# Running MongoDB in the Cloud:

# How to Reduce Infrastructure Costs Using Open Source Components and Kubernetes

Takis Stathopoulos
Enterprise Architect, Percona

?

PERCONA .CONNECT

## Agenda

1. Why proprietary MongoDB Cloud costs are getting out of control?
2. The Open, Cloud Native, Option
3. Practical approaches to cost reduction
4. Summary

## Why proprietary MongoDB Cloud costs are getting out of control?

# DBaaS convenience fee

| Cloud and region | MongoDB Atlas | IaaS pay-as-you-go | IaaS reserved (3y) |
|---|---|---|---|
| **M30 (3-node RS, 2vCPU, 8GB RAM, 40GB disk)** | | | |
| AWS us-east1 | $4,730 | **$2,666** | $1,204 |
| Azure centralus | $5,256 | **$2,450** | $1,049 |
| GCP us-east1 | $3,854 | **$2,796** | $1,393 |
| **M200 (4x3-node RS, 64vCPU, 256GB RAM, 2TB storage)** | | | |
| AWS us-east1 | $555,822 | **$355,086** | $159,674 |
| Azure centralus | $626,515 | **$366,672** | $162,575 |
| GCP us-east1 | $525,755 | **$379,376** | $198,303 |

**A premium of 138% to 501% only for the server infrastructure part! Adds-up quickly!**
**No backup, no data transfer fees, yearly, public 2023 prices.**

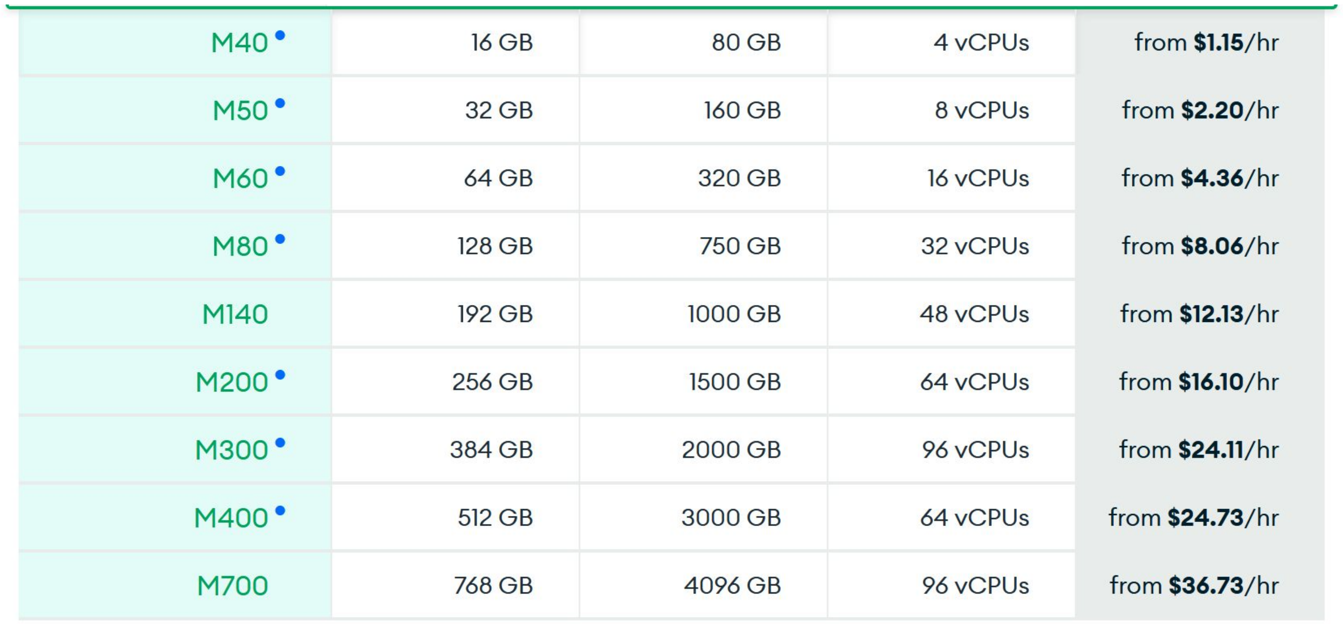
# Scaling Granularity

MongoDB Atlas, 3-node RS, AWS eu-west-1

**Couple of sizes fit all!**

**Scaling up:**

- Only 10 production-grade tiers
- Any upgrade costs double
- Limited options available



| | | | | |
|---|---|---|---|---|
| M40 ● | 16 GB | 80 GB | 4 vCPUs | from **$1.15**/hr |
| M50 ● | 32 GB | 160 GB | 8 vCPUs | from **$2.20**/hr |
| M60 ● | 64 GB | 320 GB | 16 vCPUs | from **$4.36**/hr |
| M80 ● | 128 GB | 750 GB | 32 vCPUs | from **$8.06**/hr |
| M140 | 192 GB | 1000 GB | 48 vCPUs | from **$12.13**/hr |
| M200 ● | 256 GB | 1500 GB | 64 vCPUs | from **$16.10**/hr |
| M300 ● | 384 GB | 2000 GB | 96 vCPUs | from **$24.11**/hr |
| M400 ● | 512 GB | 3000 GB | 64 vCPUs | from **$24.73**/hr |
| M700 | 768 GB | 4096 GB | 96 vCPUs | from **$36.73**/hr |

M50 $2.20/h = **$19,272/year**

M60 $4.36/hr = **$38,193.6/year**

## Ease of use **(and scale via credit card) trap**

- Cloud resources are virtually unlimited
- Pay-as-you go
- Less careful planning
- Building 20-shard 5-node each cluster? Easy!

## Limited flexibility

| What? | How? |
|---|---|
| Autoscaling | **Scale Up**<br>Avg. CPU Util > 75% for past 1h<br>Avg. Mem > 75% for past 1h<br>Next highest cluster tier<br>**Scale down**<br>Avg. Util. < 50% for the past 24h* |
| Topology | **No arbiter nodes**<br>**No single node dev clusters** |
| Uniform offering | **AWS, Azure, GCP common denominator** |

*https://www.mongodb.com/docs/atlas/cluster-autoscaling/

# Data transfer costs

- Hidden
- Hard to estimate
- Hard to verify
- Inter AZ -> Cross AZ -> Cross Region -> Internet

AWS: $0.01 - $0.09

Azure: $0.01 - $0.18

GCP: $0.01 -$0.23

## Backups

EBS Snapshots

- Snapshots: $0.05GB/month
- S3: $0.023GB/month

**Atlas AWS Snapshots**

- Snapshots: $0.19GB/month
- PITR: $1.55 - $0.40 (!)

