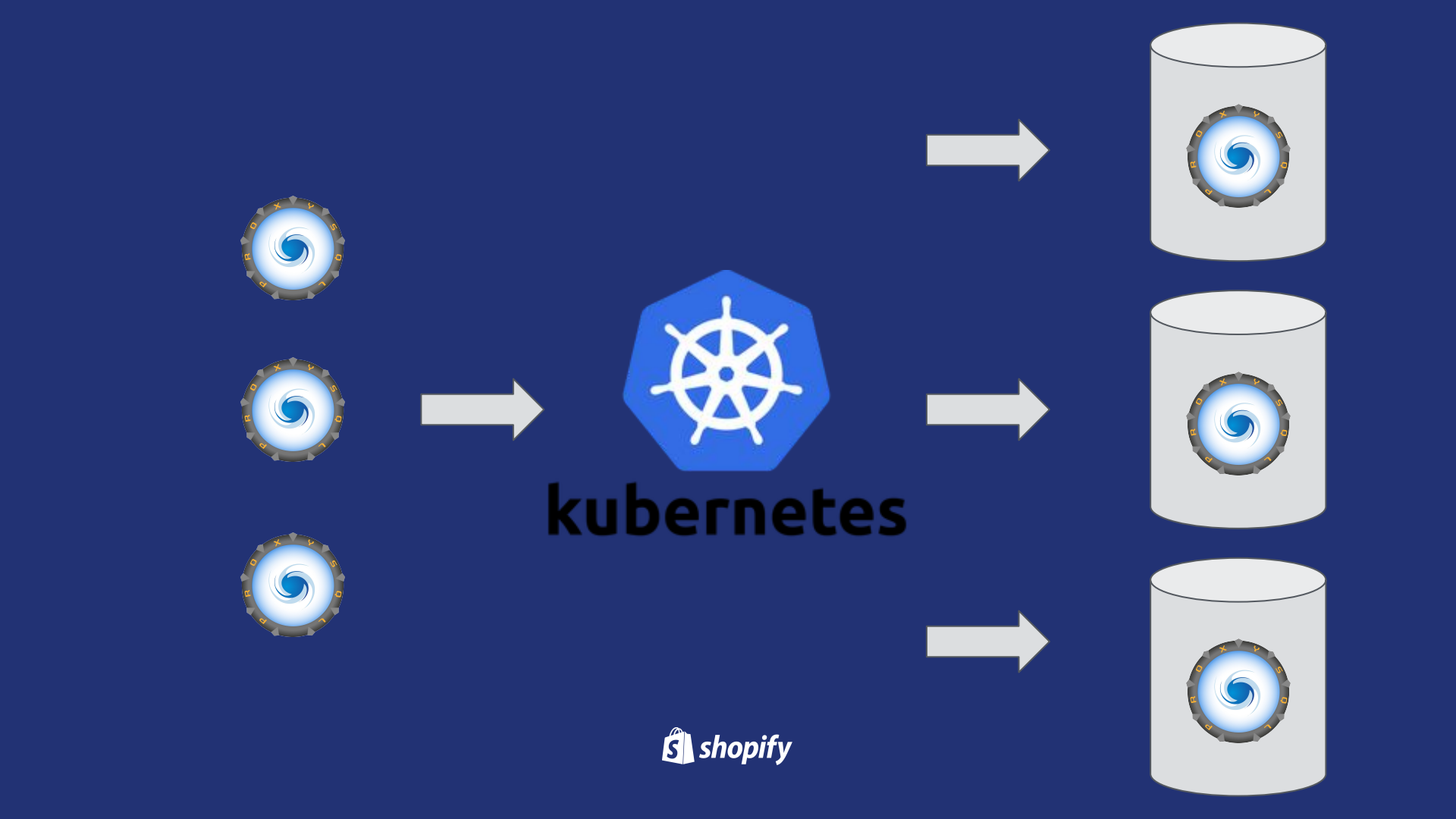
Each MySQL Thread has a connection cache that is reset before calling poll()

