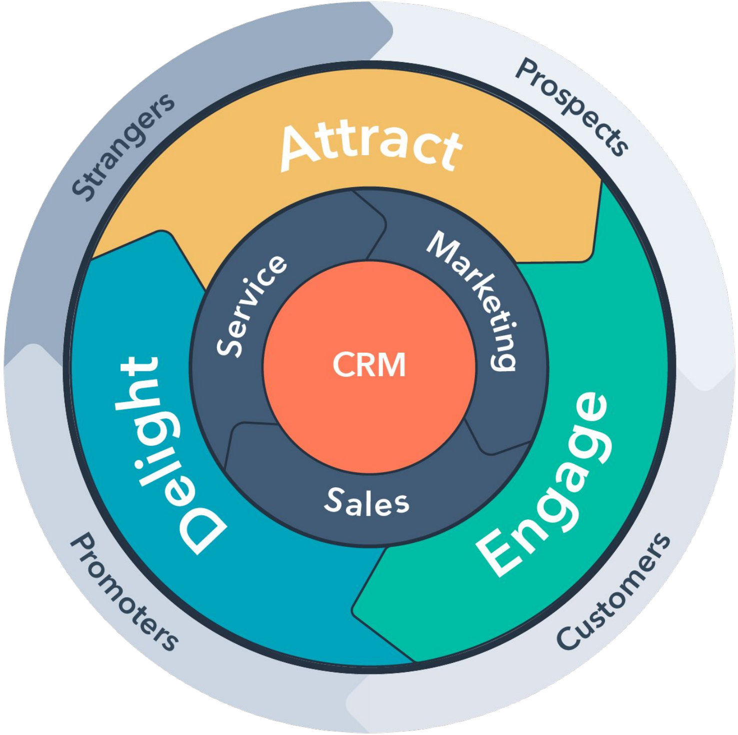
Vitess at HubSpot

How We Moved Hundreds of MySQL Databases into Kubernetes

Tom Petr, Alex Charis
KubeCon 2018

What is HubSpot?



Thousands of microservices

Hundreds of deploys each day

Many small, autonomous teams

2013...

