Organize the migration of a hundred databases to the cloud

Percona Live ONLINE
May 12th 2021

PERCONA
LIVEONLINE

Meet

Maxime Fouilleul

Engineering Manager for DBRE

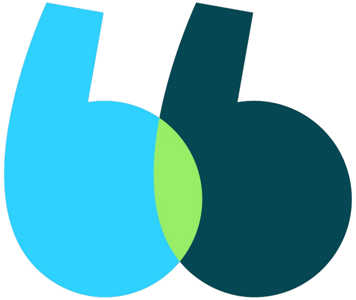
“Make the database not a problem.”

BUILD

> “ Package and support the database catalog for BlaBlaCar application services.

ADVISE

> “ Provide expertise in software engineering to help teams choose the right database for them and to ensure they use it the right way.



BlaBlaCar

The go-to marketplace for shared road travel

BlaBlaCar is a community-based marketplace allowing members to book seats in individual cars and buses alike. From carpool to buses, we have one common moto: #ZeroEmptySeats.

	Carpool	Bus Marketplace	BBC Branded Buses
Geographies	Global (22 countries)	Russia, Ukraine, Poland, Brazil	France, Germany
Position	Leader in all our markets	Leader in Eastern Europe Early stage in Brazil	Leader in France

BlaBlaCar

“The go-to marketplace for shared travel”

90 million members

25 million travelers per quarter

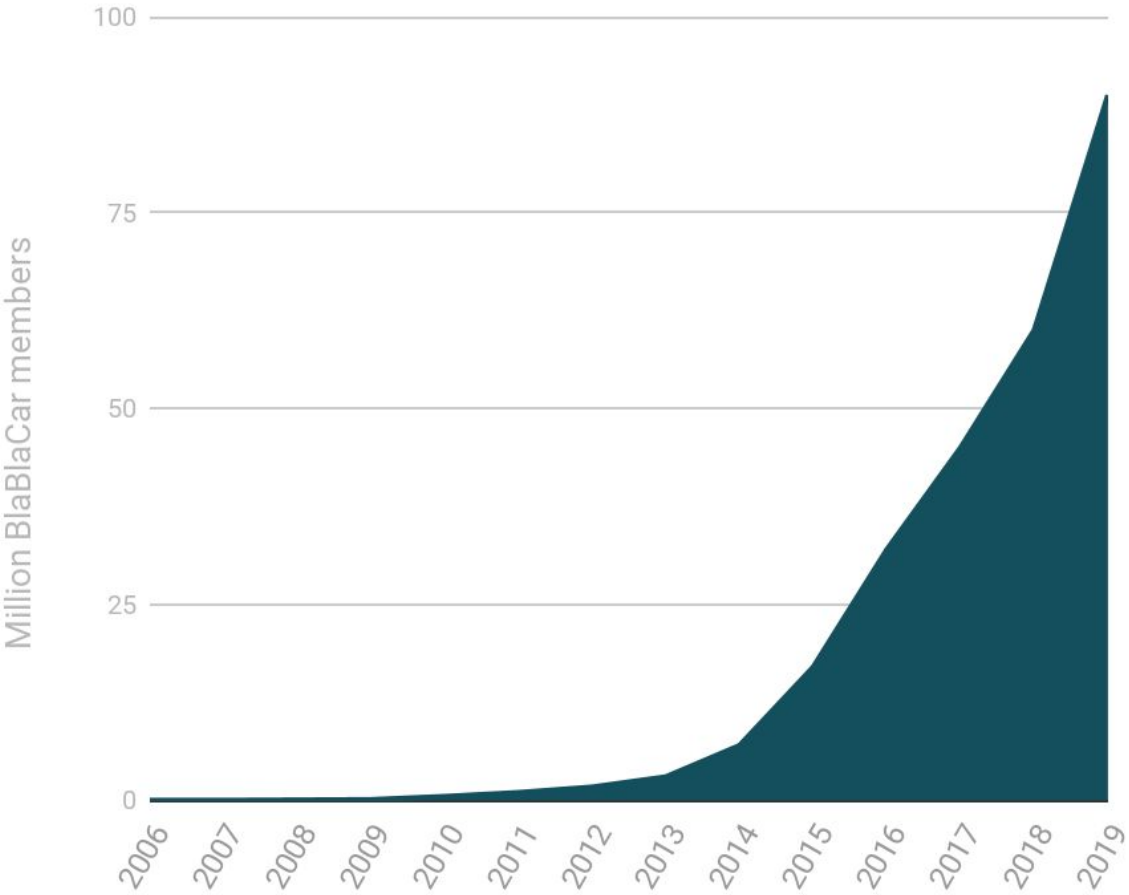
22 countries

BlaBlaCar

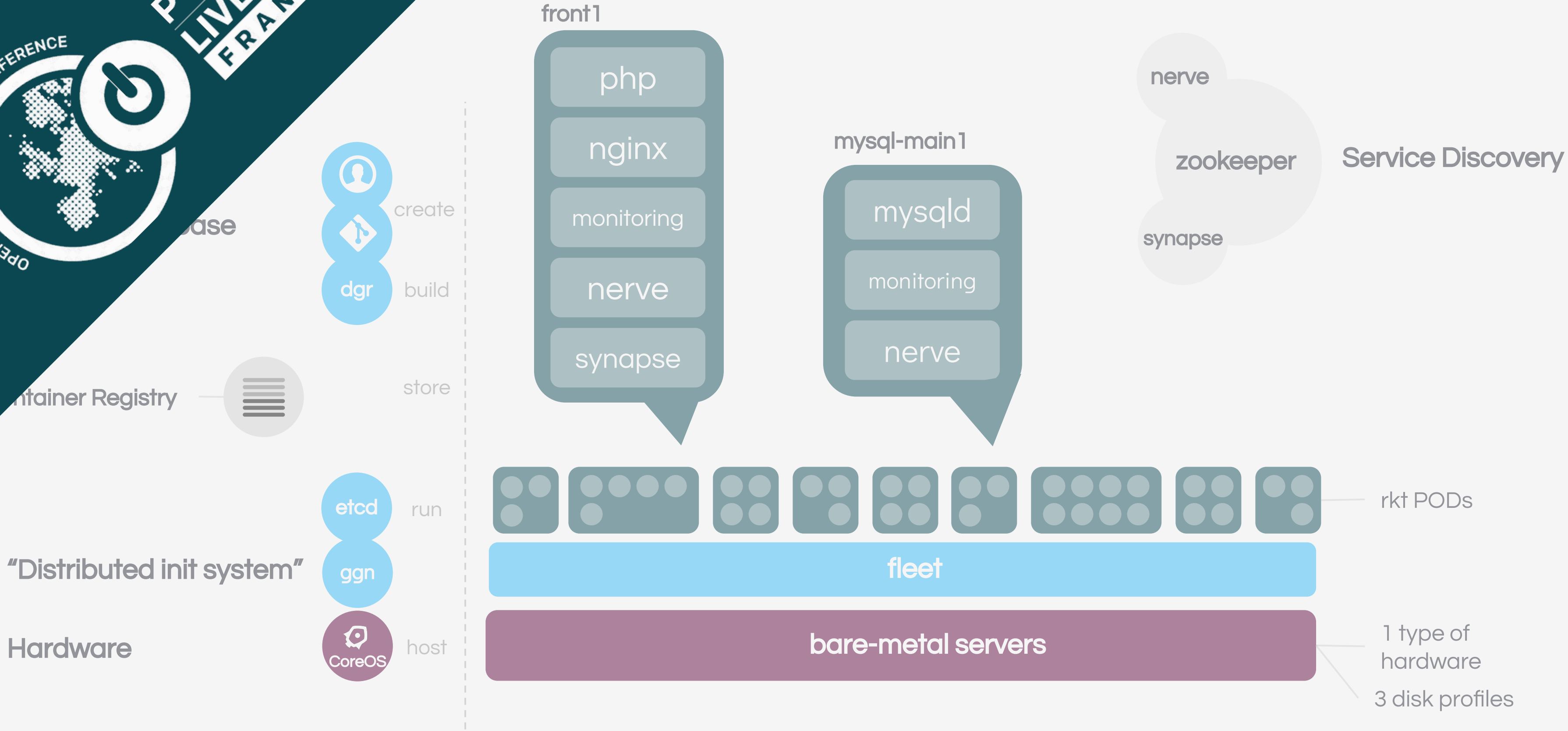
BlaBlaCar Bus

BlaBlaCar Daily

90 million members



100% Containers Powered Carpooling



Google Cloud Platform

Our production database infra in 2019

MySQL
30 Production Clusters

Cassandra
7 Production Clusters

RabbitMQ
13 Production Clusters

Redis
19 Production Clusters

Elasticsearch
6 Production Clusters

PostgreSQL
5 Production Clusters

Kafka
8 Production Clusters

Couchbase
5 Production Clusters

The mission

2019

sign cloud provider

end of 2020

close on-premise

Consolidate the DBRE team

staffing plan includes 4 SRE database enthusiastic

Migrate 100+ databases

Package reliable systems, accompany the migration and decommissioning

The dream team

Database Reliability Engineering (DBRE)

Engineering Manager

Product Owner

SRE

Distributed Databases

SRE

Cloud & Kubernetes

SRE

Kafka

SRE

Databases

“Fly me to the cloud”

The DBRE vision

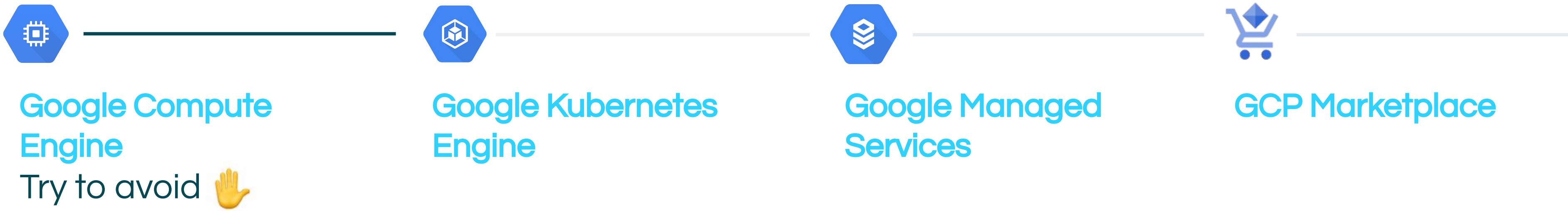
Google Compute Engine

Google Kubernetes Engine

Google Managed Services

GCP Marketplace

The DBRE vision



Google Compute Engine

Try to avoid ✋

Google Kubernetes Engine

Google Managed Services

GCP Marketplace

The DBRE vision

Google Compute Engine

Try to avoid ✋

Google Kubernetes Engine

Prefer 👌

Google Managed Services

GCP Marketplace

The DBRE vision

Google Compute Engine
Try to avoid ✋

Google Kubernetes Engine
Prefer 👌

Google Managed Services
Do 🤏

GCP Marketplace

The DBRE vision

Google Compute Engine
Try to avoid ✋

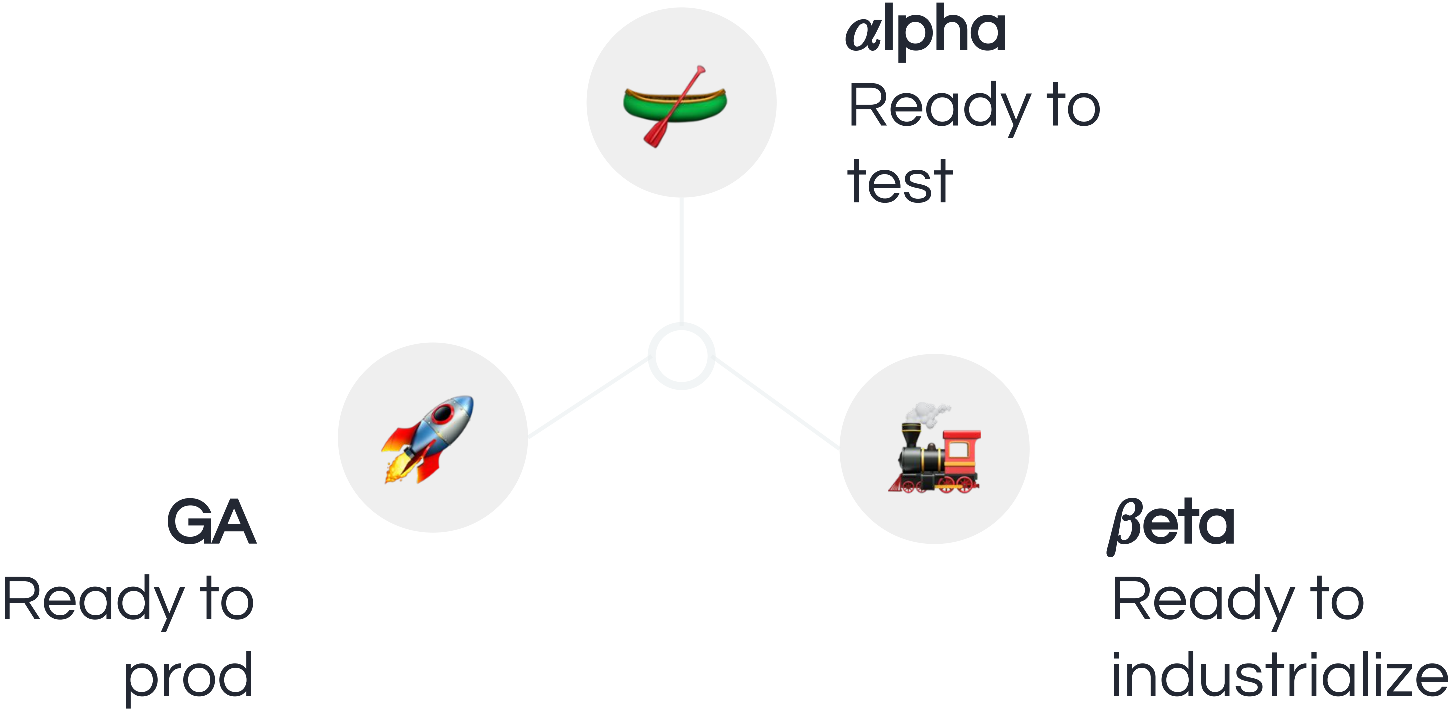
Google Kubernetes Engine
Prefer 👌

Google Managed Services
Do 🤏

GCP Marketplace
Don’t ✋

“Be transparent to ensure buy-in”