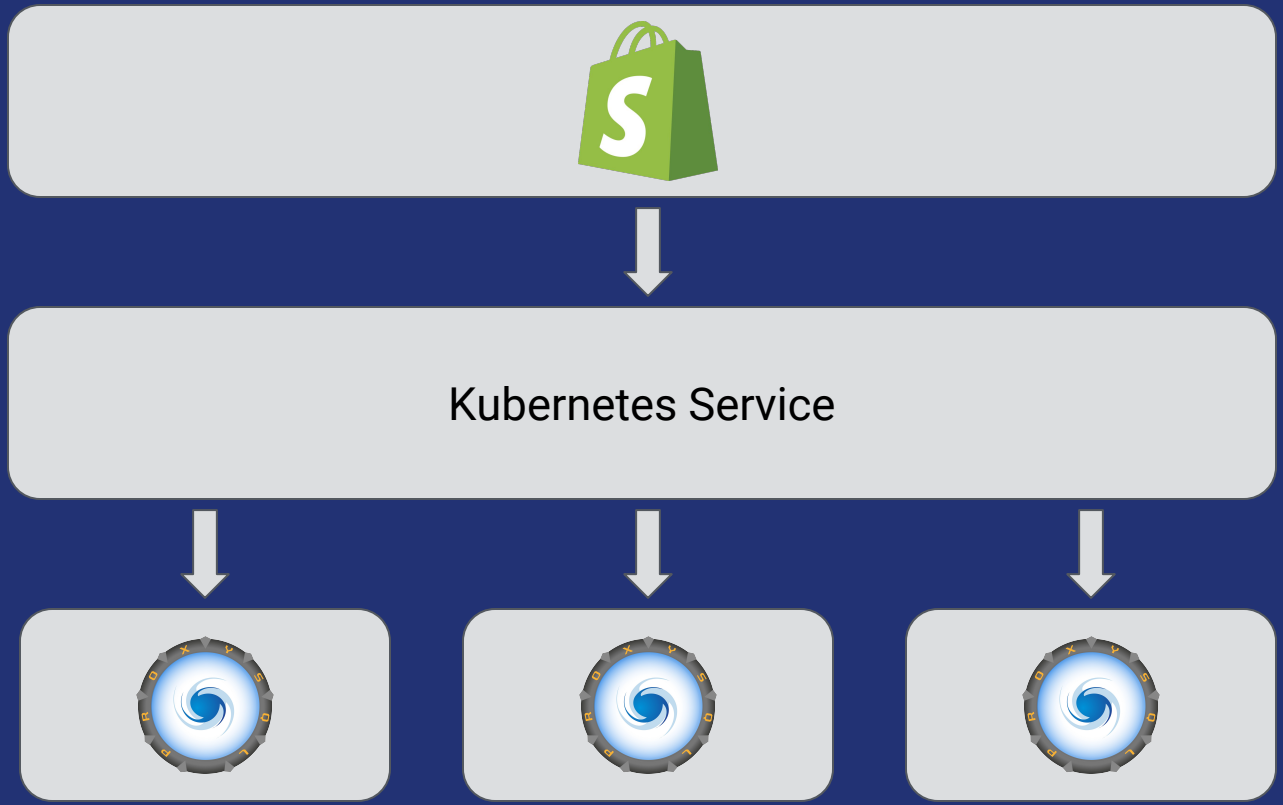
shopify

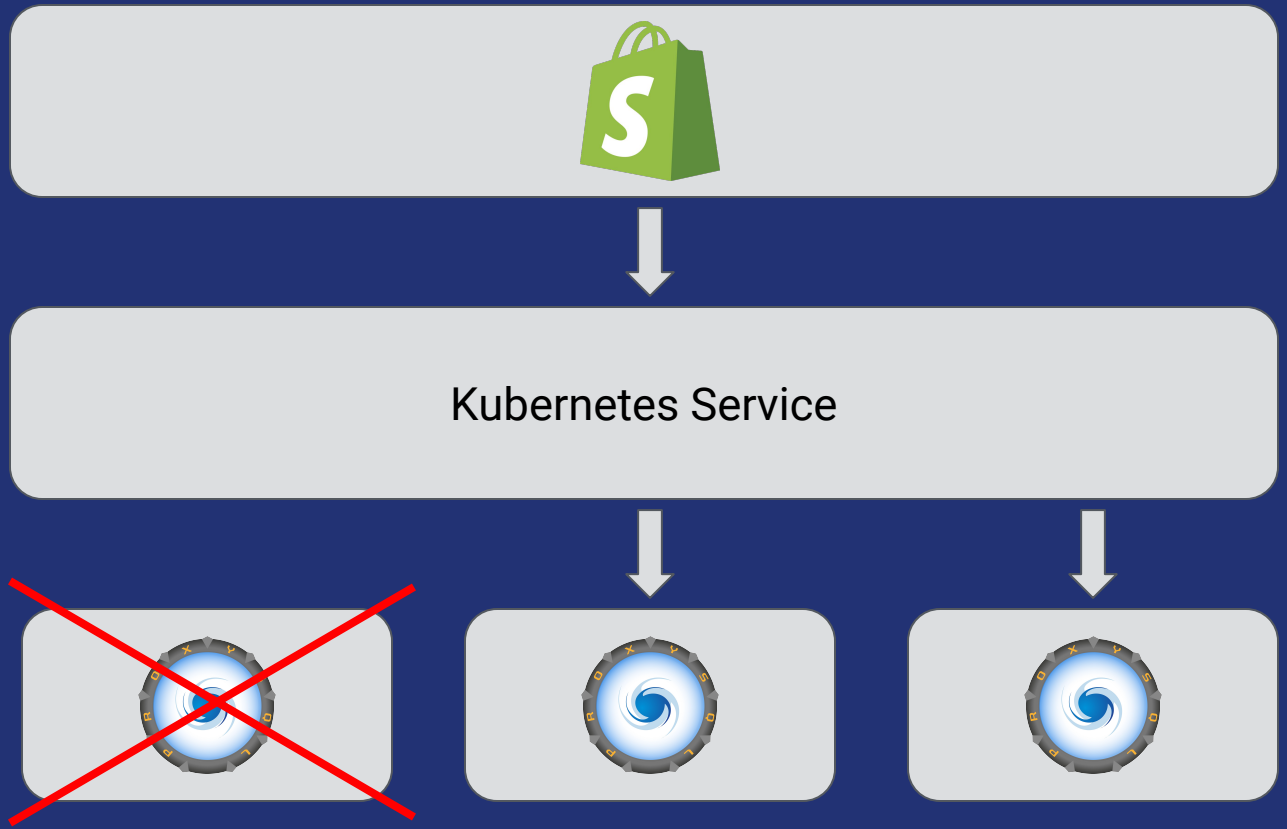
kubernetes

shopify

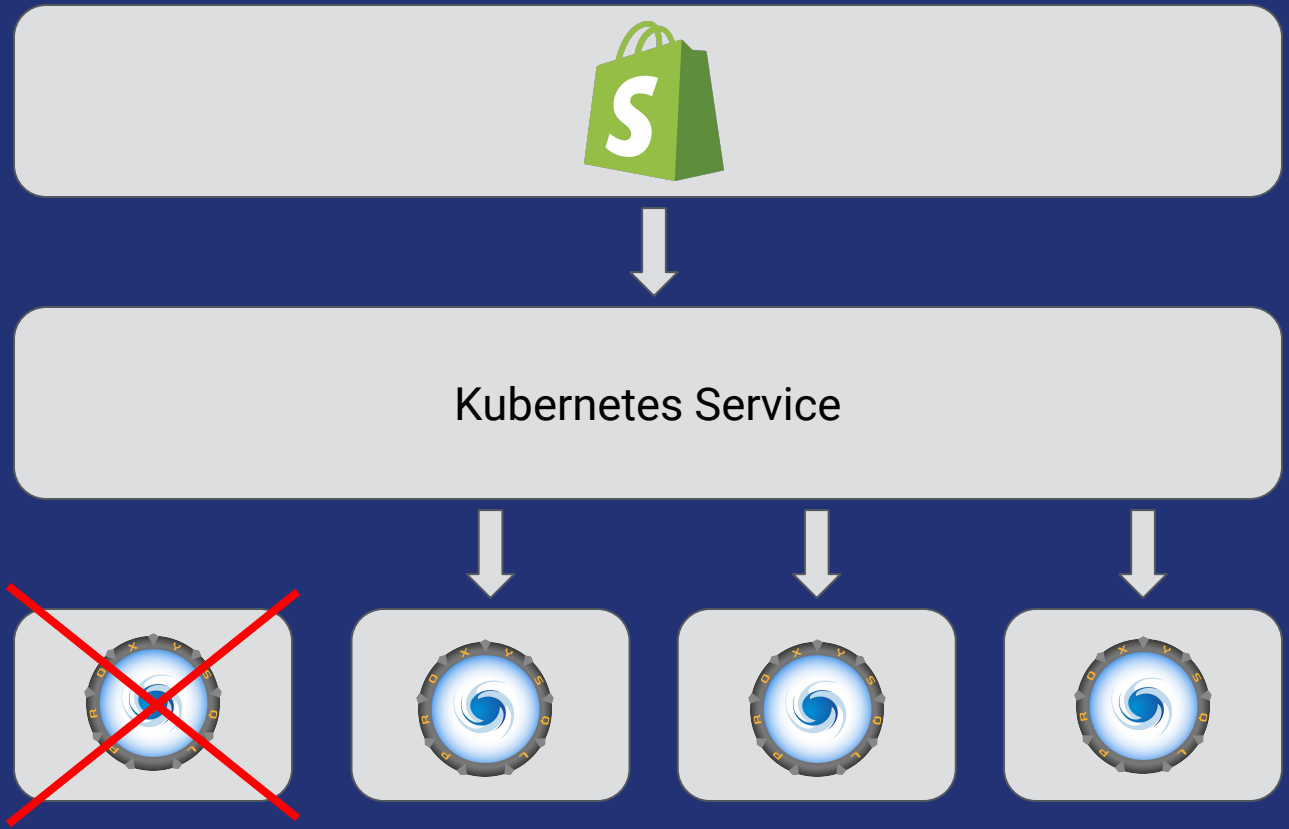