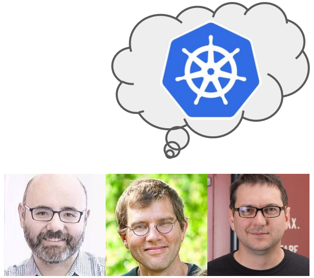
Clunky CI pipeline

Flaky ssh-based deployments

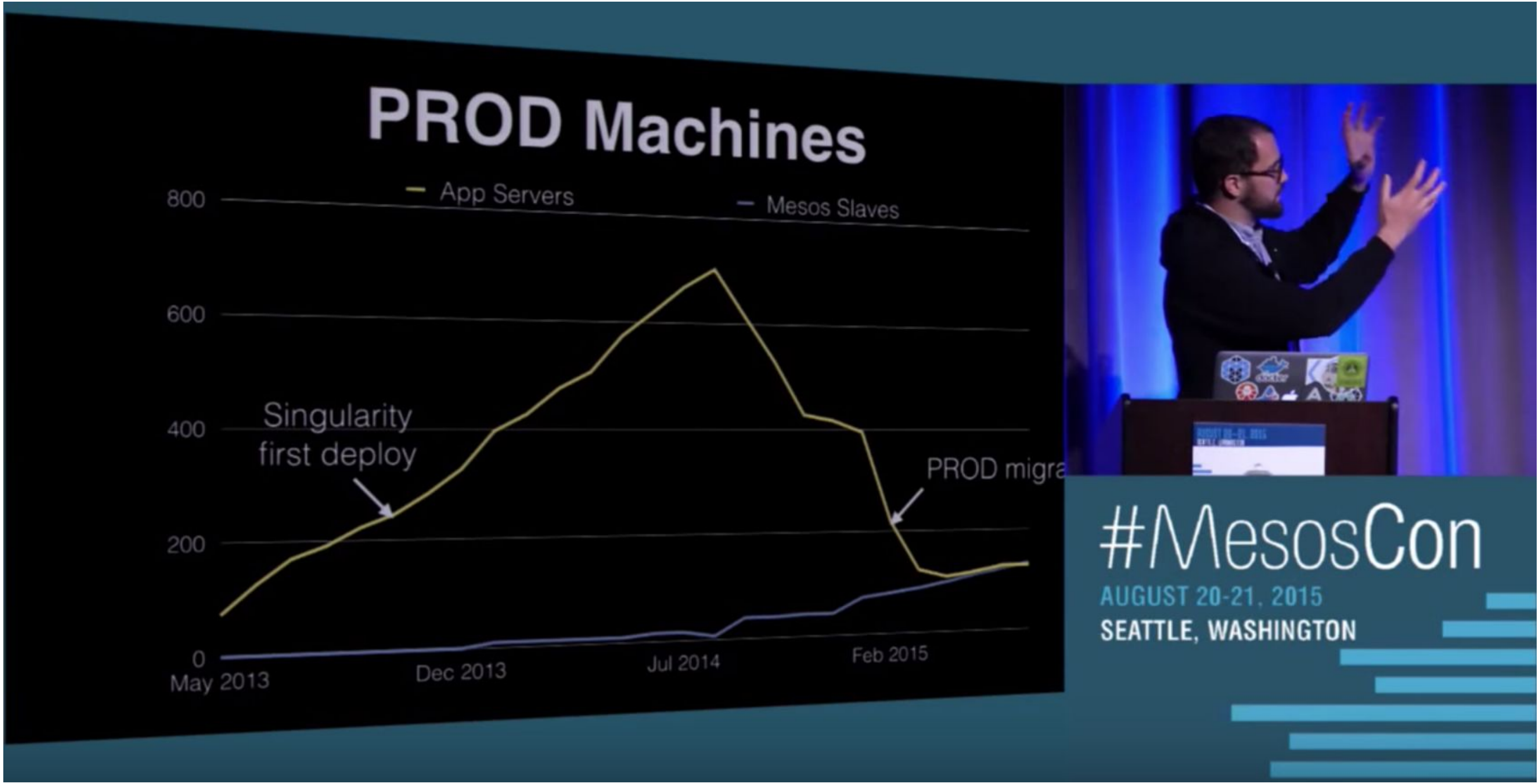
No elasticity

No automation

Singularity



PROD Machines

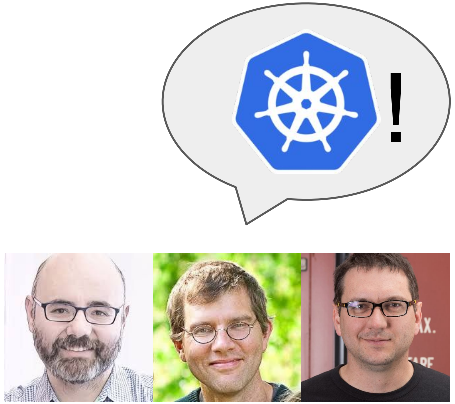


Singularity first deploy

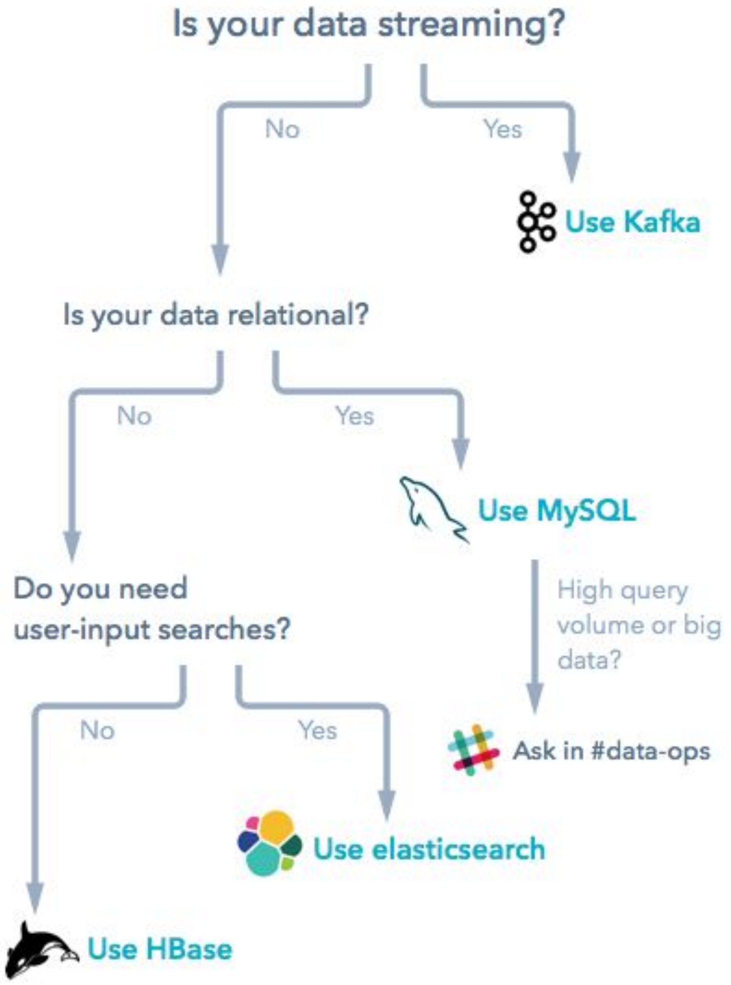