# Symptoms of credit card scaling

- Overprovisioning config replica set nodes
- Too many shards for the dataset
- Lack of tuning and sticking to defaults
- Over Indexing
- Suboptimal shard keys
  - Unbalanced shards
  - Queries spanning all shards
- No data archival
- Default cache options

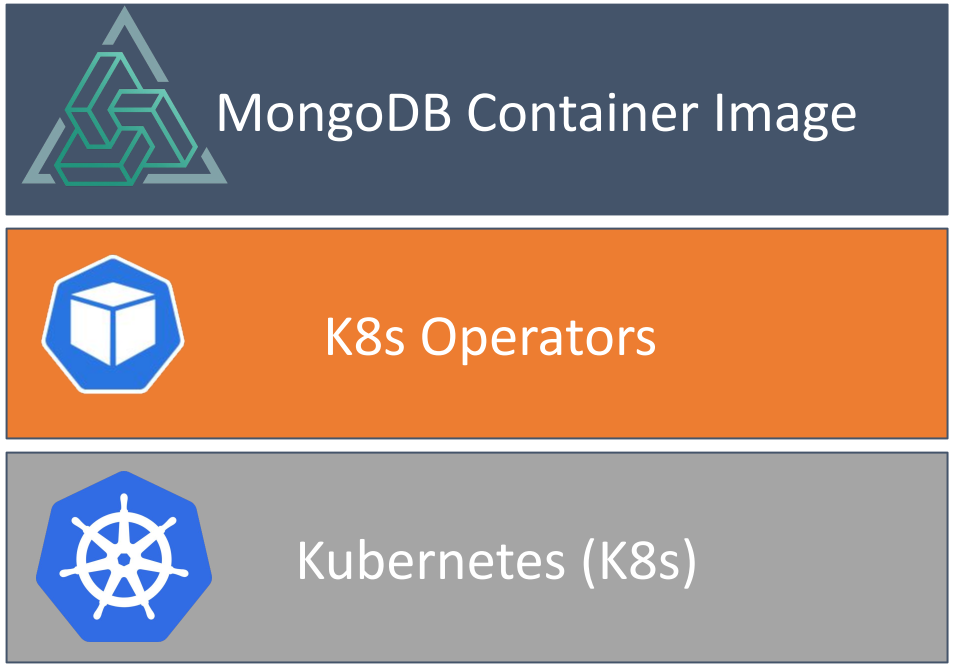
**What's the best tool to resolve MongoDB performance issues quickly?**

## Cloud DBaaS lock-in

PERCONA .CONNECT

## The Open, Cloud Native, Option

# The Open, way



**Observable, Containers, Microservices, …**

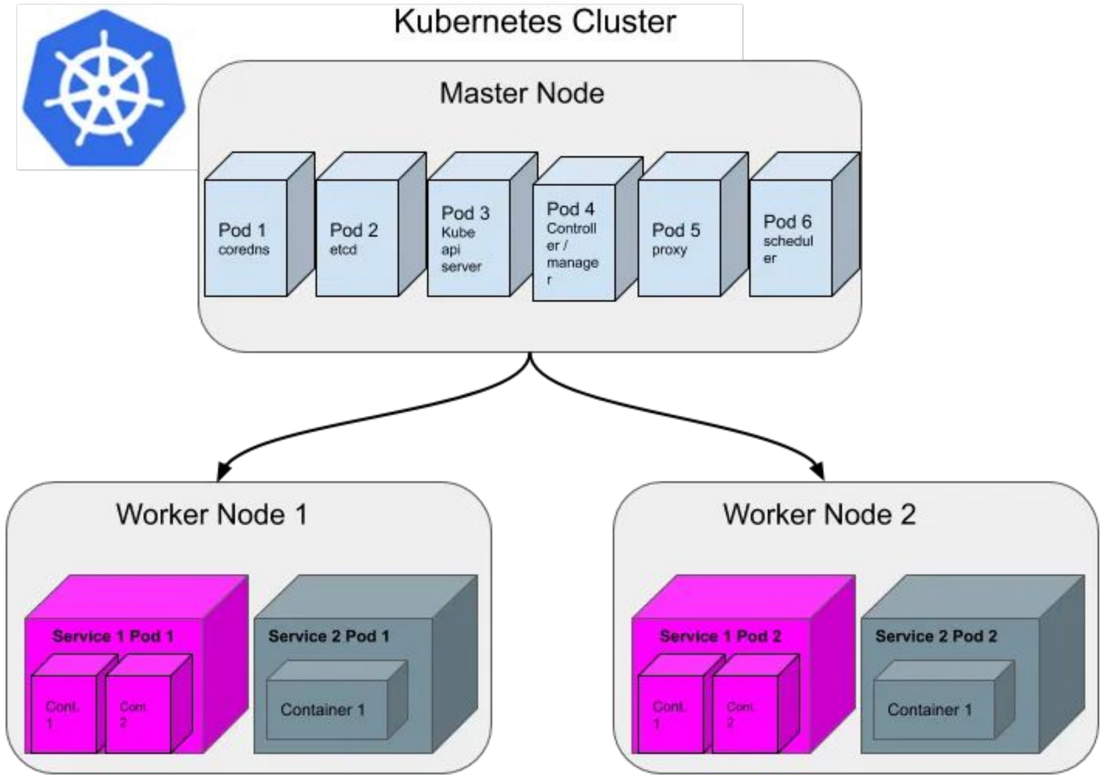
**Automated, Manageable, Easy to use, …**

**Scalable, Cloud Independent, Resilient, Declarative, …**

# Kubernetes in a single slide

**Basic objects**

**Cluster, Pods, Worker Nodes, Volumes, Secrets, Deployments, Services, ReplicaControllers, StatefulSets, Persistent Volume Claims ...**



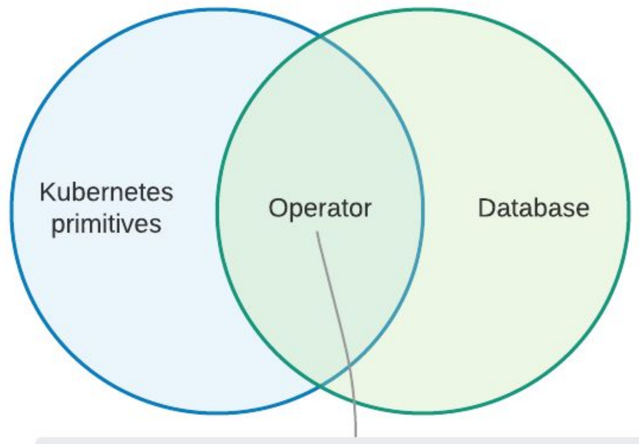
# But is it complex?