kubernetes
shopify


kubernetes
shopify

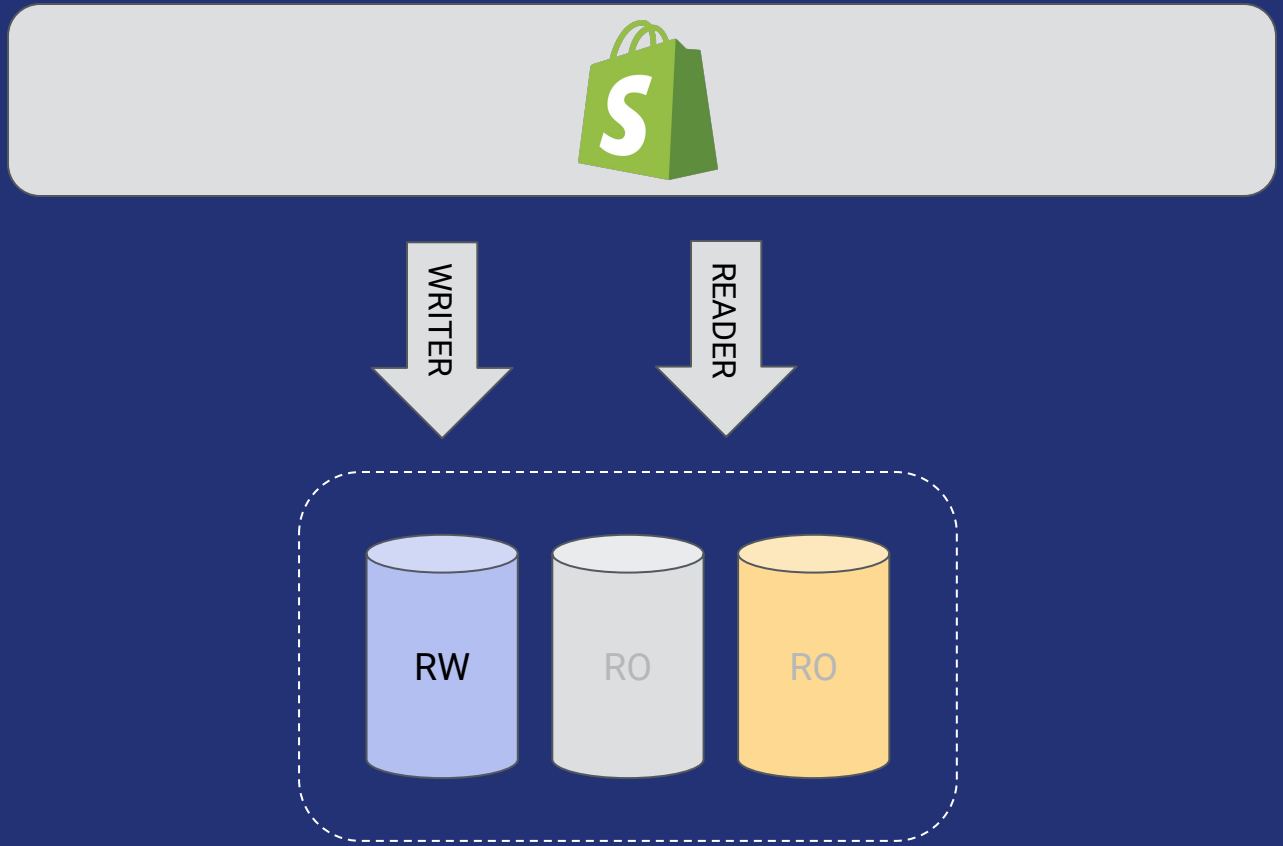

Kubernetes Service
shopify