Be clear on iterations

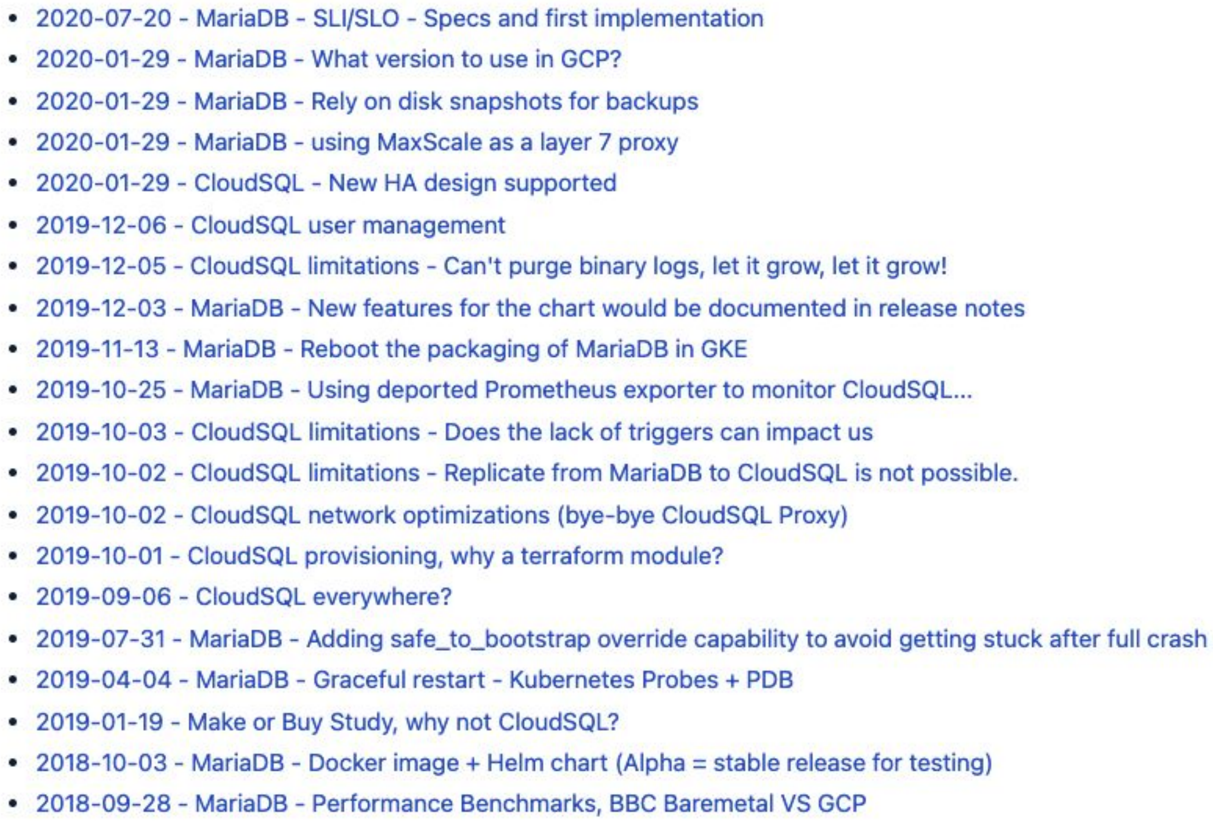


αlpha
Ready to test

βeta
Ready to industrialize

GA
Ready to prod

Documentation as log



- 2020-07-20 - MariaDB - SLI/SLO - Specs and first implementation
- 2020-01-29 - MariaDB - What version to use in GCP?
- 2020-01-29 - MariaDB - Rely on disk snapshots for backups
- 2020-01-29 - MariaDB - using MaxScale as a layer 7 proxy
- 2020-01-29 - CloudSQL - New HA design supported
- 2019-12-06 - CloudSQL user management
- 2019-12-05 - CloudSQL limitations - Can't purge binary logs, let it grow, let it grow!
- 2019-12-03 - MariaDB - New features for the chart would be documented in release notes
- 2019-11-13 - MariaDB - Reboot the packaging of MariaDB in GKE
- 2019-10-25 - MariaDB - Using deported Prometheus exporter to monitor CloudSQL...
- 2019-10-03 - CloudSQL limitations - Does the lack of triggers can impact us
- 2019-10-02 - CloudSQL limitations - Replicate from MariaDB to CloudSQL is not possible.
- 2019-10-02 - CloudSQL network optimizations (bye-bye CloudSQL Proxy)
- 2019-10-01 - CloudSQL provisioning, why a terraform module?
- 2019-09-06 - CloudSQL everywhere?
- 2019-07-31 - MariaDB - Adding safe_to_bootstrap override capability to avoid getting stuck after full crash
- 2019-04-04 - MariaDB - Graceful restart - Kubernetes Probes + PDB
- 2019-01-19 - Make or Buy Study, why not CloudSQL?
- 2018-10-03 - MariaDB - Docker image + Helm chart (Alpha = stable release for testing)
- 2018-09-28 - MariaDB - Performance Benchmarks, BBC Baremetal VS GCP

Runbooks

How do I? 🧰

🔥 What to do when?

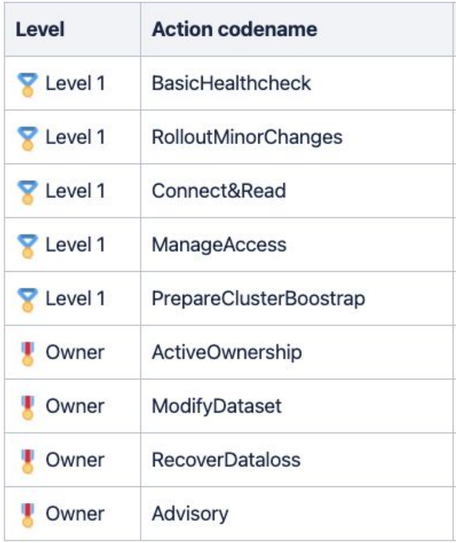
Gamify the knowledge sharing process

Level 1

Actions are basic tasks that should be mastered by each team member.

Level "Owner"

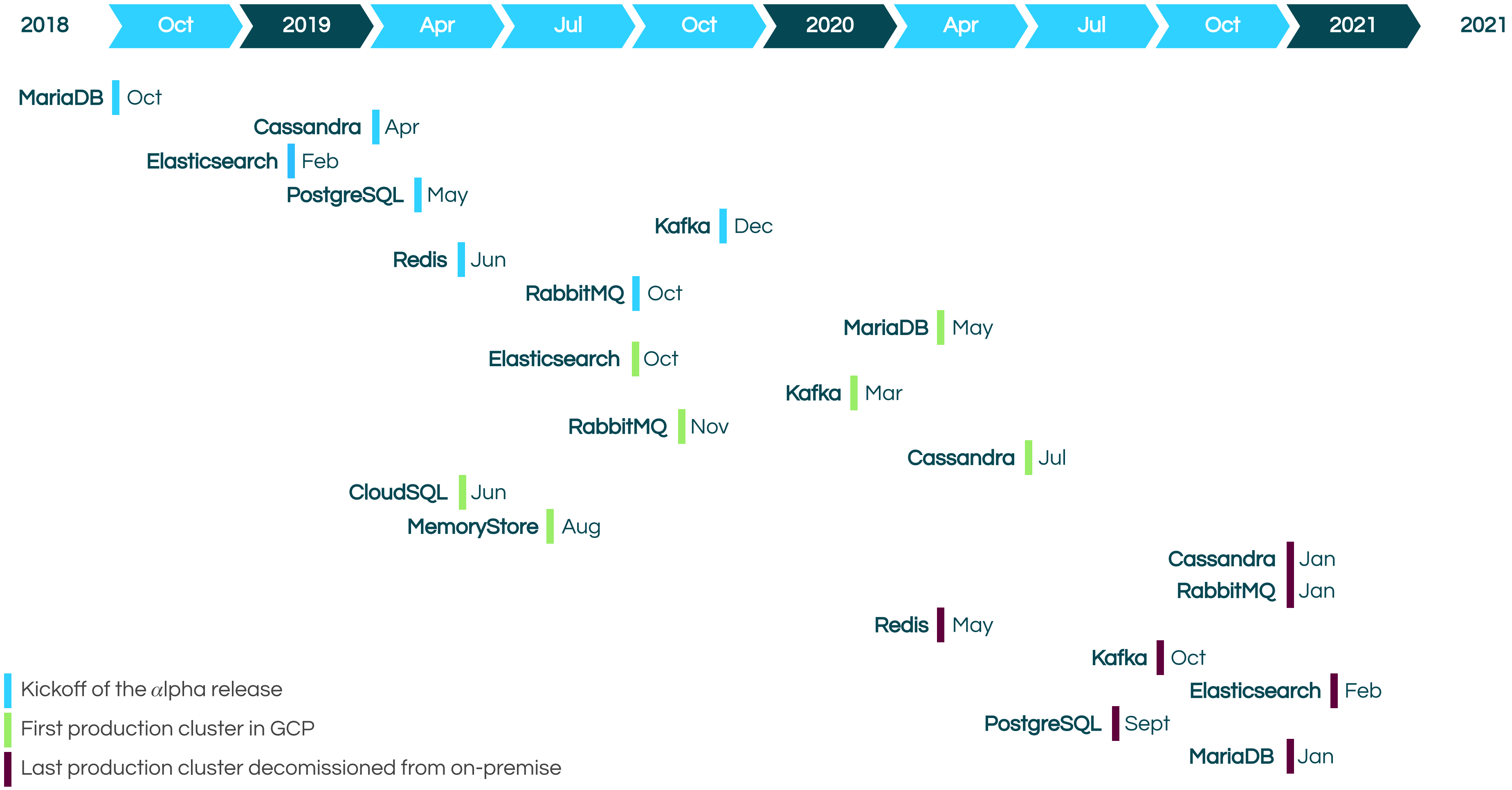
Actions allow the component to be actively supported, they should be mastered by at least 2 members.