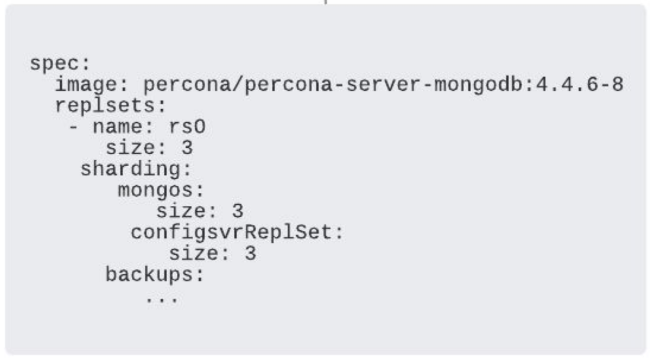
**Operators abstract and automate**

**Database - level concepts to K8s primitive transparently for the end - user**

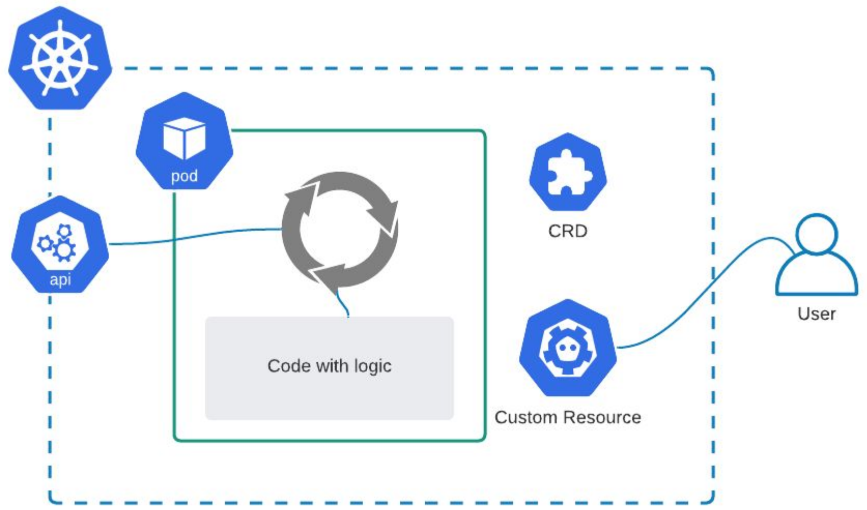
Operators are software extensions to Kubernetes that make use of Custom Resources Definitions (CRDs) to manage applications and their components.

**Percona Everest makes running DBs on K8s even simpler!**





```
spec:
  image: percona/percona-server-mongodb:4.4.6-8
  replsets:
   - name: rs0
       size: 3
    sharding:
       mongos:
          size: 3
        configsvrReplSet:
           size: 3
      backups:
          ...
```



# Deploying replica set Percona Operator for MongoDB

```
apiVersion: psmdb.percona.com/v1
kind: PerconaServerMongoDB
metadata:
  name: percona-live-cluster
spec:
  crVersion: 1.15.0
  image:
percona/percona-server-mongodb:6.0.4-3
  secrets:
    users: minimal-cluster
  replsets:
  - name: shard1
    size: 3
    resources:
      limits:
        cpu: "4"
        memory: "8G"
      requests:
        cpu: "4"
        memory: "8G"
    volumeSpec:
      persistentVolumeClaim:
        resources:
          requests:
            storage: 30Gi
```

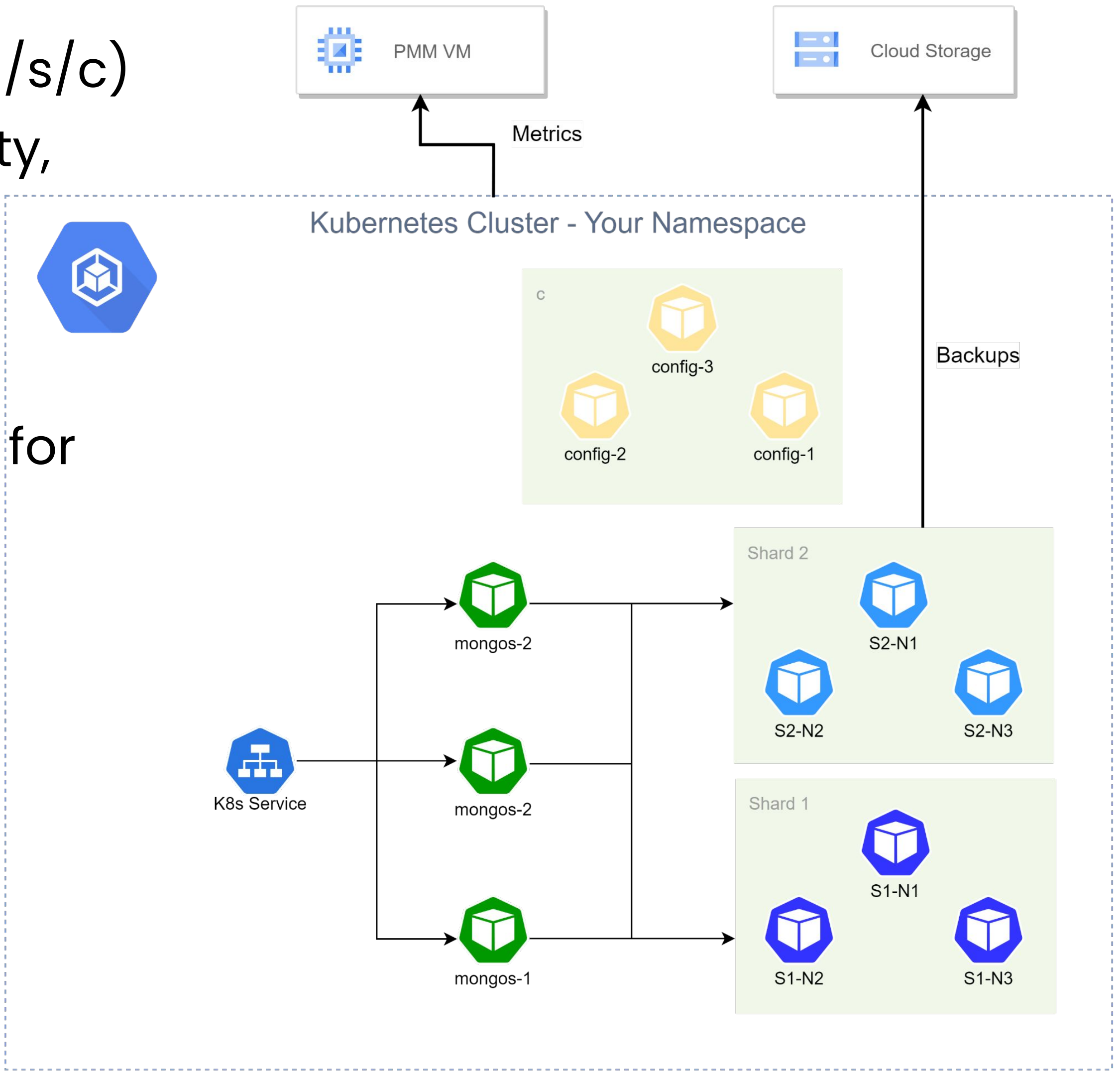
```
  sharding:
    enabled: true
    configsvrReplSet:
      size: 3
      resources:
        limits:
          cpu: "2"
          memory: "4G"
        requests:
          cpu: "2"
          memory: "4G"
      volumeSpec:
        persistentVolumeClaim:
          resources:
            requests:
              storage: 3Gi
    mongos:
      size: 3
```