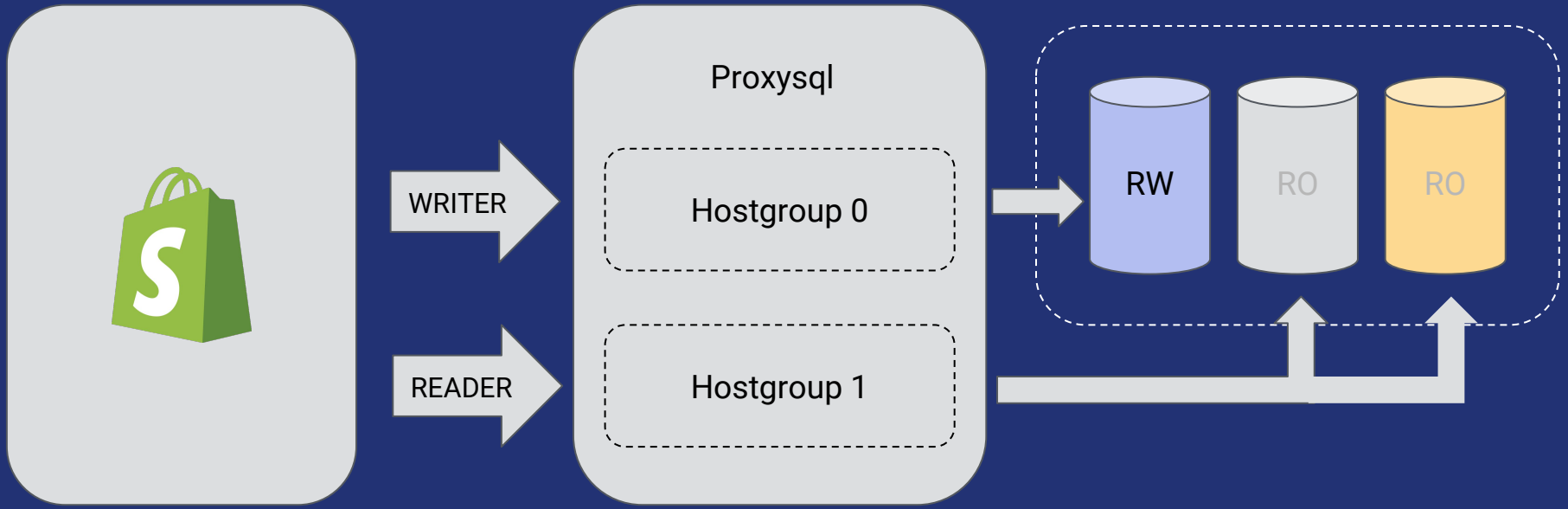
shopify



shopify

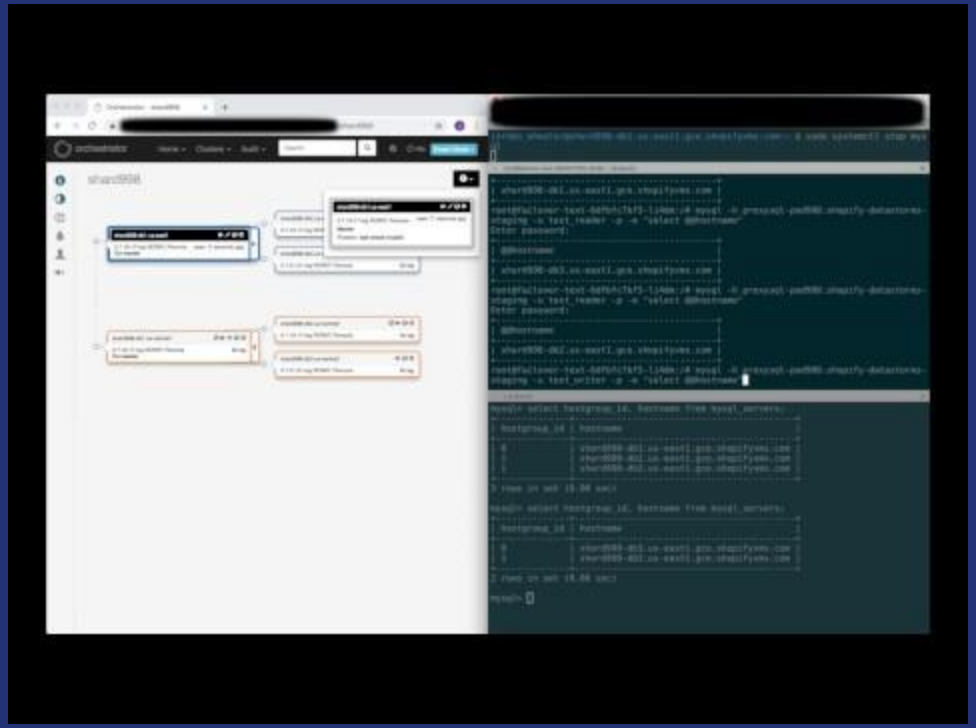


shopify

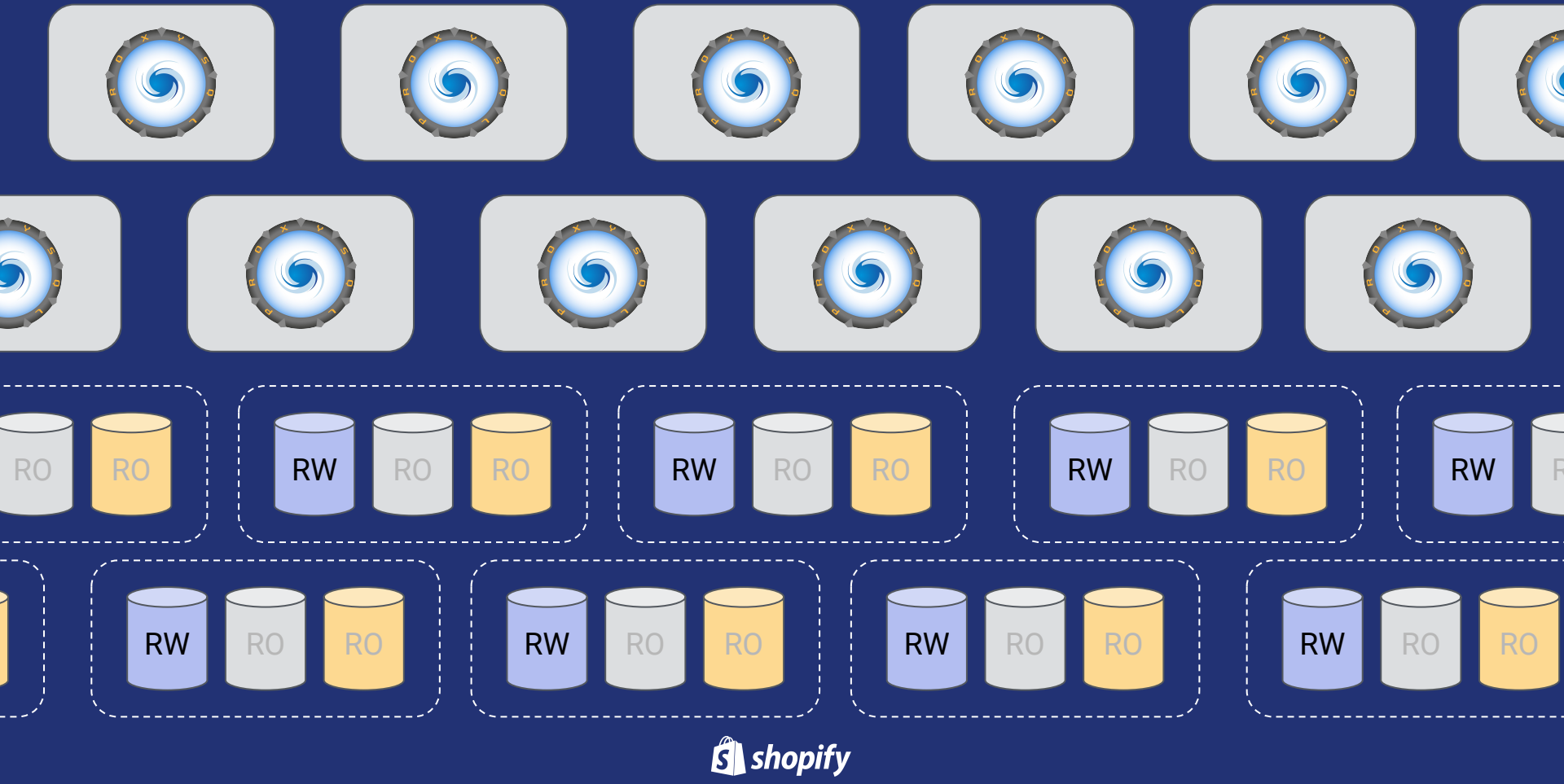