Level	Action codename
🥇 Level 1	BasicHealthcheck
🥇 Level 1	RolloutMinorChanges
🥇 Level 1	Connect&Read
🥇 Level 1	ManageAccess
🥇 Level 1	PrepareClusterBoostrap
🎖️ Owner	ActiveOwnership
🎖️ Owner	ModifyDataset
🎖️ Owner	RecoverDataloss
🎖️ Owner	Advisory

Implementations

and migration paths



2018 Oct 2019 Apr Jul Oct 2020 Apr Jul Oct 2021 2021

MariaDB Oct

Cassandra Apr

Elasticsearch Feb

PostgreSQL May

Kafka Dec

Redis Jun

RabbitMQ Oct

MariaDB May

Elasticsearch Oct

Kafka Mar

RabbitMQ Nov

Cassandra Jul

CloudSQL Jun

MemoryStore Aug

Cassandra Jan

RabbitMQ Jan

Redis May

Kafka Oct

Elasticsearch Feb

PostgreSQL Sept

MariaDB Jan

🟦 Kickoff of the αlpha release

🟩 First production cluster in GCP

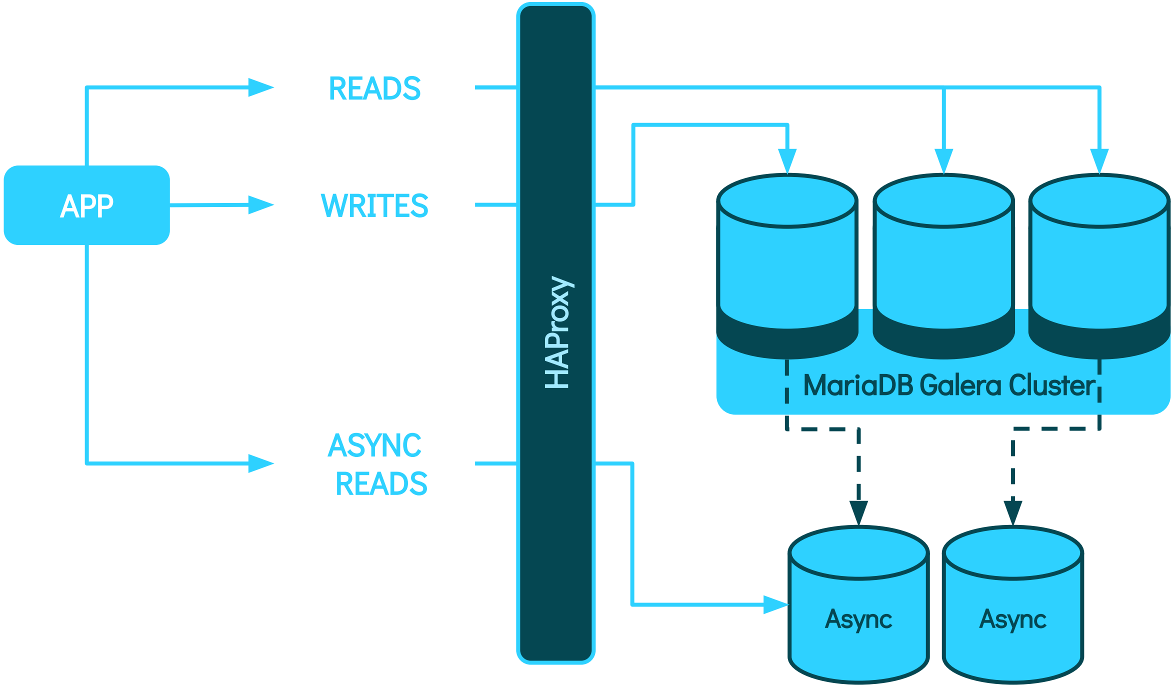
🟪 Last production cluster decomissioned from on-premise

Key elements for the implementations

- Leverage **Kubernetes Statefulset**
- master **affinity** in the PodSpec
- prefer **Google Persistent disks** over Local SSD
- leverage Persistent disk **Snapshots**
- promote **distributed ownership**
- use **Terraform** for Google managed services

MySQL

A production MySQL service in 2019



Gather requierements

- Will your database be migrated or abandoned?
- What is the tolerated downtime for the migration?
- Can we migrate the reads separately from the writes?

Google CloudSQL

1. The application can tolerate several minutes of downtime
2. Writes can be stopped during the dataset migration
3. The need is fairly lightweight

MariaDB in Kubernetes

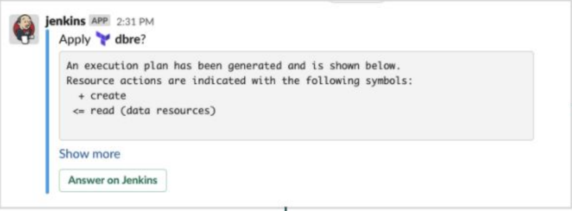
What DBRE is packaging for CloudSQL?

Terraform Module
To ease and standardize the usages

Kubernetes Operator
To setup a Prometheus exporter

What DBRE is packaging for CloudSQL?

```
$ git commit -m"[preprod-1][dbre] Create cloudsql-test"
$ git push origin preprod-dbre-cloudsql-test
```

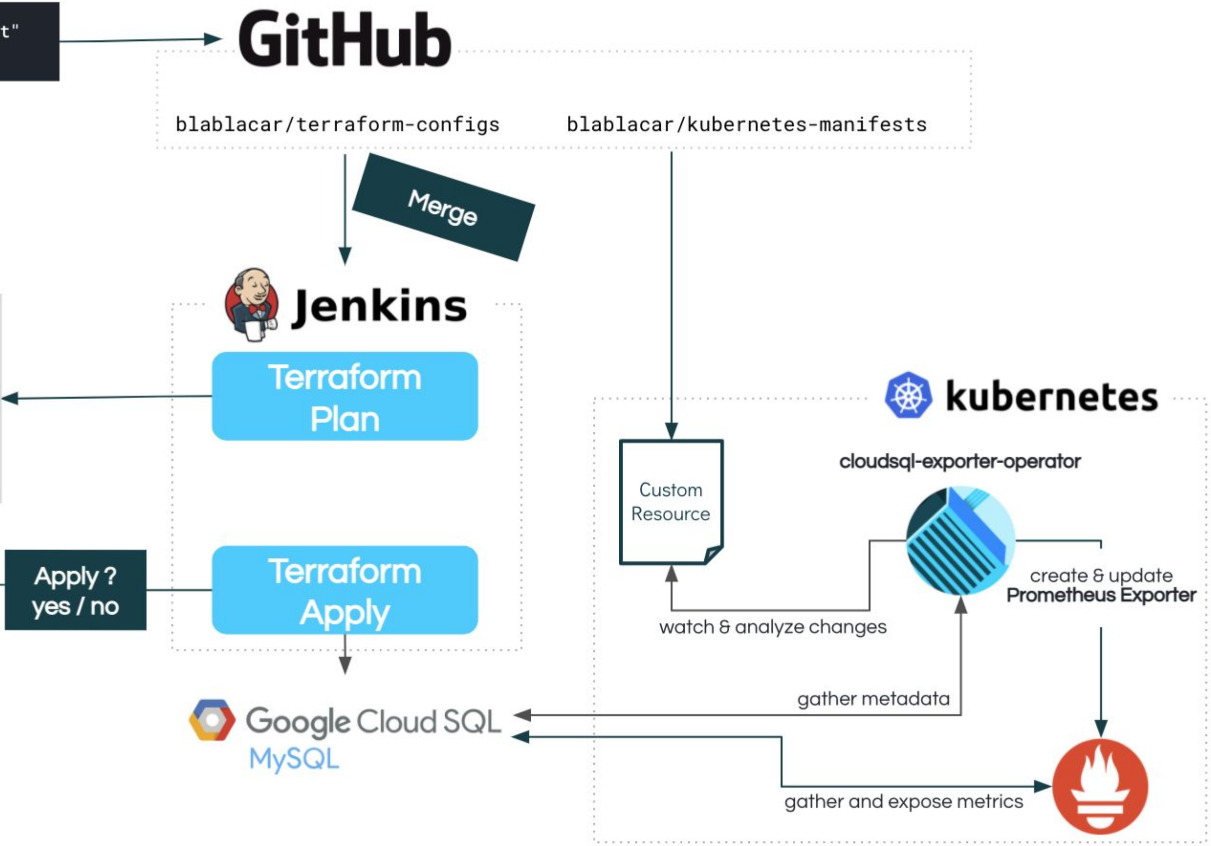


jenkins APP 2:31 PM

Apply dbre?

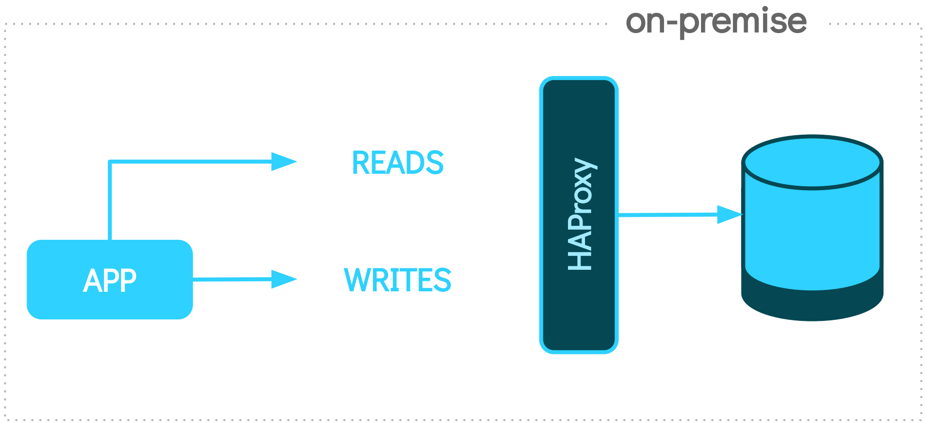
```
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
  + create
 <= read (data resources)
```

Show more



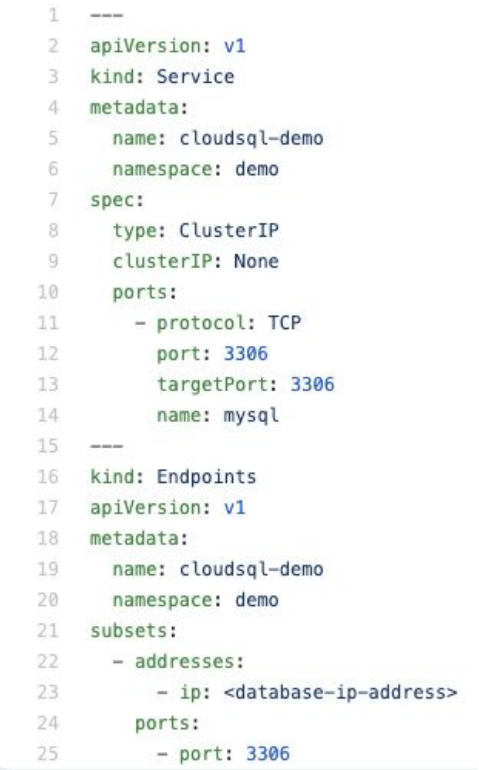
CloudSQL Migration Path

Initial stage

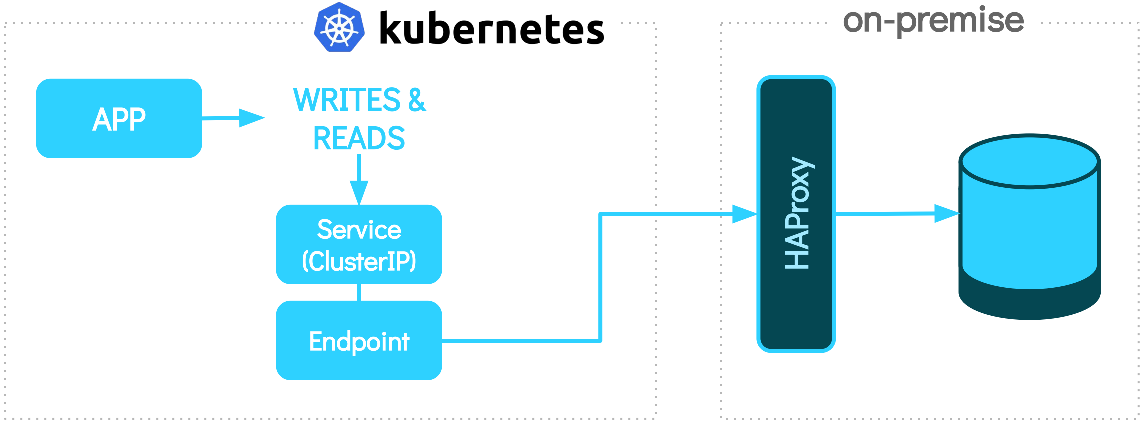


CloudSQL Migration Path

Move the application

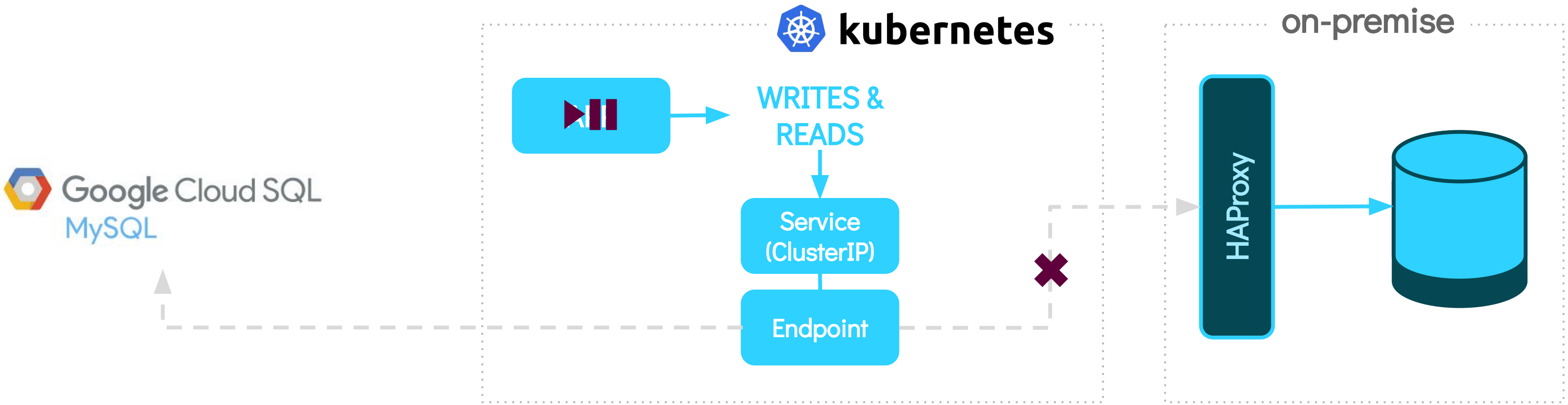


```
---
apiVersion: v1
kind: Service
metadata:
  name: cloudsql-demo
  namespace: demo
spec:
  type: ClusterIP
  clusterIP: None
  ports:
    - protocol: TCP
      port: 3306
      targetPort: 3306
      name: mysql
---
kind: Endpoints
apiVersion: v1
metadata:
  name: cloudsql-demo
  namespace: demo
subsets:
  - addresses:
      - ip: <database-ip-address>
    ports:
      - port: 3306
```