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

https://docs.percona.com/percona-operator-for-mongodb/compare.html
Operators compared

# Percona Operator for MongoDB

1. Deploy easily: replica sets, shards, (mongo/d/s/c)
2. Topology management (arbiters, node affinity, scaling)
3. Monitoring integration
4. Network exposure and load balancing
5. Backups management with Percona Backup for MongoDB
6. Self-healing
7. Upgrade automation (minor, manual major)
8. Configuration adjustments



kubernetes OPENSHIFT Google Cloud Platform aws Microsoft Azure

# Percona Server for MongoDB

Binary compatible, drop-in replacement

for MongoDB CE . No license fees, free to use

Enterprise features, without the restrictions

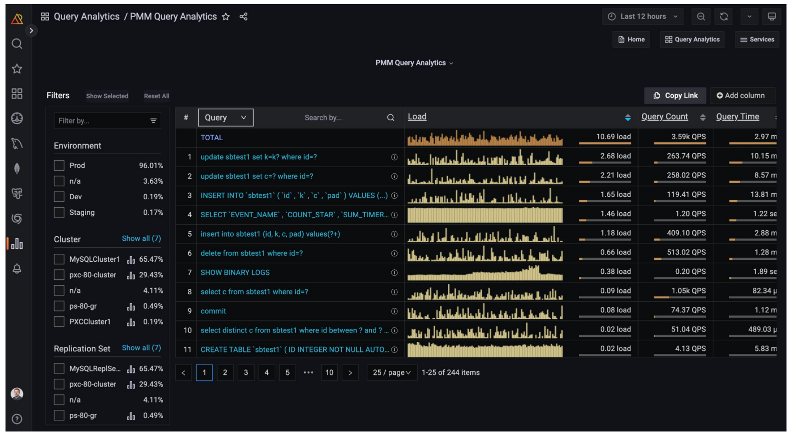
1. Advanced backups (Physical, PITR)
2. LDAP Integration
3. Data-at-rest encryption
4. KMIP integration
5. Auditing
6. PMM Monitoring

**Enterprise Level QA**

Test and package for everyone!

**Enterprise Features**

Bring in the enterprise features companies need.



Query Analytics / PMM Query Analytics

Last 12 hours

Home | Query Analytics | Services

PMM Query Analytics

Filters — Show Selected — Reset All

Filter by...

Environment

|  |  |
|---|---|
| Prod | 96.01% |
| n/a | 3.63% |
| Dev | 0.19% |
| Staging | 0.17% |

Cluster — Show all (7)

|  |  |
|---|---|
| MySQLCluster1 | 65.47% |
| pxc-80-cluster | 29.43% |
| n/a | 4.11% |
| ps-80-gr | 0.49% |
| PXCCluster1 | 0.19% |

Replication Set — Show all (7)

|  |  |
|---|---|
| MySQLReplSe... | 65.47% |
| pxc-80-cluster | 29.43% |
| n/a | 4.11% |
| ps-80-gr | 0.49% |

Copy Link — Add column

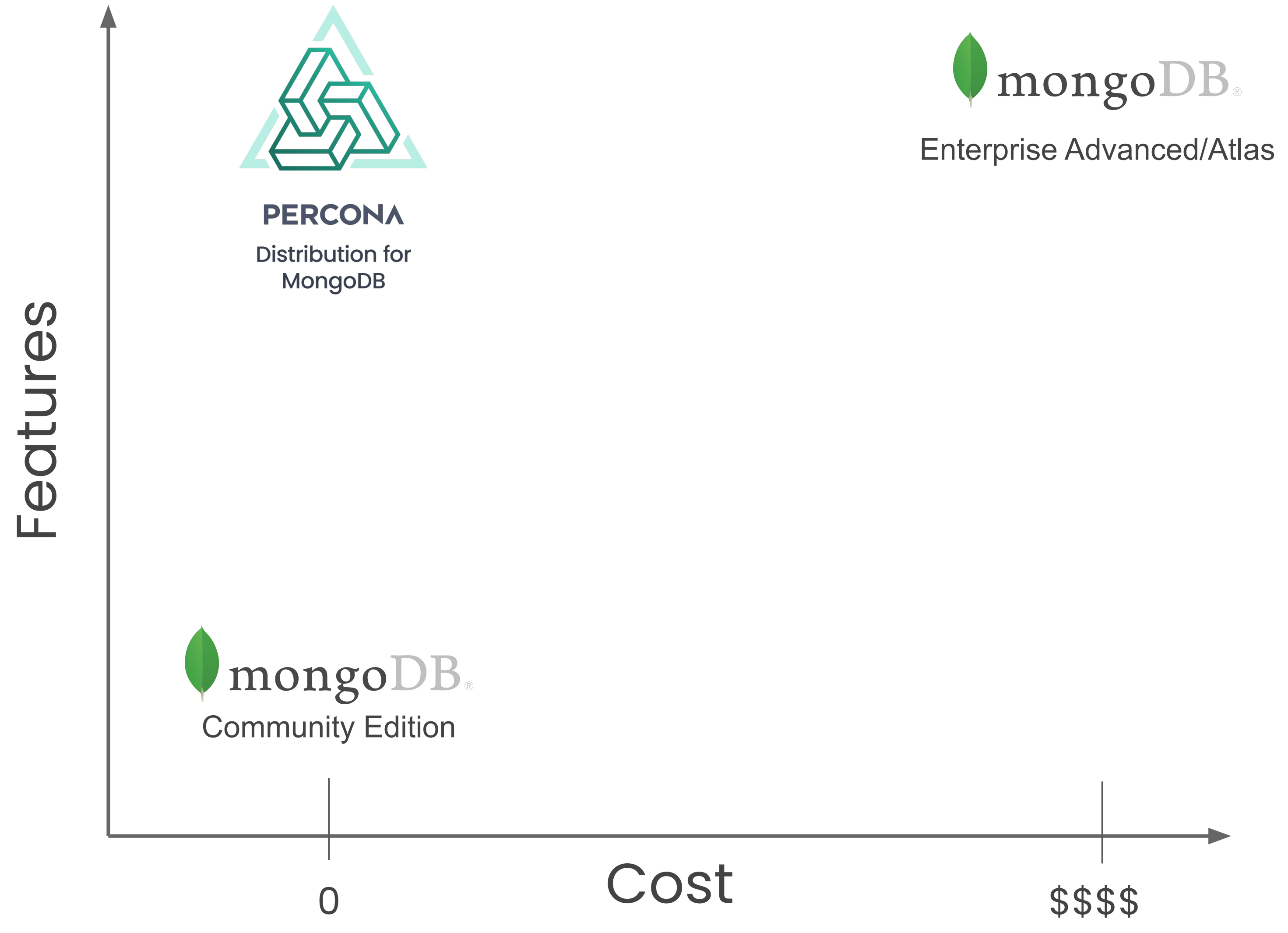
| # | Query | Load | Query Count | Query Time |
|---|---|---|---|---|
|  | TOTAL | 10.69 load | 3.59k QPS | 2.97 m |
| 1 | update sbtest1 set k=k? where id=? | 2.68 load | 263.74 QPS | 10.15 m |
| 2 | update sbtest1 set c=? where id=? | 2.21 load | 258.02 QPS | 8.57 m |
| 3 | INSERT INTO `sbtest1` ( `id` , `k` , `c` , `pad` ) VALUES (...) | 1.65 load | 119.41 QPS | 13.81 m |
| 4 | SELECT `EVENT_NAME` , `COUNT_STAR` , `SUM_TIMER... | 1.46 load | 1.20 QPS | 1.22 se |
| 5 | insert into sbtest1 (id, k, c, pad) values(?+) | 1.18 load | 409.10 QPS | 2.88 m |
| 6 | delete from sbtest1 where id=? | 0.66 load | 513.02 QPS | 1.28 m |
| 7 | SHOW BINARY LOGS | 0.38 load | 0.20 QPS | 1.89 se |
| 8 | select c from sbtest1 where id=? | 0.09 load | 1.05k QPS | 82.34 µ |
| 9 | commit | 0.08 load | 74.37 QPS | 1.12 m |
| 10 | select distinct c from sbtest1 where id between ? and ? ... | 0.02 load | 51.04 QPS | 489.03 µ |
| 11 | CREATE TABLE `sbtest1` ( ID INTEGER NOT NULL AUTO... | 0.02 load | 4.13 QPS | 5.83 m |