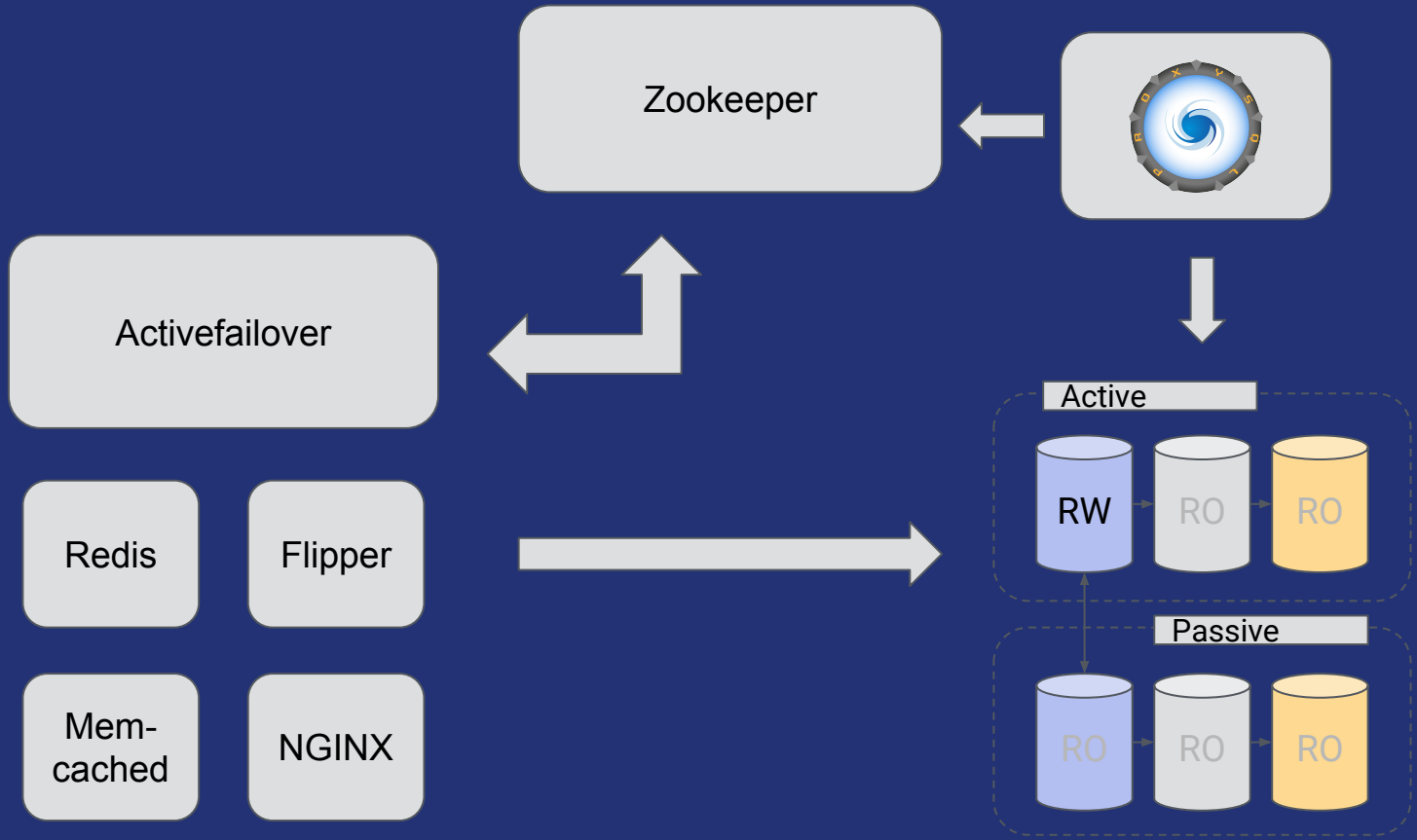
shopify

shopify

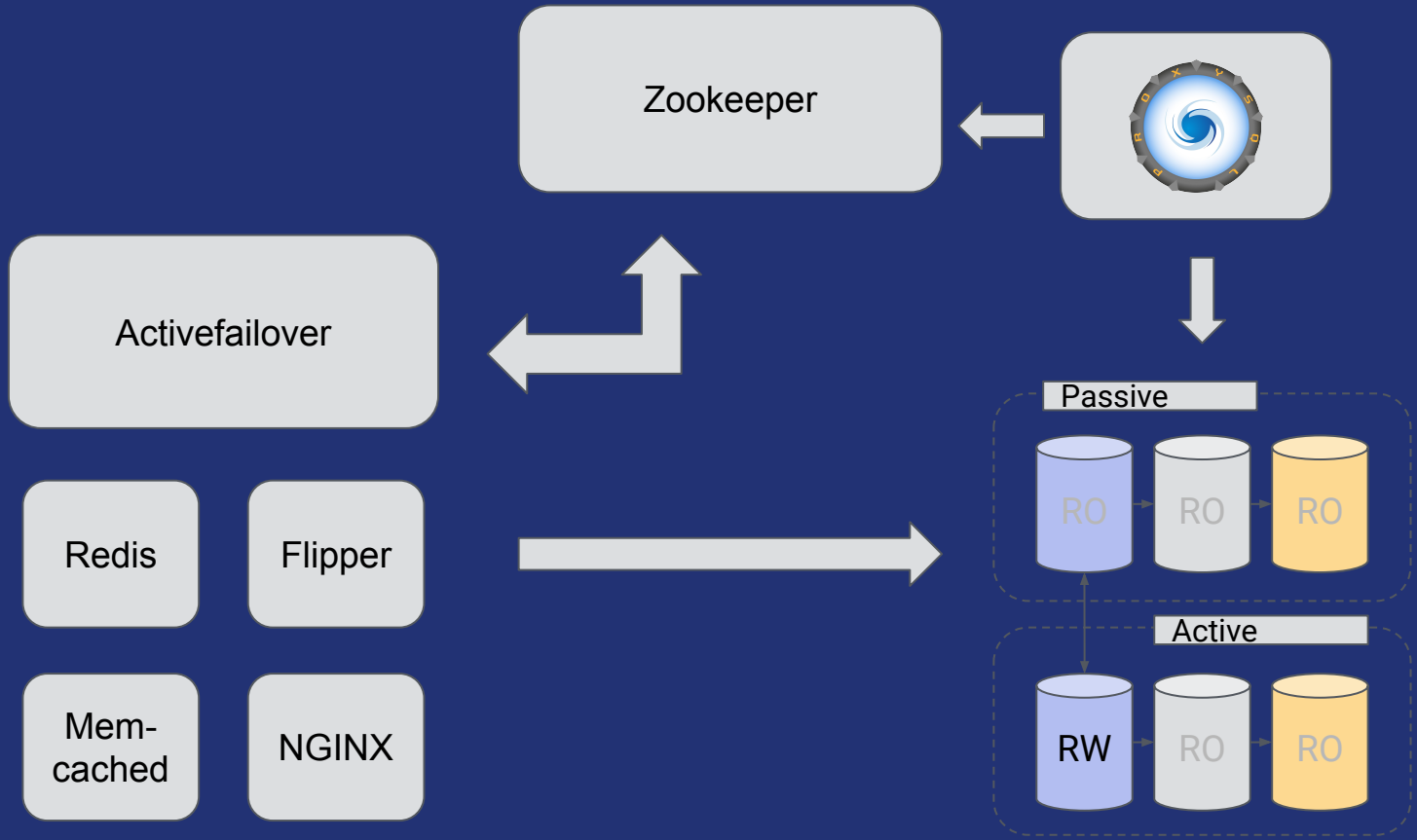


shopify


RO
RO
RW
RO
RO
RW
RO
RO
RW
RO