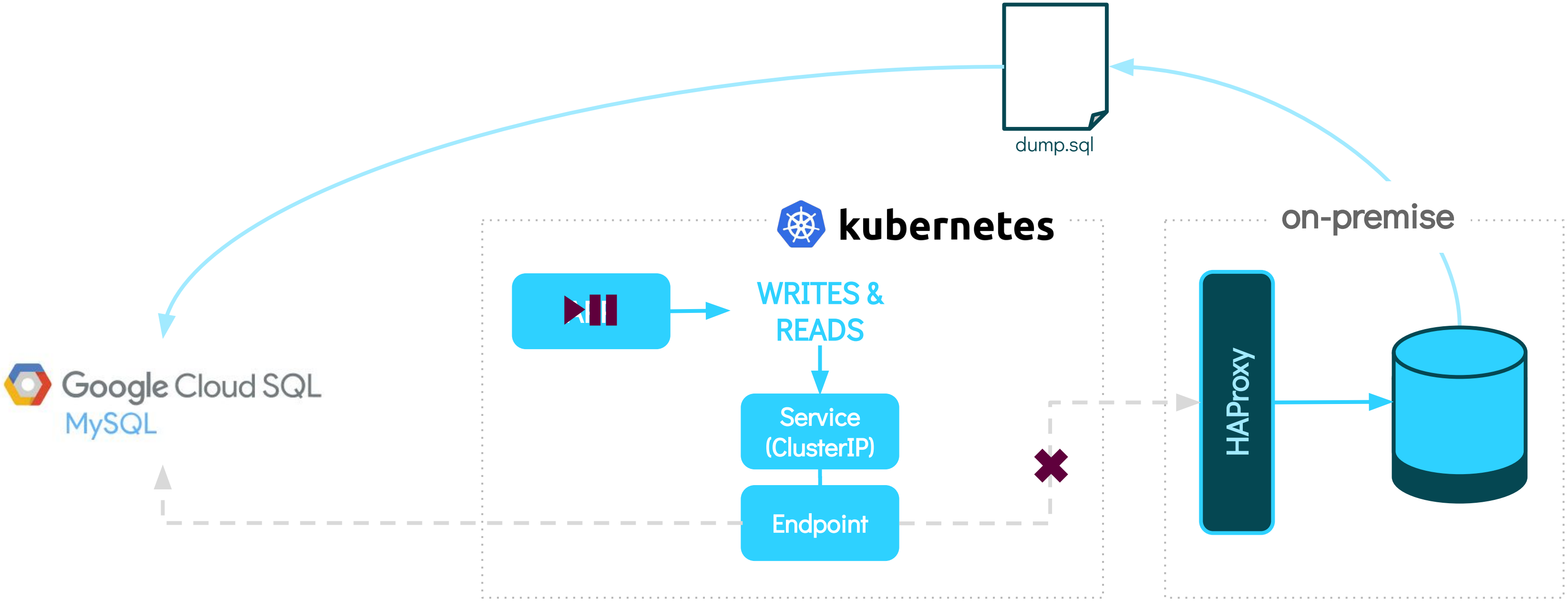
CloudSQL Migration Path

Stop the application and switch endpoint



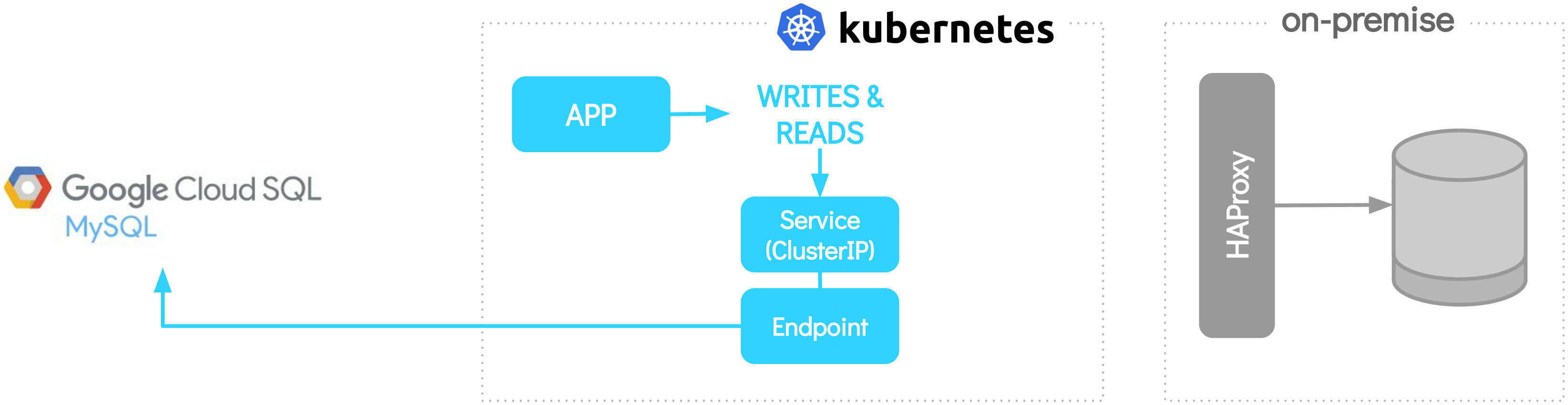
CloudSQL Migration Path

Copy dataset



CloudSQL Migration Path

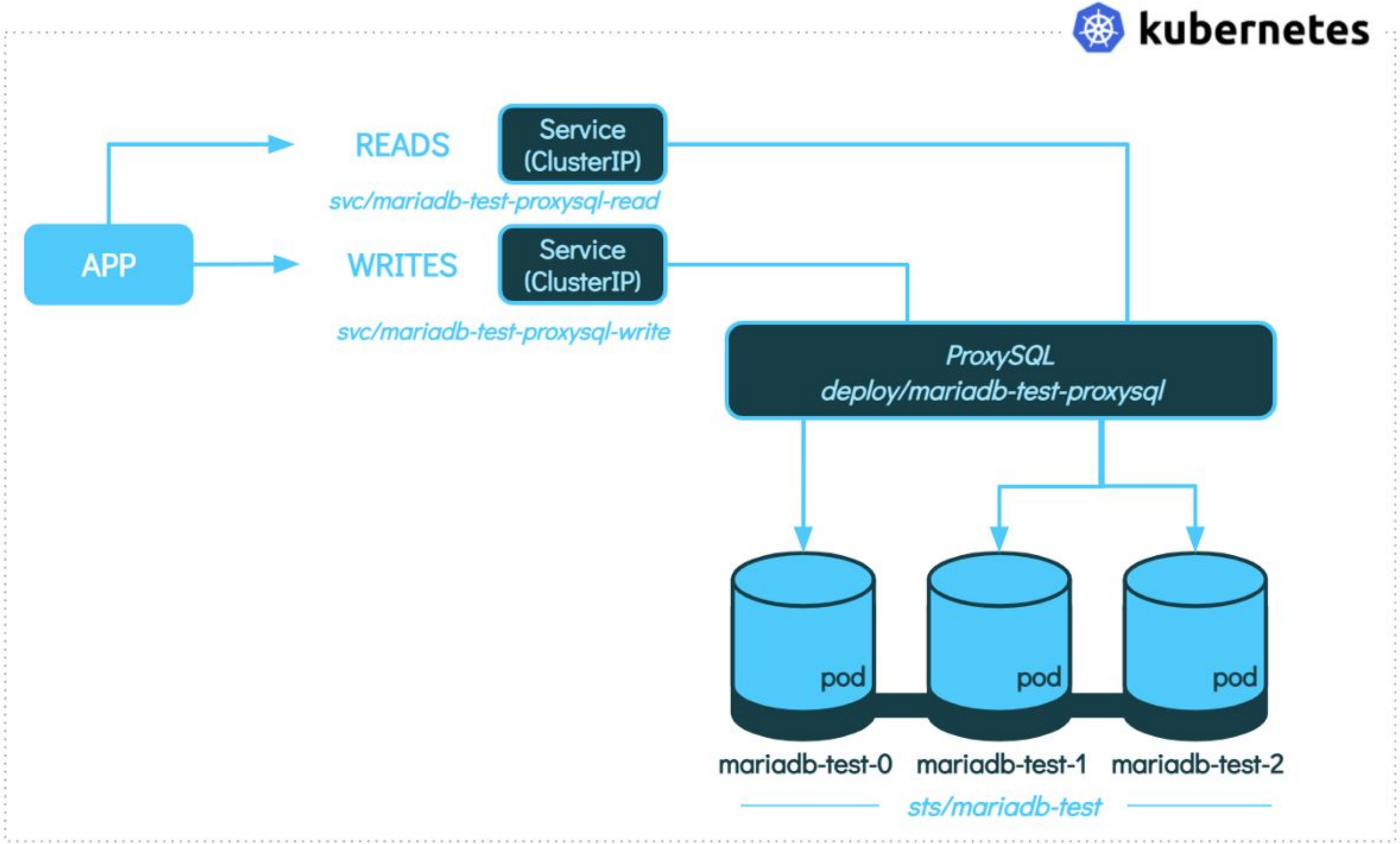
Restart the application



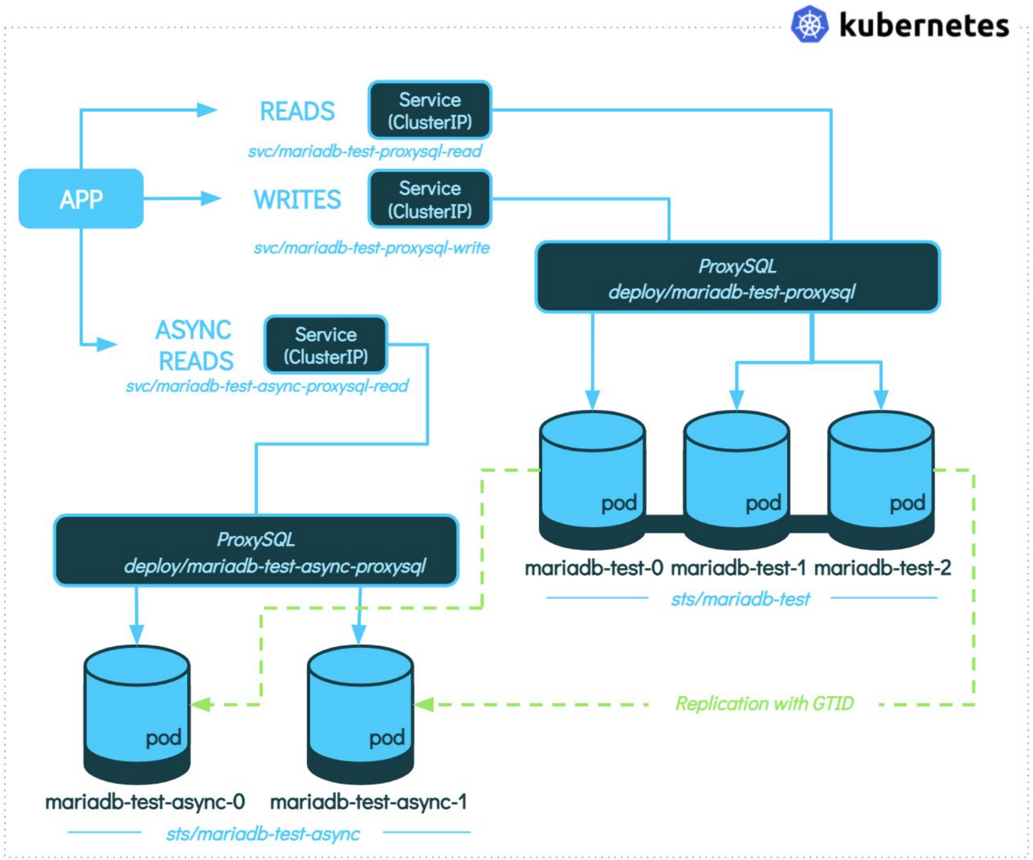
MariaDB in-house packaging via a Helm Chart

- A StatefulSet with Galera enabled...or not
- A Deployment running ProxySQL
- Prometheus exporter sidecars to export metrics
- A bunch of Jobs that manipulate disk snapshots
- A Deployment running an SLI Prober
- Services, RBAC, and PDB...