< 1 2 3 4 5 ••• 10 > 25 / page 1-25 of 244 items

## Practical approaches to cost reduction

# Use open source

1. Advanced backups
2. LDAP
3. Data-at-rest encryption
4. KMIP integration
5. Auditing
6. Monitoring



## K8s resources are cost-efficient

- Managed K8s is ~$70/month at any major cloud provider
- Utilize “raw” resources at “raw” resource prices
- Embrace discounts:
  - Reserved prices (41% 1 year, 62% 3-year)
  - Spot instances

# Embrace the variety of available cloud compute resources

```
eks.amazonaws.com/capacityType=ON_DEMAND

eks.amazonaws.com/nodegroup=critical

node.kubernetes.io/instance-type=r5.2xlarge

topology.kubernetes.io/region=eu-west-1

topology.kubernetes.io/zone=eu-west-1b
```

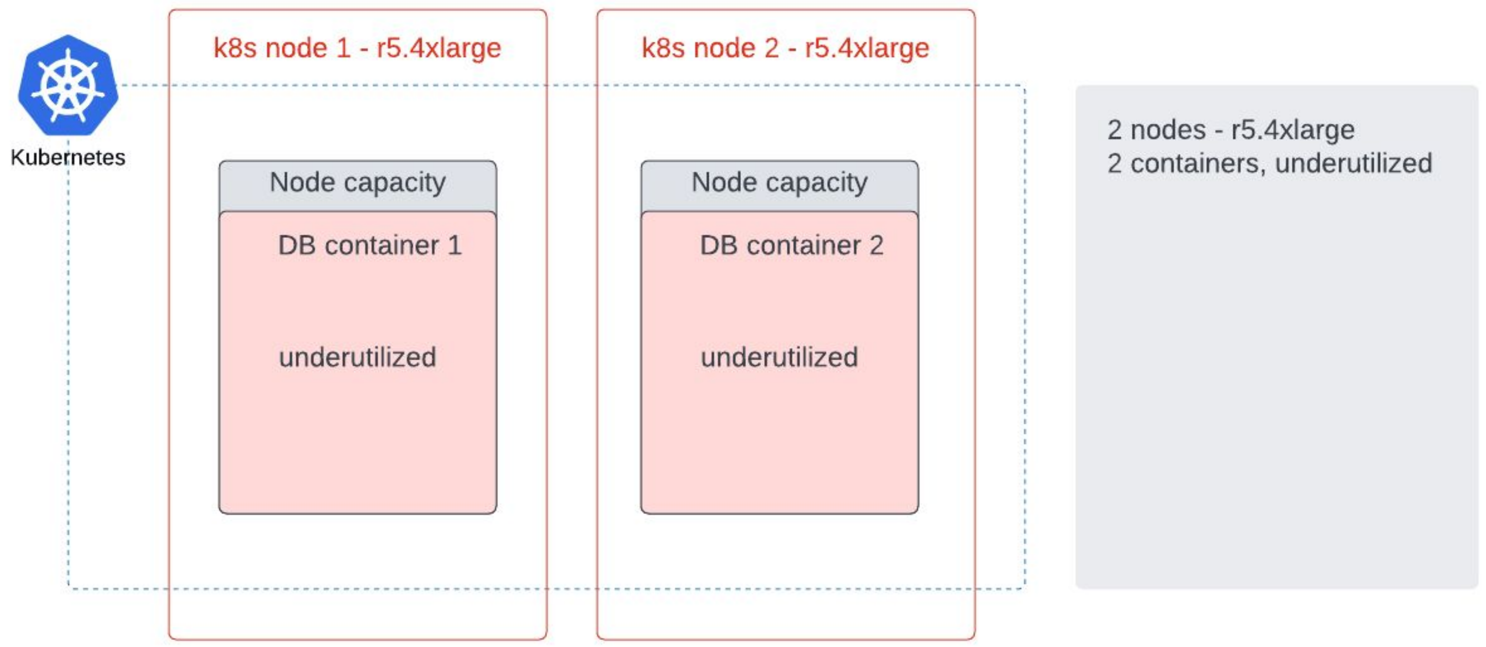
# Assign resources granularly

1. Each shard
    a. Data nodes
    b. Hidden nodes
    c. Arbiters
2. Config RS
3. Mongos
4. Backup pod

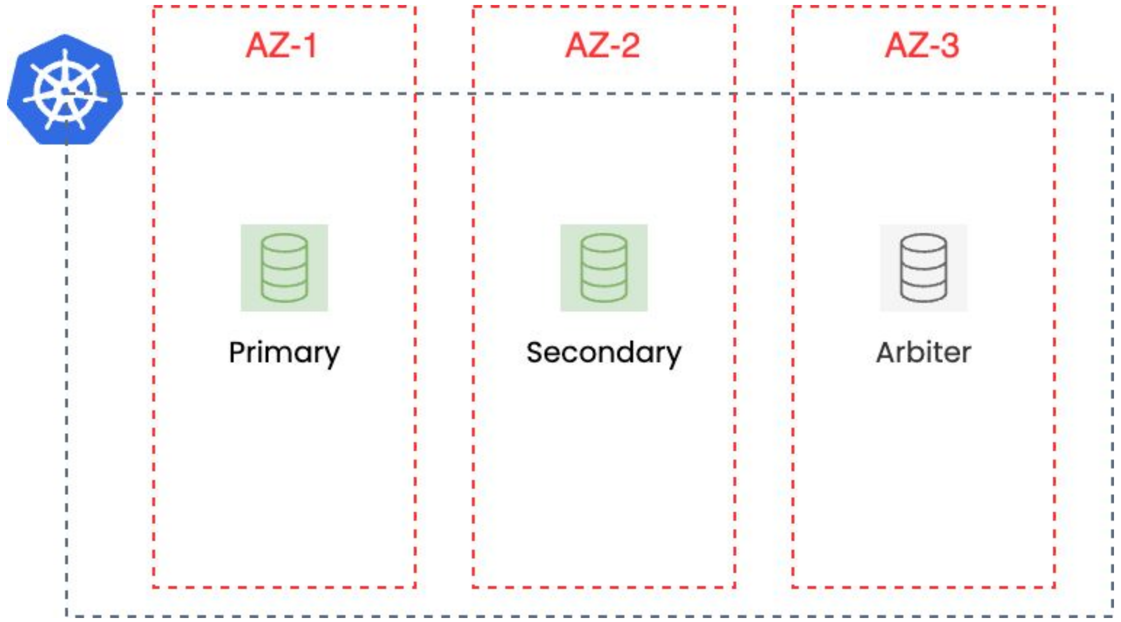
```
replsets:
- name: shard1
  size: 3
  resources:
    limits:
      cpu: "12.5"
      memory: "19G"
    requests:
      cpu: "12.5"
      memory: "19G"
```

# Embrace K8s Autoscaling flexibility

1. VPA
2. HPA/KEDA
3. Cluster Autoscaler



## Choose the right HA-level and topology

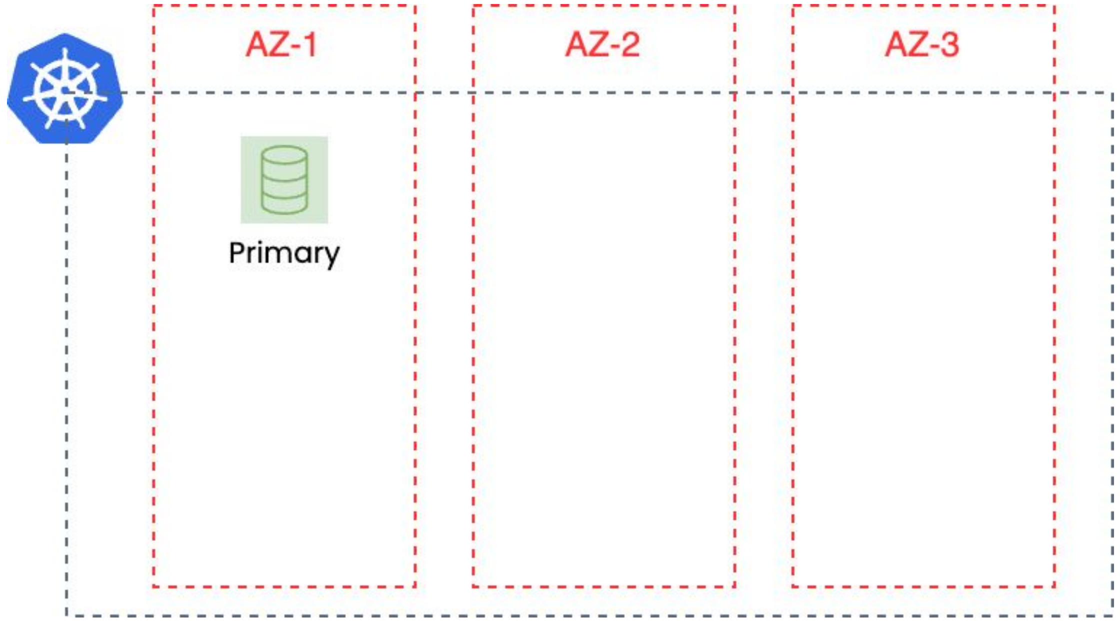


```
spec:
  allowUnsafeConfigurations: false

  replsets:
    - name: shard1
      size: 3
      affinity:
        antiAffinityTopologyKey:
         "kubernetes.io/zone"

     arbiter:
       enabled: true
       size: 1
```

## Cost effective Development Environments



```
spec:
  allowUnsafeConfigurations: true

  replsets:
    - name: shard1
      size: 1
      affinity:
        antiAffinityTopologyKey:
         "kubernetes.io/host"

     arbiter:
       enabled: false
```

# Use multiple storage classes

3

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: bronze
parameters:
  type: gp2
  fsType: xfs
```

2

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: silver
parameters:
  type: gp3
  iops: 6000
  fsType: xfs
```