RO
RW
RO
RO
RW
RO
RO
RW
RO
RO
RW
RO
RO
shopify


Zookeeper
Activefailover
Redis
Flipper
Mem-
cached
NGINX
Active
RW
RO
RO
Passive
RO
RO
RO

shopify

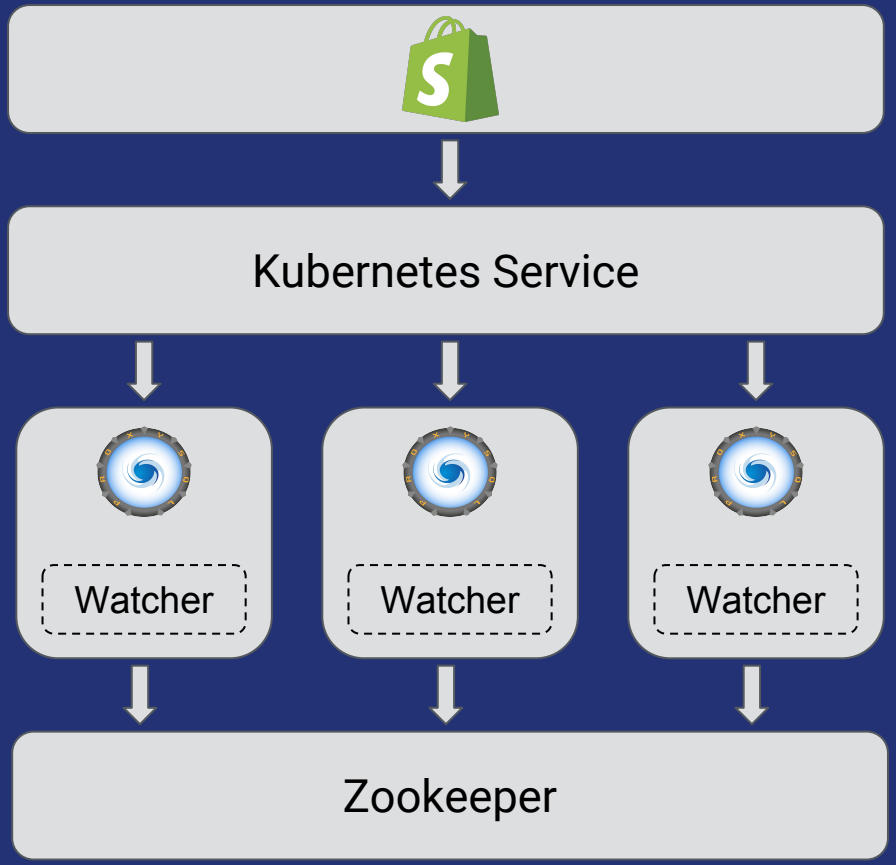

Zookeeper
Activefailover
Redis
Flipper
Mem-
cached
NGINX
Passive
RO
RO
RO
Active
RW
RO
RO