A production MySQL service in 2021



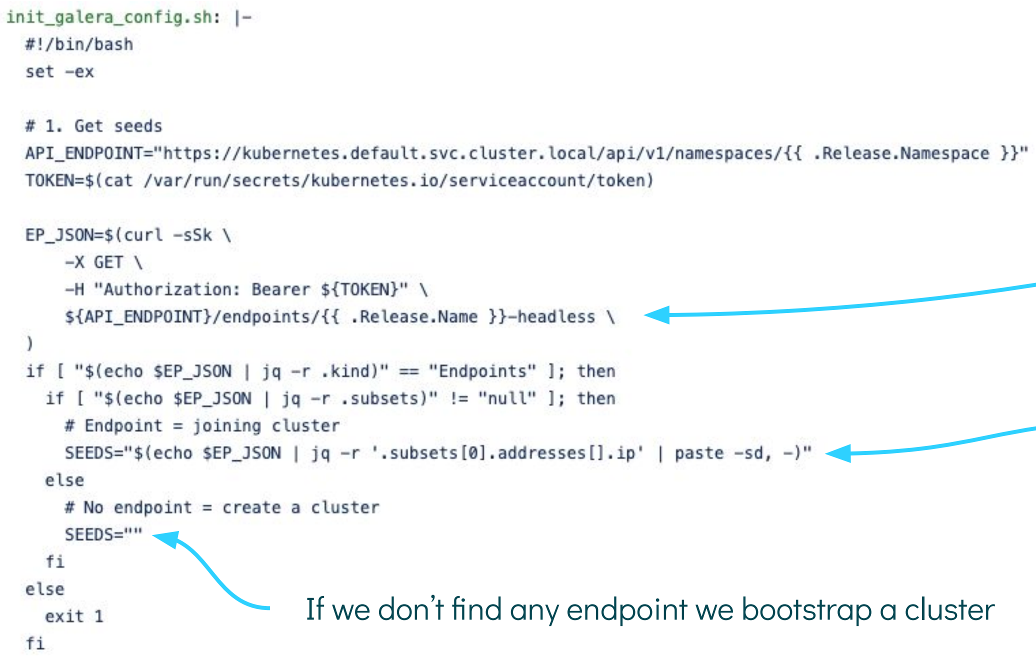
With asynchronous replicas



MariaDB packaging tips

Dynamically find Galera seeds

wsrep_cluster_address



```
init_galera_config.sh: |-
  #!/bin/bash
  set -ex

  # 1. Get seeds
  API_ENDPOINT="https://kubernetes.default.svc.cluster.local/api/v1/namespaces/{{ .Release.Namespace }}"
  TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token)

  EP_JSON=$(curl -sSk \
      -X GET \
      -H "Authorization: Bearer ${TOKEN}" \
      ${API_ENDPOINT}/endpoints/{{ .Release.Name }}-headless \
  )
  if [ "$(echo $EP_JSON | jq -r .kind)" == "Endpoints" ]; then
    if [ "$(echo $EP_JSON | jq -r .subsets)" != "null" ]; then
      # Endpoint = joining cluster
      SEEDS="$(echo $EP_JSON | jq -r '.subsets[0].addresses[].ip' | paste -sd, -)"
    else
      # No endpoint = create a cluster
      SEEDS=""
    fi
  else
    exit 1
  fi
```

We use a Kubernetes **Headless Service** to get available (ready) endpoints

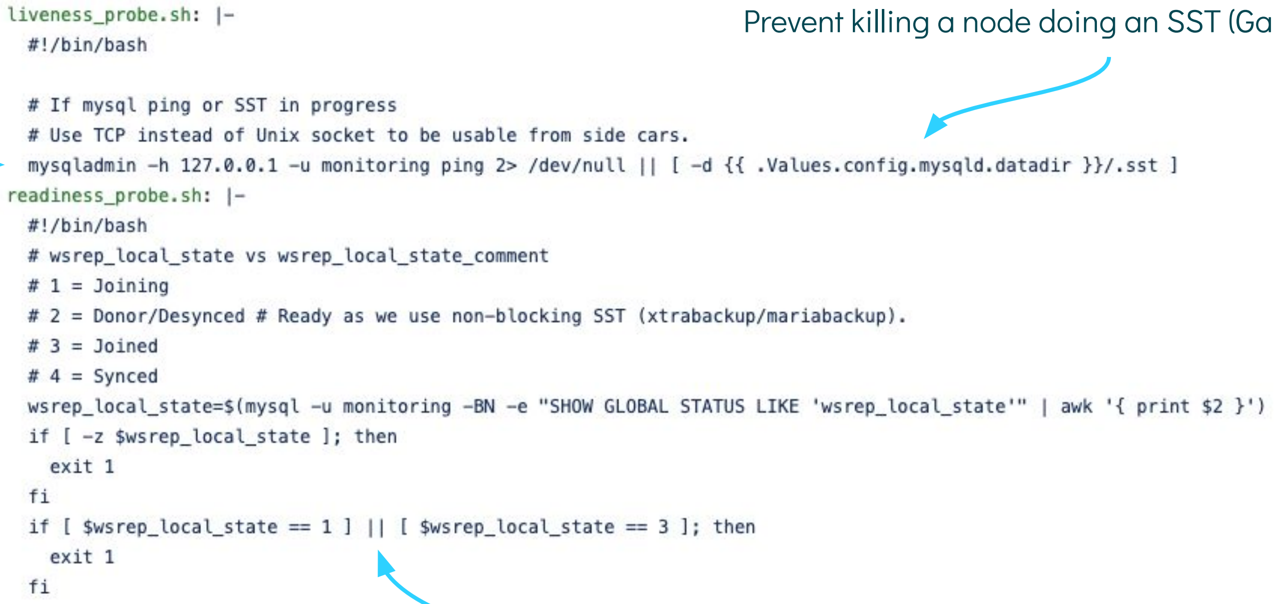
If we find endpoints we join a cluster

If we don’t find any endpoint we bootstrap a cluster

Having accurate **Liveness** and **Readiness**

Simple ping to report the MySQL is live or not

Prevent killing a node doing an SST (Galera full resync)



```
liveness_probe.sh: |-
  #!/bin/bash

  # If mysql ping or SST in progress
  # Use TCP instead of Unix socket to be usable from side cars.
  mysqladmin -h 127.0.0.1 -u monitoring ping 2> /dev/null || [ -d {{ .Values.config.mysqld.datadir }}/.sst ]
readiness_probe.sh: |-
  #!/bin/bash
  # wsrep_local_state vs wsrep_local_state_comment
  # 1 = Joining
  # 2 = Donor/Desynced # Ready as we use non-blocking SST (xtrabackup/mariabackup).
  # 3 = Joined
  # 4 = Synced
  wsrep_local_state=$(mysql -u monitoring -BN -e "SHOW GLOBAL STATUS LIKE 'wsrep_local_state'" | awk '{ print $2 }')
  if [ -z $wsrep_local_state ]; then
    exit 1
  fi
  if [ $wsrep_local_state == 1 ] || [ $wsrep_local_state == 3 ]; then
    exit 1
  fi
```

Only nodes **Synced** and **Donor** are considered “ready”

Having fun with the **Persistent disk Snapshots**

MariaDB

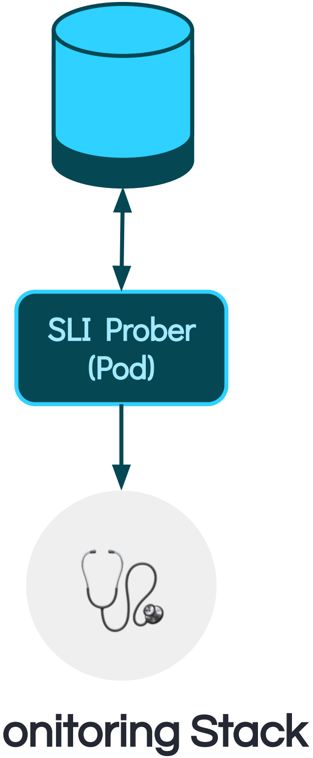
Snapshot Validator
To ensure we can restore backups

On-demand copy
Restore a backup in a minute

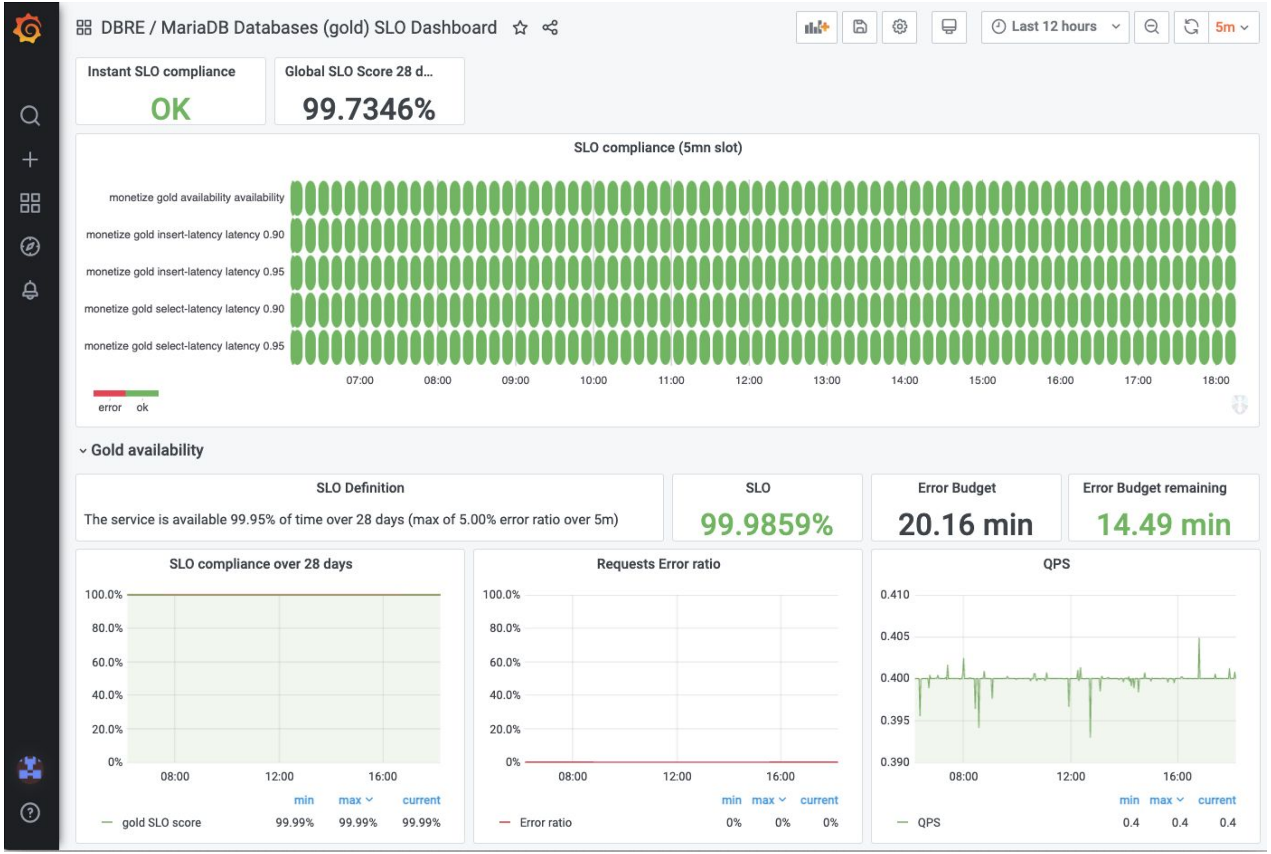
Daily Copy
To expose fresh dataset for BI Joes

Extend a Galera Clusters
Avoid SST

Having a SLI Prober to implement **SLO**


SLI Prober
(Pod)

Monitoring Stack



DBRE / MariaDB Databases (gold) SLO Dashboard

Instant SLO compliance

OK

Global SLO Score 28 d...