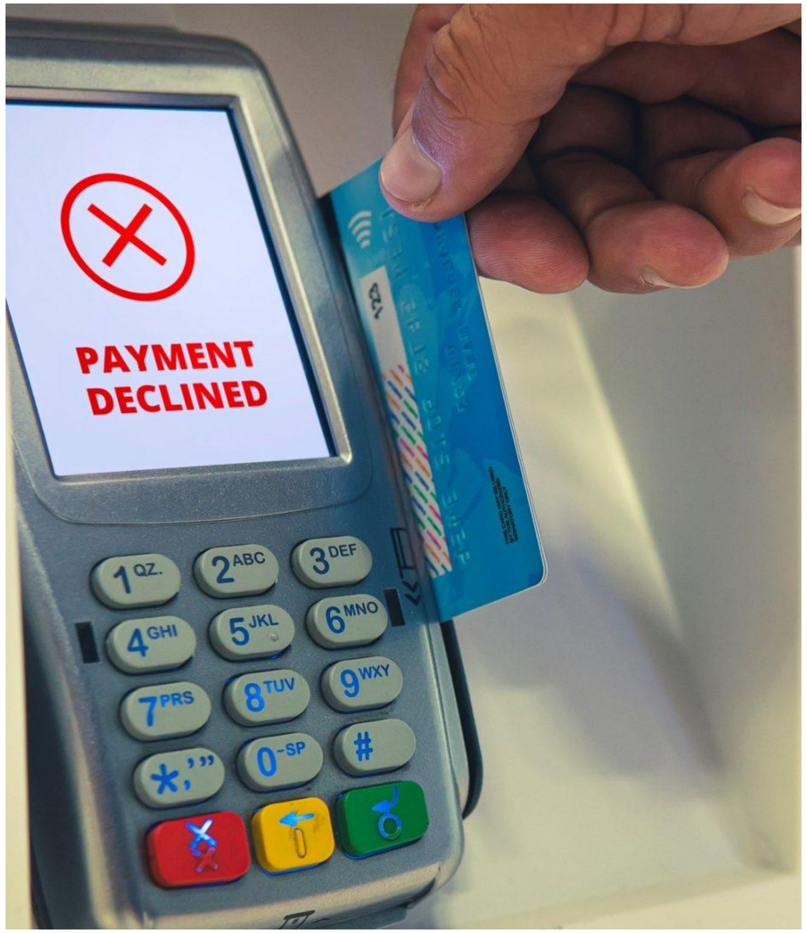
1

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: gold
parameters:
  type: io2
  iopsPerGB: 40
  allowAutoIOPSPerGBIncrease:
"true"
  fsType: xfs
```

```
replsets:
  - name: shard1
    size: 3
    volumeSpec:
      persistentVolumeClaim:
        storageClassName: gold
        resources:
          requests:
            storage: 300G
```

## Stop solving issues with credit card

- Indexes:
  - Check usage stats
  - Consider partial indexes
  - Remove unused indexes
- Sharding key:
  - Plan it
  - Adjust it
- Data archival
  - Capped collection
  - TTL indexes



# Understanding is a key

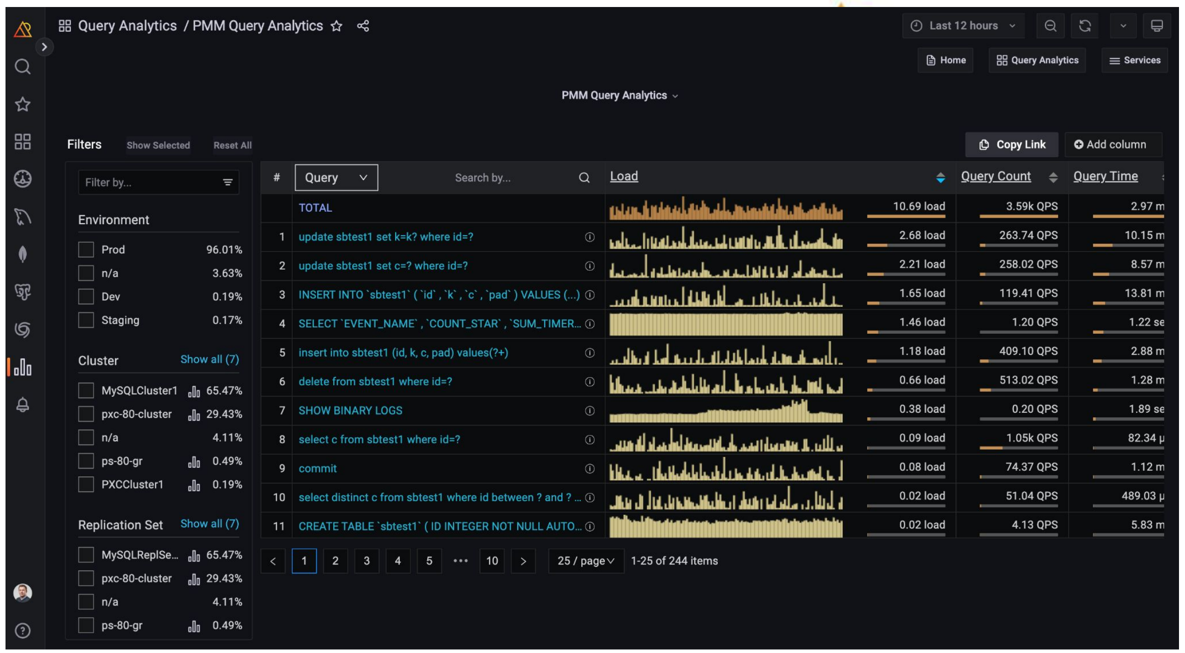
- Understand the workload
- Review outstanding queries
- Look for underutilization, too

```
helm repo add percona
https://percona.github.io/percona-helm-charts/
helm install pmm \
--set secret.create=false \
--set secret.name=pmm-secret \
percona/pmm
```