shopify

Taiji Service Discovery

MySQL

shopify



shopify

shard123-db1

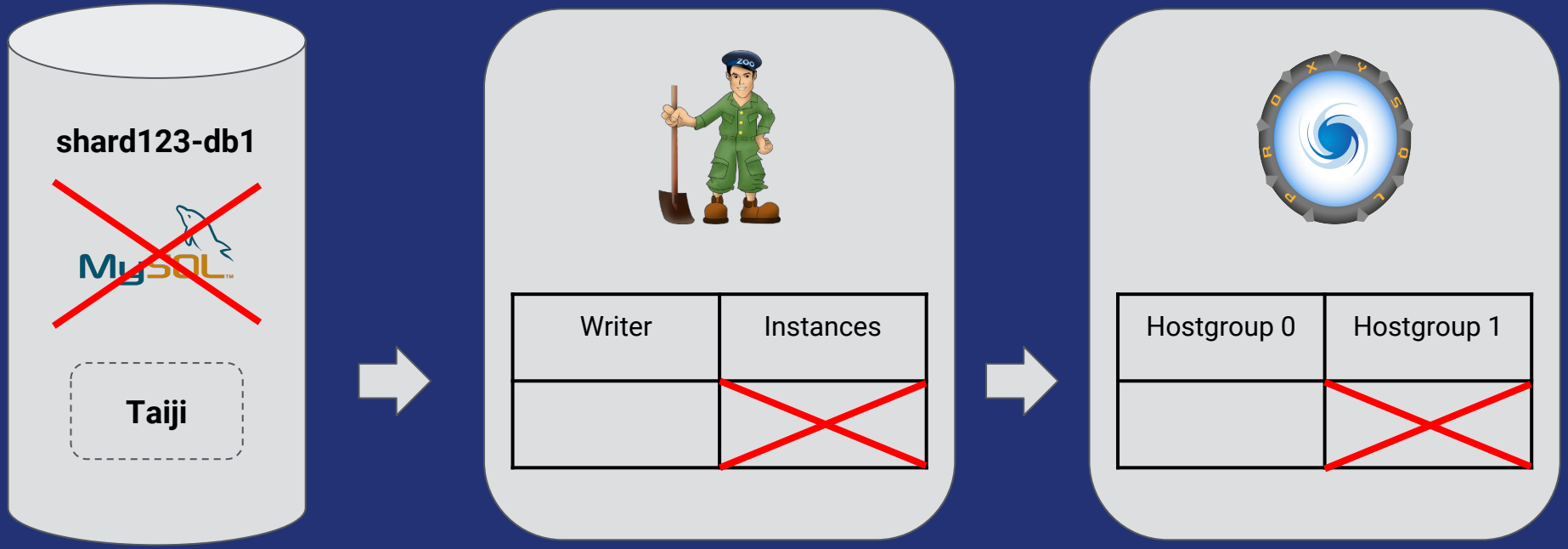
MySQL

Taiji

Writer	Instances
	shard123-db1

Hostgroup 0	Hostgroup 1
	shard123-db1

shopify


shard123-db1
MySQL
Taiji
Writer
Instances
Hostgroup 0
Hostgroup 1

shopify

shard123-db1

MySQL

~~Taiji~~

Writer	Instances
	shard123-db1

Hostgroup 0	Hostgroup 1
	shard123-db1

shopify

shard123-db1

MySQL

~~Taiji~~

Writer	Instances
	~~shard123-db1~~

Hostgroup 0	Hostgroup 1
	shard123-db1

shopify

shard123-db1

MySQL

Taiji

Writer	Instances

Hostgroup 0	Hostgroup 1
	shard123-db1

shopify

shard123-db1

MySQL

~~Taiji~~

Writer	Instances
	~~shard123-db1~~

Hostgroup 0	Hostgroup 1
	shard123-db1

shopify

shard123-db1

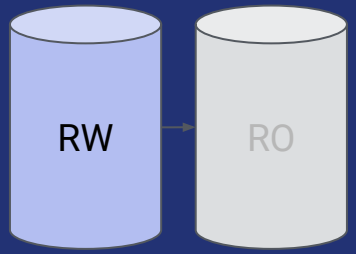
MySQL

Taiji

Writer	Instances

Hostgroup 0	Hostgroup 1

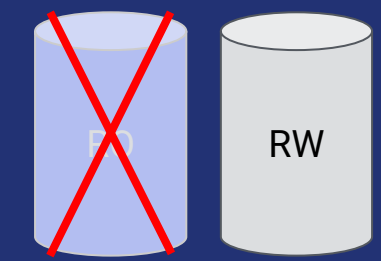
shopify



Writer	Instances
shard123-db1	**shard123-db1**
	shard123-db2

Hostgroup 0	Hostgroup 1
shard123-db1	**shard123-db2**

shopify



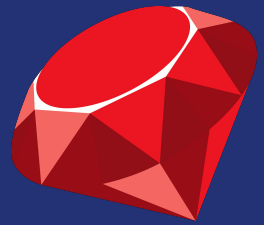
Writer	Instances
shard123-db2	**shard123-db2**

Hostgroup 0	Hostgroup 1
shard123-db2	**shard123-db2**

shopify

orchestrator

<> with ♥ by GitHub



kubernetes

MySQL™

Google Cloud Platform

shopify

Questions?

shopify