99.7346%

SLO compliance (5mn slot)

monetize gold availability availability
monetize gold insert-latency latency 0.90
monetize gold insert-latency latency 0.95
monetize gold select-latency latency 0.90
monetize gold select-latency latency 0.95
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
error
ok

Gold availability

SLO Definition

The service is available 99.95% of time over 28 days (max of 5.00% error ratio over 5m)

SLO

99.9859%

Error Budget

20.16 min

Error Budget remaining

14.49 min

SLO compliance over 28 days

100.0%
80.0%
60.0%
40.0%
20.0%
0%
08:00
12:00
16:00

	min	max	current
gold SLO score	99.99%	99.99%	99.99%

Requests Error ratio

	min	max	current
Error ratio	0%	0%	0%

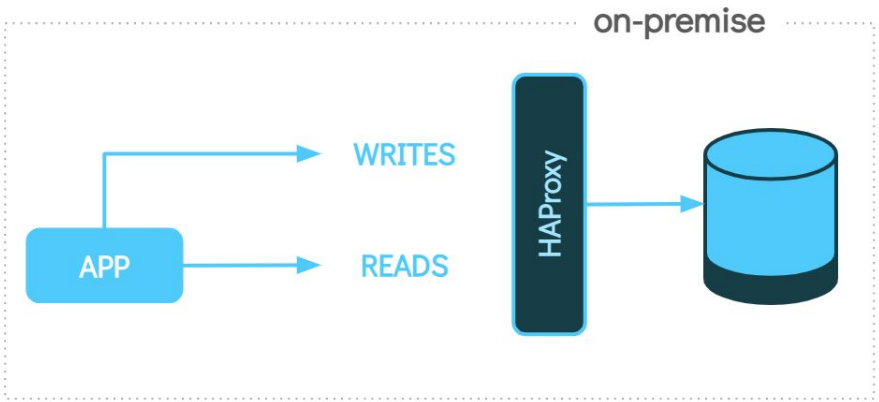
QPS

	min	max	current
QPS	0.4	0.4	0.4

MariaDB migration path

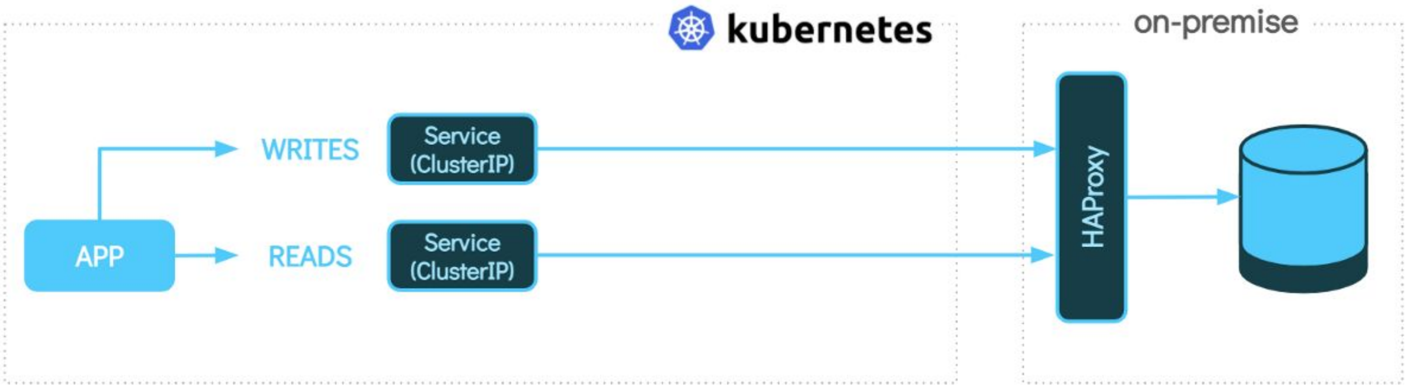
MariaDB Migration Path

Initial stage



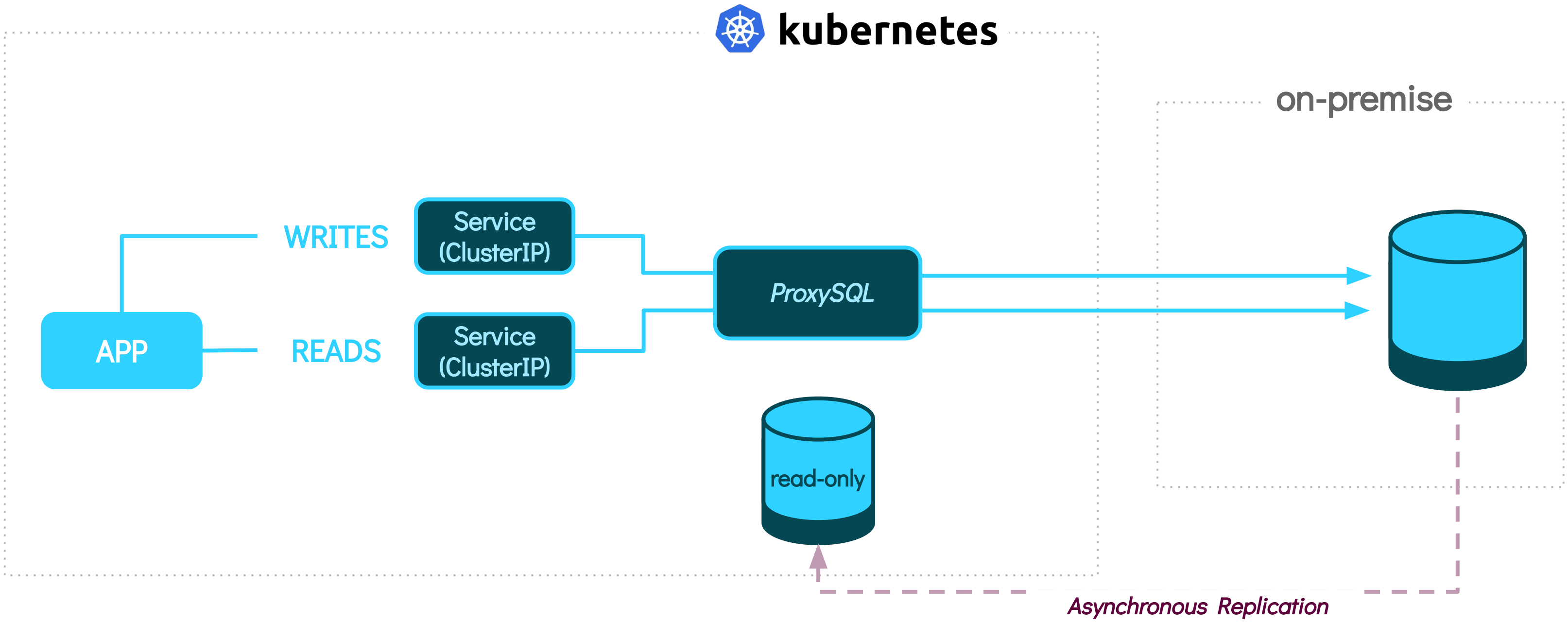
MariaDB Migration Path

Move the application



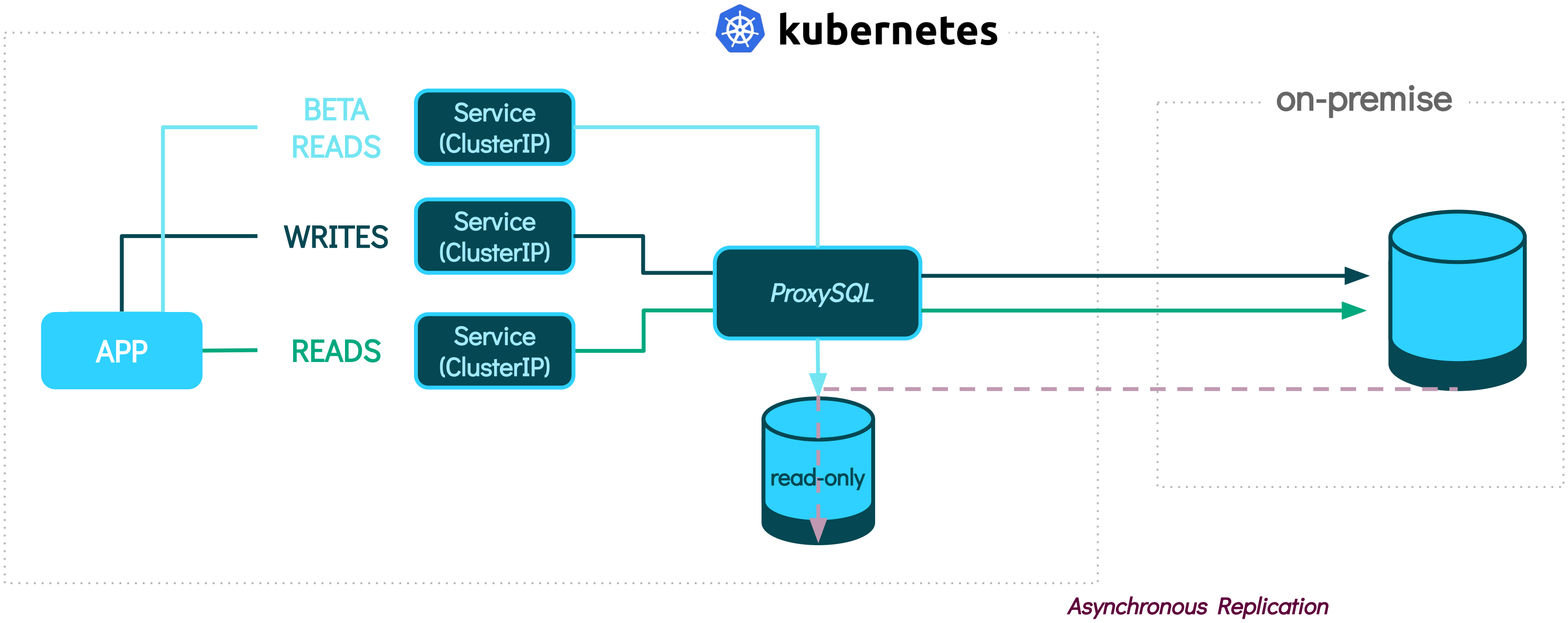
MariaDB Migration Path

Setup the database in GCP (with replication)