```
pmm:
    enabled: true
    image: percona/pmm-client:2.37.0
    serverHost: monitoring-service
```

# Understanding is key

- Understand the workload
- Review outstanding queries
- Look for underutilization, too



Query Analytics / PMM Query Analytics

Last 12 hours

Home | Query Analytics | Services

PMM Query Analytics

Filters  Show Selected  Reset All

Copy Link  Add column

Filter by...

Environment

| | |
|---|---|
| Prod | 96.01% |
| n/a | 3.63% |
| Dev | 0.19% |
| Staging | 0.17% |

Cluster  Show all (7)

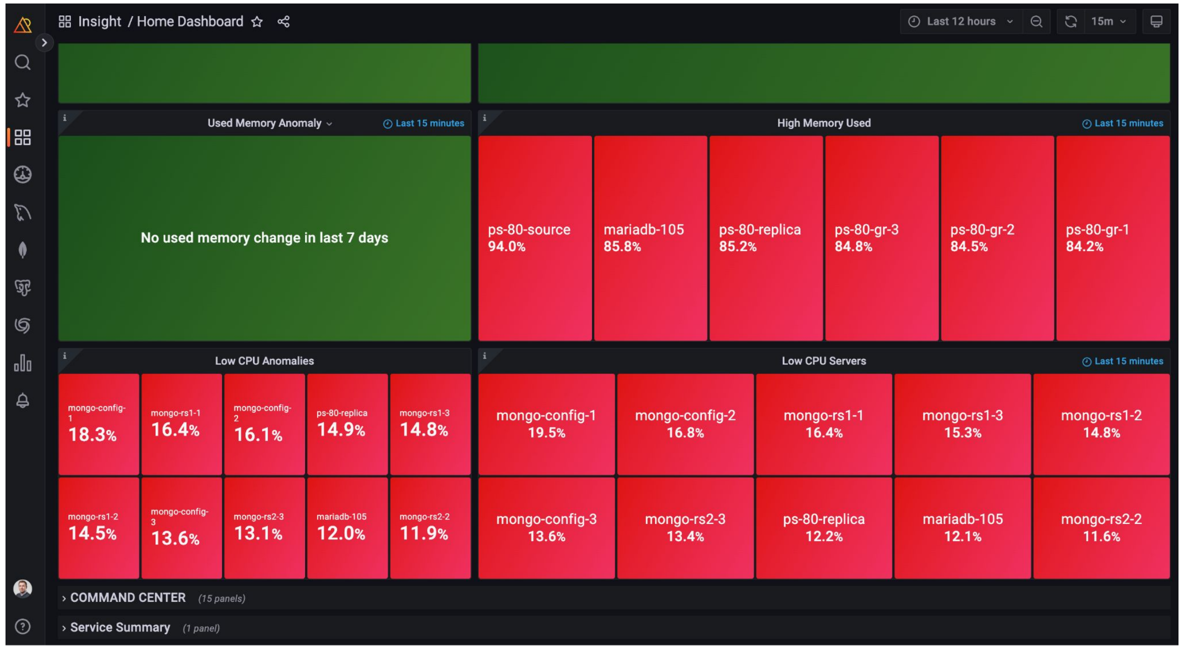
| | |
|---|---|
| MySQLCluster1 | 65.47% |
| pxc-80-cluster | 29.43% |
| n/a | 4.11% |
| ps-80-gr | 0.49% |
| PXCCluster1 | 0.19% |

Replication Set  Show all (7)

| | |
|---|---|
| MySQLReplSe... | 65.47% |
| pxc-80-cluster | 29.43% |
| n/a | 4.11% |
| ps-80-gr | 0.49% |

| # | Query | Load | Query Count | Query Time |
|---|---|---|---|---|
| | TOTAL | 10.69 load | 3.59k QPS | 2.97 m |
| 1 | update sbtest1 set k=k? where id=? | 2.68 load | 263.74 QPS | 10.15 m |
| 2 | update sbtest1 set c=? where id=? | 2.21 load | 258.02 QPS | 8.57 m |
| 3 | INSERT INTO `sbtest1` ( `id` , `k` , `c` , `pad` ) VALUES (...) | 1.65 load | 119.41 QPS | 13.81 m |
| 4 | SELECT `EVENT_NAME` , `COUNT_STAR` , `SUM_TIMER... | 1.46 load | 1.20 QPS | 1.22 se |
| 5 | insert into sbtest1 (id, k, c, pad) values(?+) | 1.18 load | 409.10 QPS | 2.88 m |
| 6 | delete from sbtest1 where id=? | 0.66 load | 513.02 QPS | 1.28 m |
| 7 | SHOW BINARY LOGS | 0.38 load | 0.20 QPS | 1.89 se |
| 8 | select c from sbtest1 where id=? | 0.09 load | 1.05k QPS | 82.34 µ |
| 9 | commit | 0.08 load | 74.37 QPS | 1.12 m |
| 10 | select distinct c from sbtest1 where id between ? and ? ... | 0.02 load | 51.04 QPS | 489.03 µ |
| 11 | CREATE TABLE `sbtest1` ( ID INTEGER NOT NULL AUTO... | 0.02 load | 4.13 QPS | 5.83 m |

< 1 2 3 4 5 ••• 10 >  25 / page  1-25 of 244 items



Insight / Home Dashboard

Last 12 hours  15m

Used Memory Anomaly  Last 15 minutes

No used memory change in last 7 days

High Memory Used  Last 15 minutes

ps-80-source 94.0%
mariadb-105 85.8%
ps-80-replica 85.2%
ps-80-gr-3 84.8%
ps-80-gr-2 84.5%
ps-80-gr-1 84.2%

Low CPU Anomalies

mongo-config-1 18.3%
mongo-rs1-1 16.4%
mongo-config-2 16.1%
ps-80-replica 14.9%
mongo-rs1-3 14.8%
mongo-rs1-2 14.5%
mongo-config-3 13.6%
mongo-rs2-3 13.1%
mariadb-105 12.0%
mongo-rs2-2 11.9%

Low CPU Servers  Last 15 minutes

mongo-config-1 19.5%
mongo-config-2 16.8%
mongo-rs1-1 16.4%
mongo-rs1-3 15.3%
mongo-rs1-2 14.8%
mongo-config-3 13.6%
mongo-rs2-3 13.4%
ps-80-replica 12.2%
mariadb-105 12.1%
mongo-rs2-2 11.6%

› COMMAND CENTER (15 panels)

› Service Summary (1 panel)

## Summary

## What’s the potential?

It depends.

CASE STUDY

PERCONA

minsait

An Indra company

# Minsait Migrates Tier 1 Telecoms Customers to MongoDB on Google Cloud with Percona

## Cloud Migration Challenges

Minsait is the digital transformation specialist company and part of the Indra Group. The firm is one of Spain's leading IT services and transformation companies and operates across 100 countries worldwide. Minsait has experience across cloud, security, payments and combined physical and digital channels, so it can provide the most innovative technology to drive businesses and support innovation in public sector organizations alike, generating a positive impact on society.

For the Minsait team, helping customers migrate to the cloud is a key component in their digital transformation initiatives. For a tier-one telecoms company, this meant a move to Google Cloud. The customer had a MongoDB deployment in their data center, and the team at Minsait had to consider how to migrate this implementation into the cloud as well.

For Minsait, the key goals of this project were to manage costs and maintain control over the deployment. For the Product Director on the project, getting the right advice was essential.

# Key takeaways

1. Cloud DBaaS such as MongoDB Atlas comes with a huge premium fee

   In some cases 5x cloud resources cost.

2. Hidden costs and suboptimal MongoDB configuration make the situation worse

   Cloud DBaaS fees are difficult to predict and understand.

3. Kubernetes and Percona MongoDB Operator can handle any MongoDB workload

   The solution has been proven in production for many years.

4. Running MongoDB on K8s, or Percona Everest has a significant cost saving potential

   It makes it possible to utilize cheap “raw” cloud resources without giving up convenience and automation

## Learn more

Run MongoDB in your terms

PERCONA .CONNECT

## Thank you!