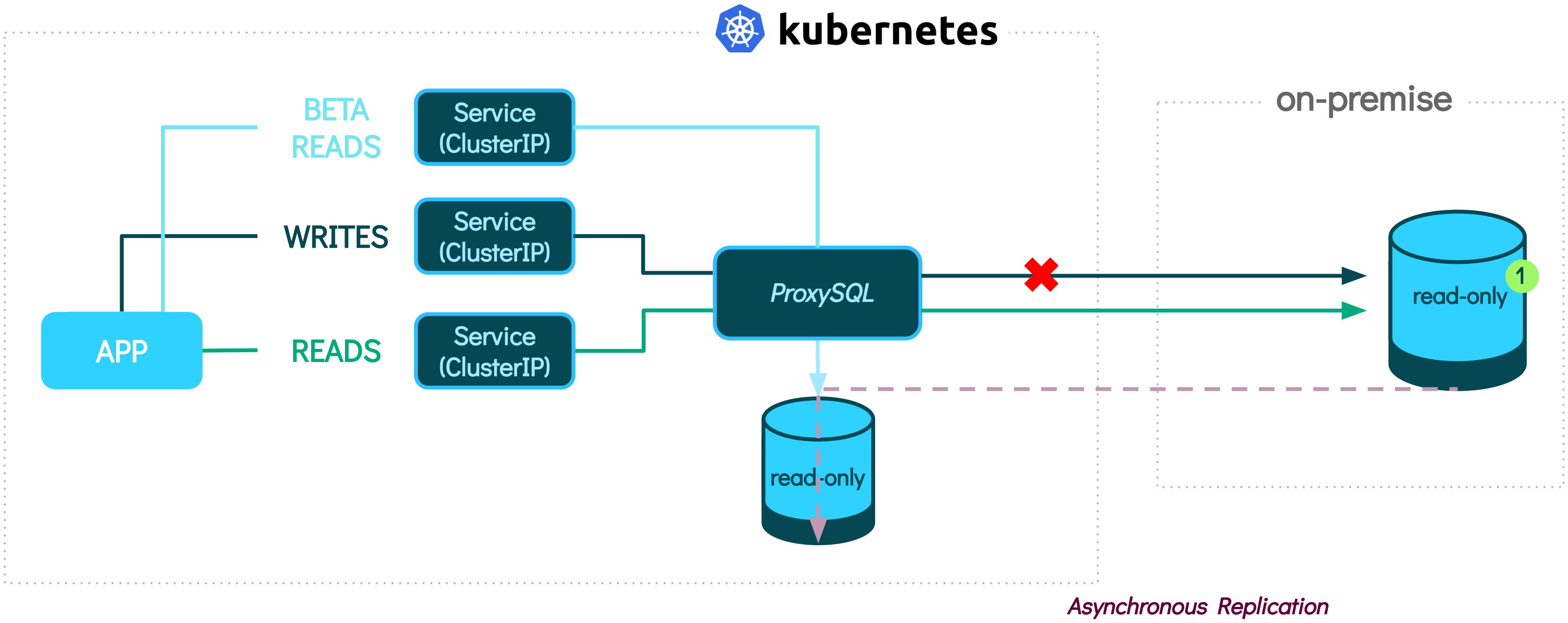
MariaDB Migration Path

Open a Beta endpoint for reads



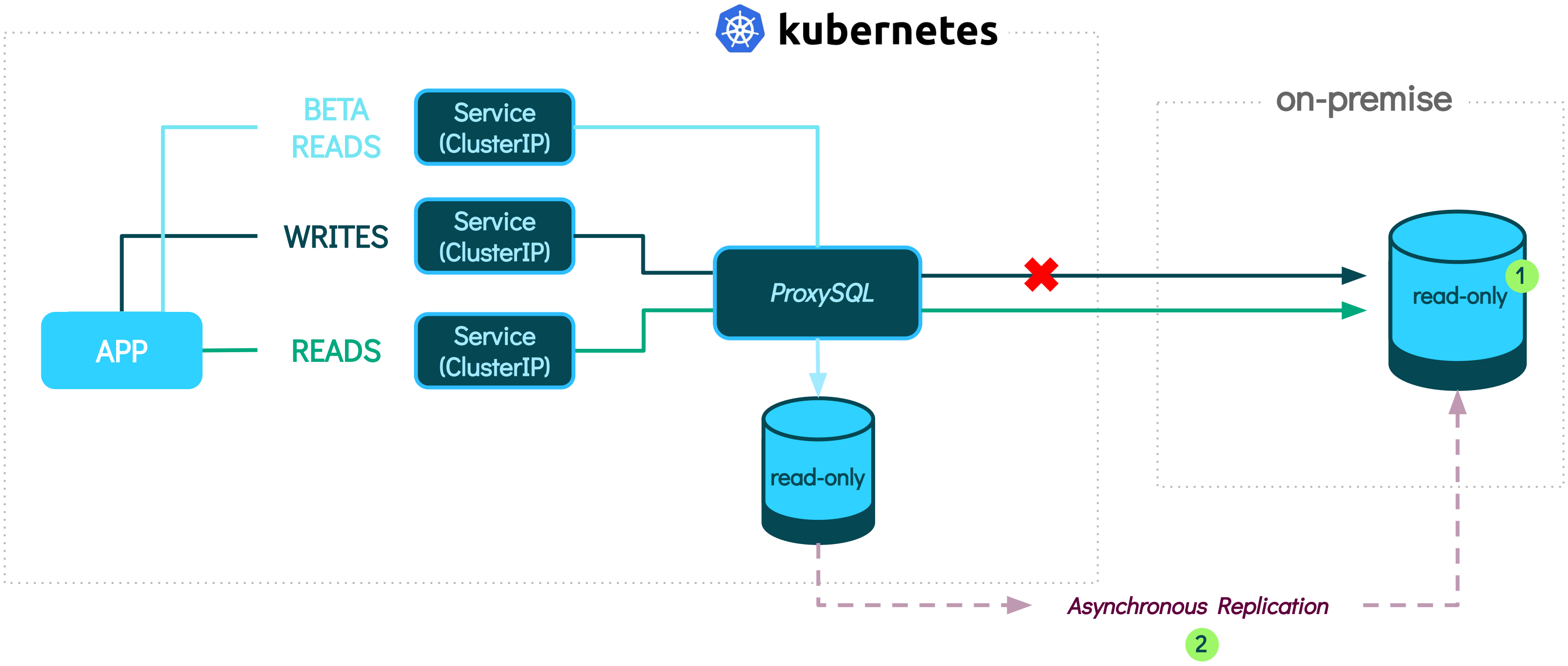
MariaDB Migration Path

D-Day: Set read-only



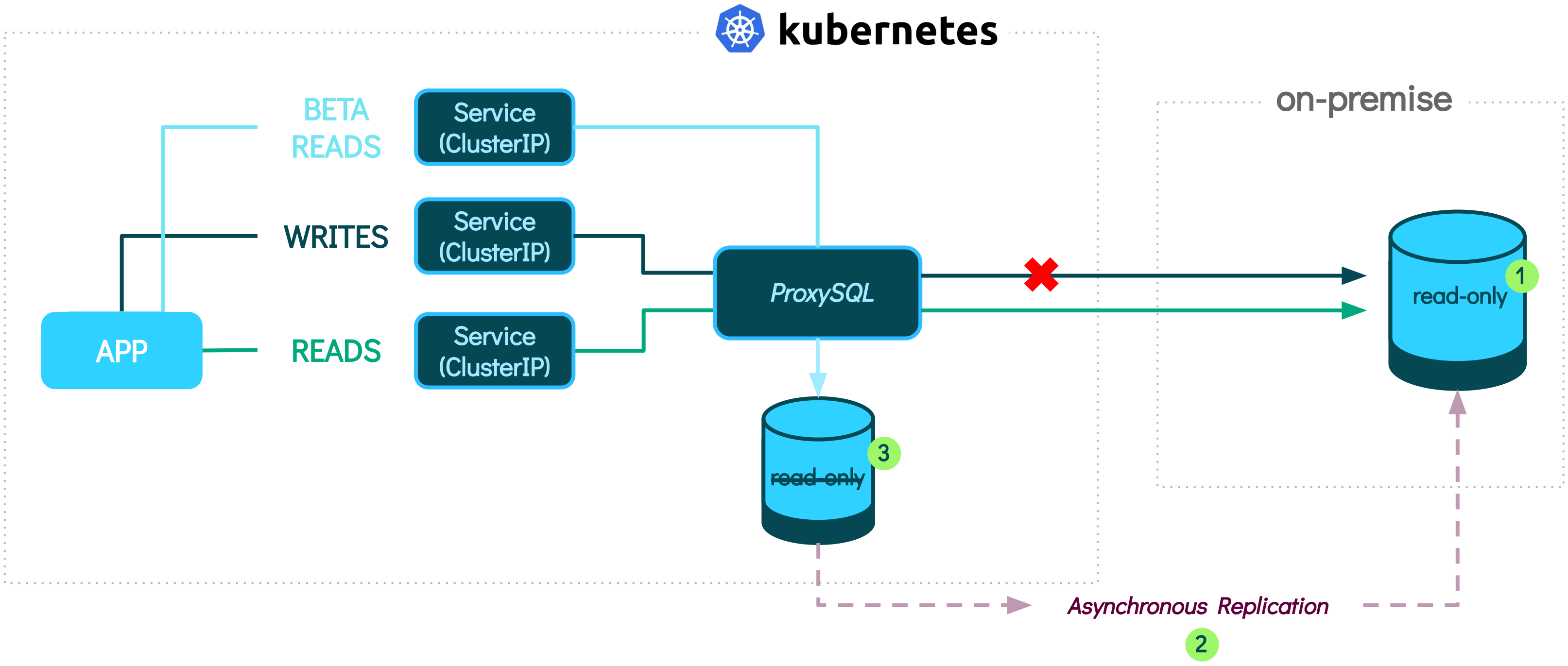
MariaDB Migration Path

D-Day: Reverse the replication stream



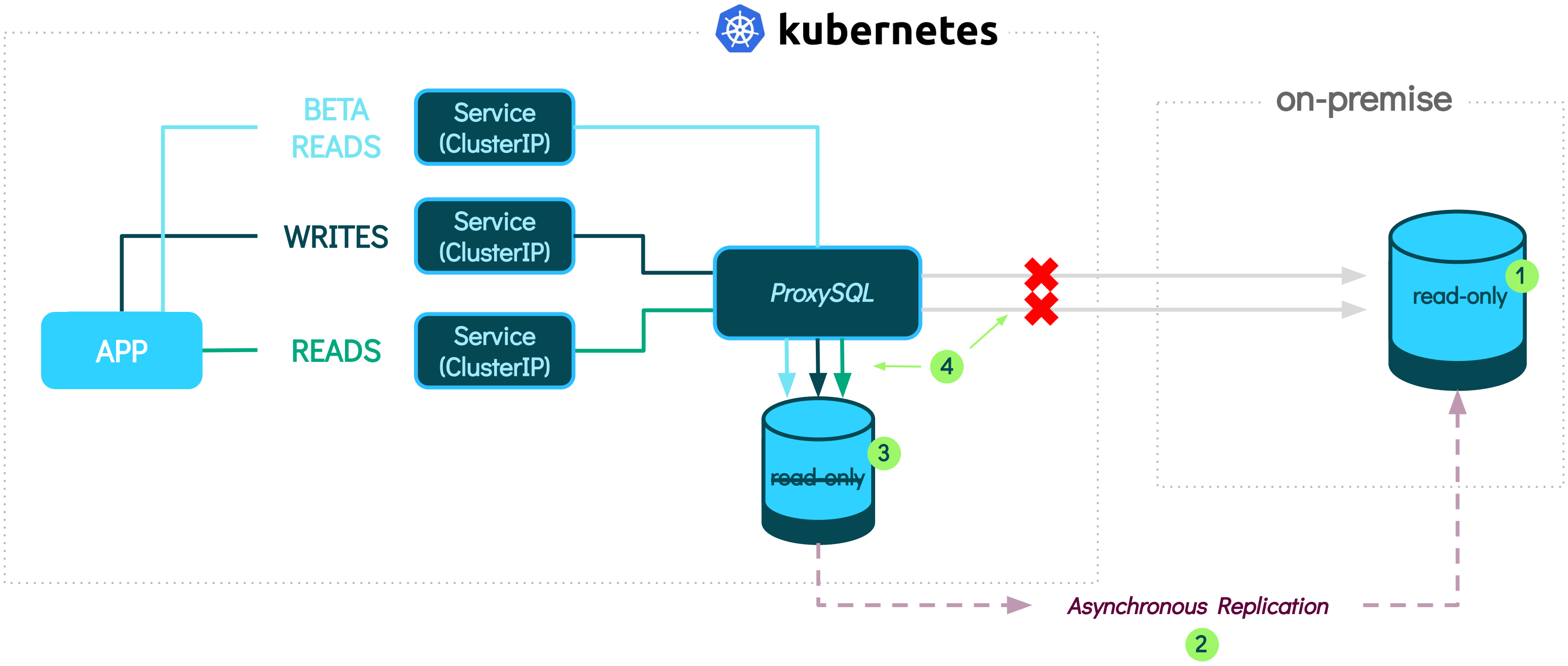
MariaDB Migration Path

D-Day: Enable writes in GCP



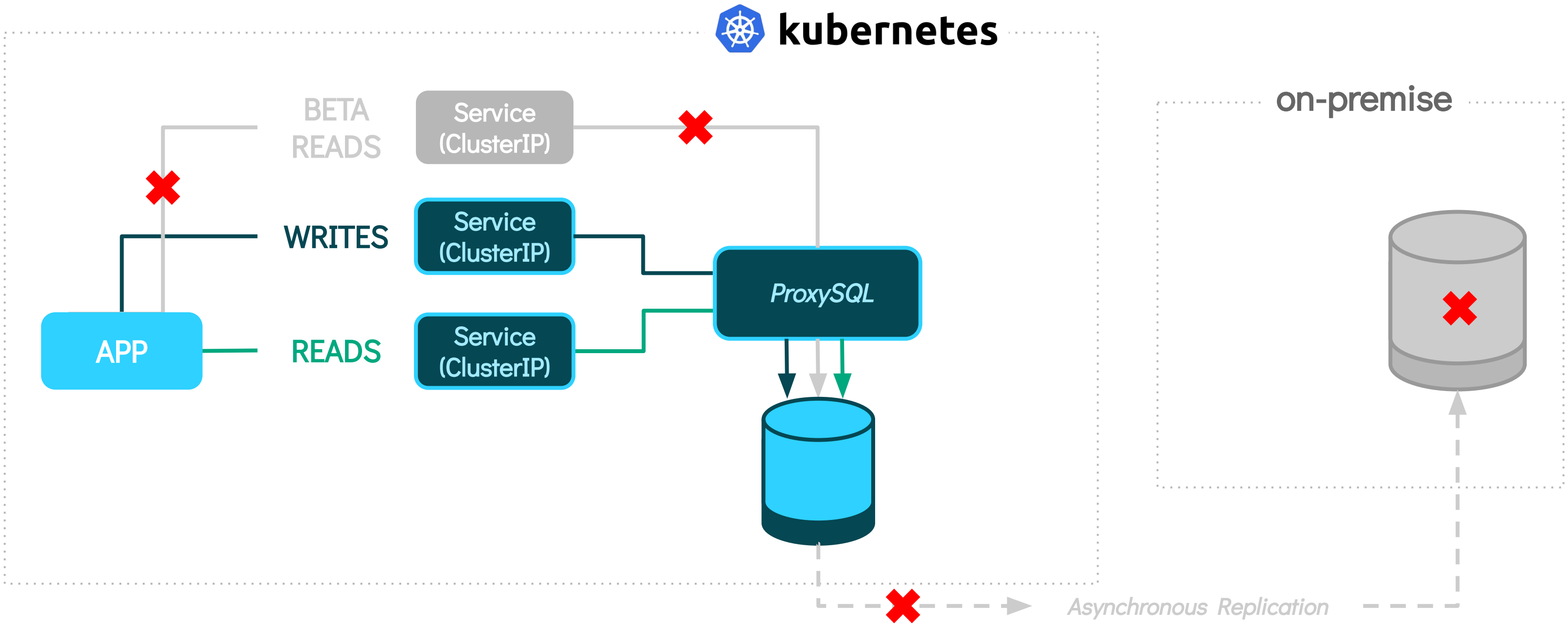
MariaDB Migration Path

D-Day: Change the endpoints



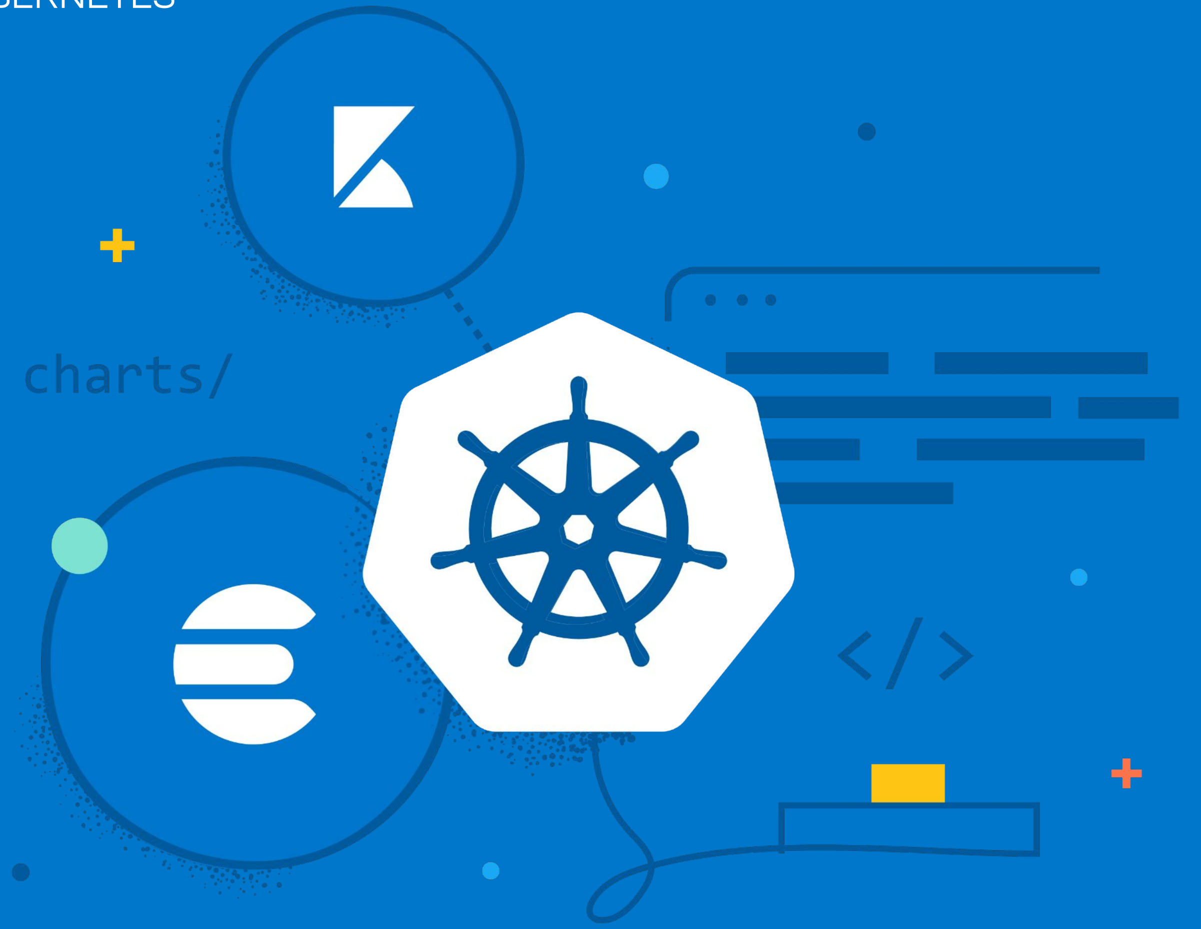
MariaDB Migration Path

Cleaning and decommissioning



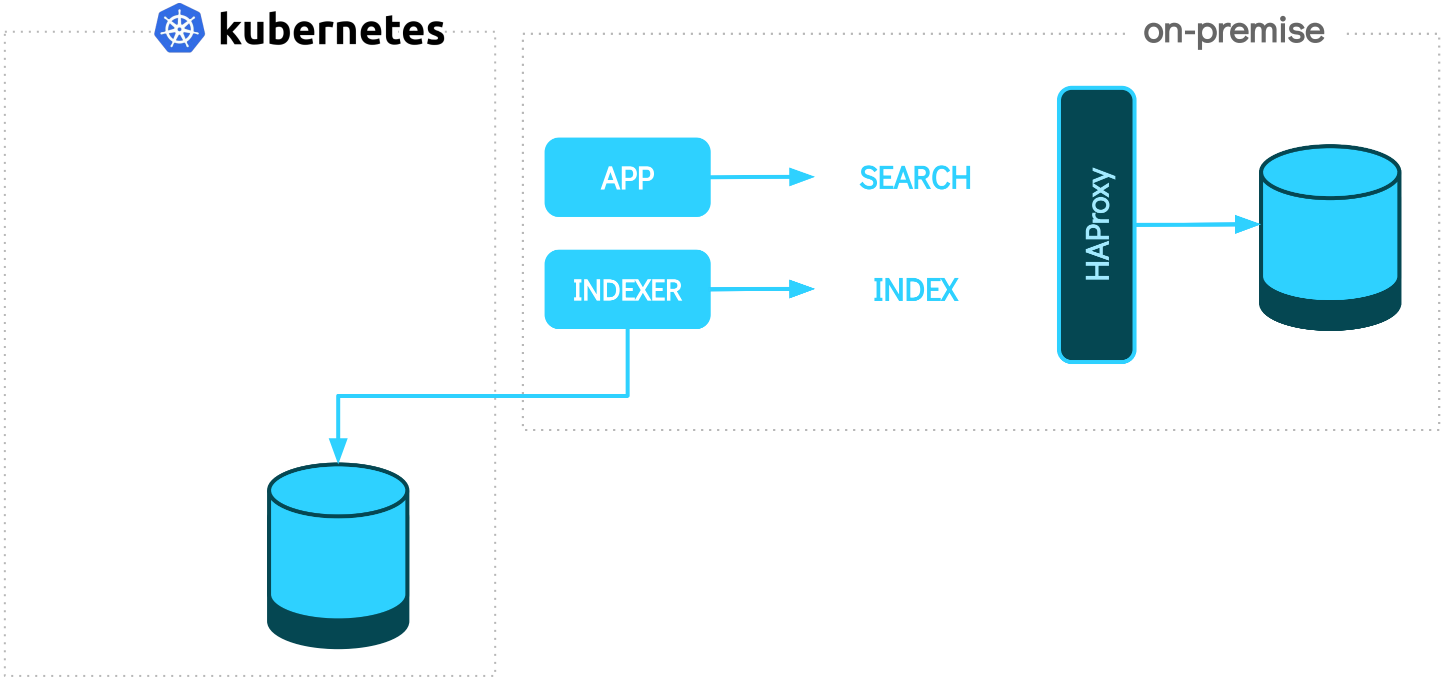
Elasticsearch

ELASTIC CLOUD ON KUBERNETES



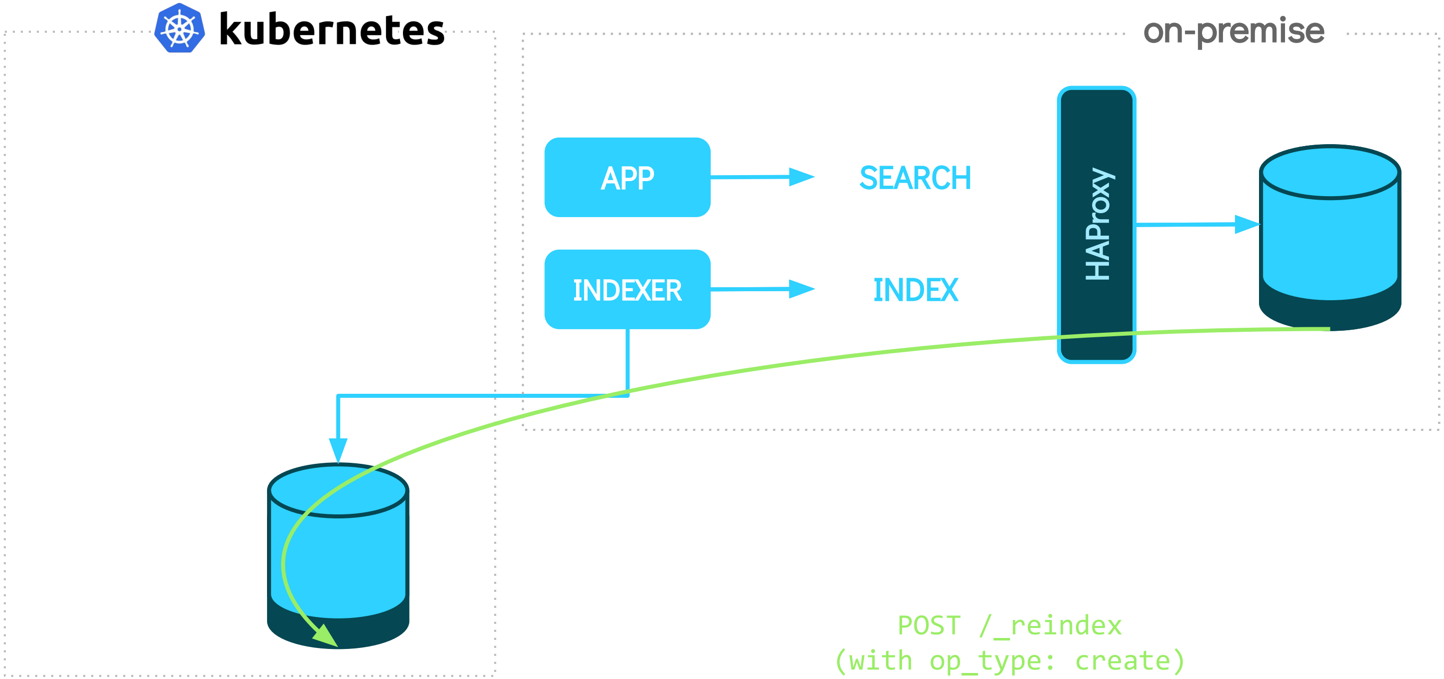
Elasticsearch Migration Path

Implement double-writes in indexer



Elasticsearch Migration Path

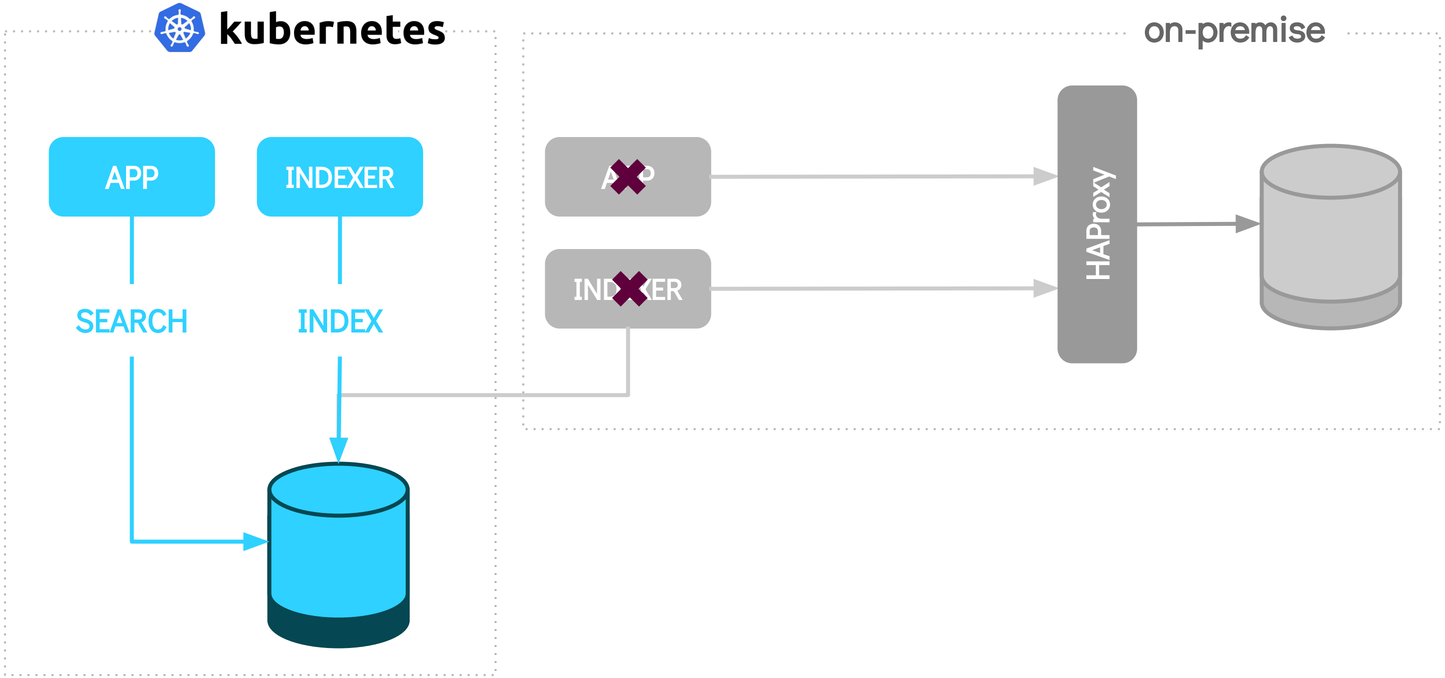
Get missing data from on-premise



POST /_reindex
(with op_type: create)

Elasticsearch Migration Path

Move the application



That’s only two use cases…

Solutions chosen in GCP

Cassandra	→	Helm Chart BlaBlaCar
RabbitMQ	→	Helm Chart Bitnami
Redis	→	MemoryStore
PostgreSQL	→	CloudSQL
Kafka	→	K8S Operator Strimzi
Couchbase	→	MemoryStore

“Thanks!

ONLINE, AROUND-THE-WORLD OPEN SOURCE DATABASE CONFERENCE

PERCONA

LIVEONLINE

#ZeroEmptySeats