PROD migra

#MesosCon
AUGUST 20-21, 2015
SEATTLE, WASHINGTON

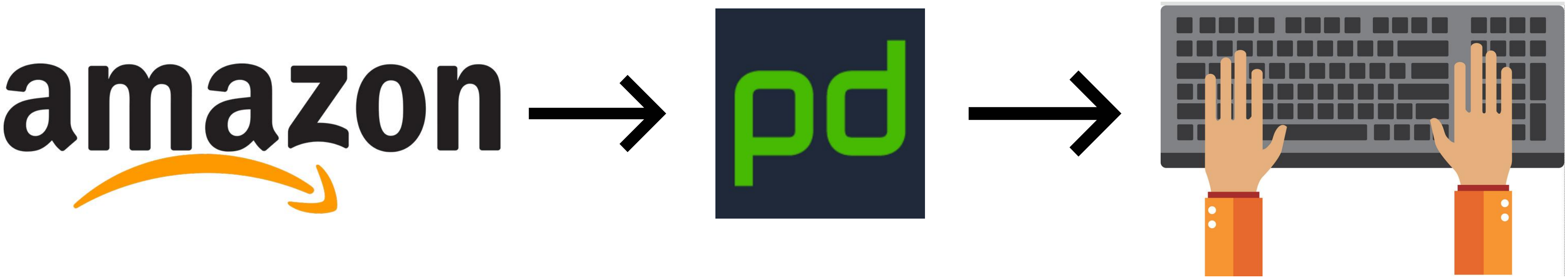
...2016: What about infrastructure?



Data at HubSpot in 2016

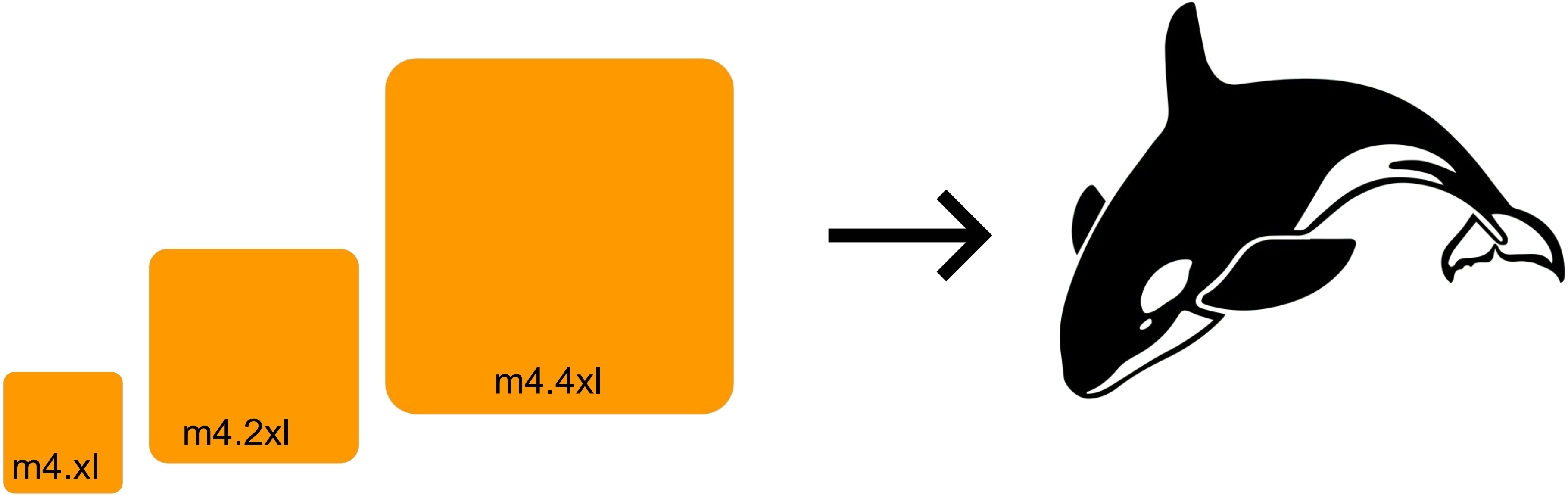


MySQL Scalability Woes



MySQL Scalability Woes

MySQL Scalability Woes

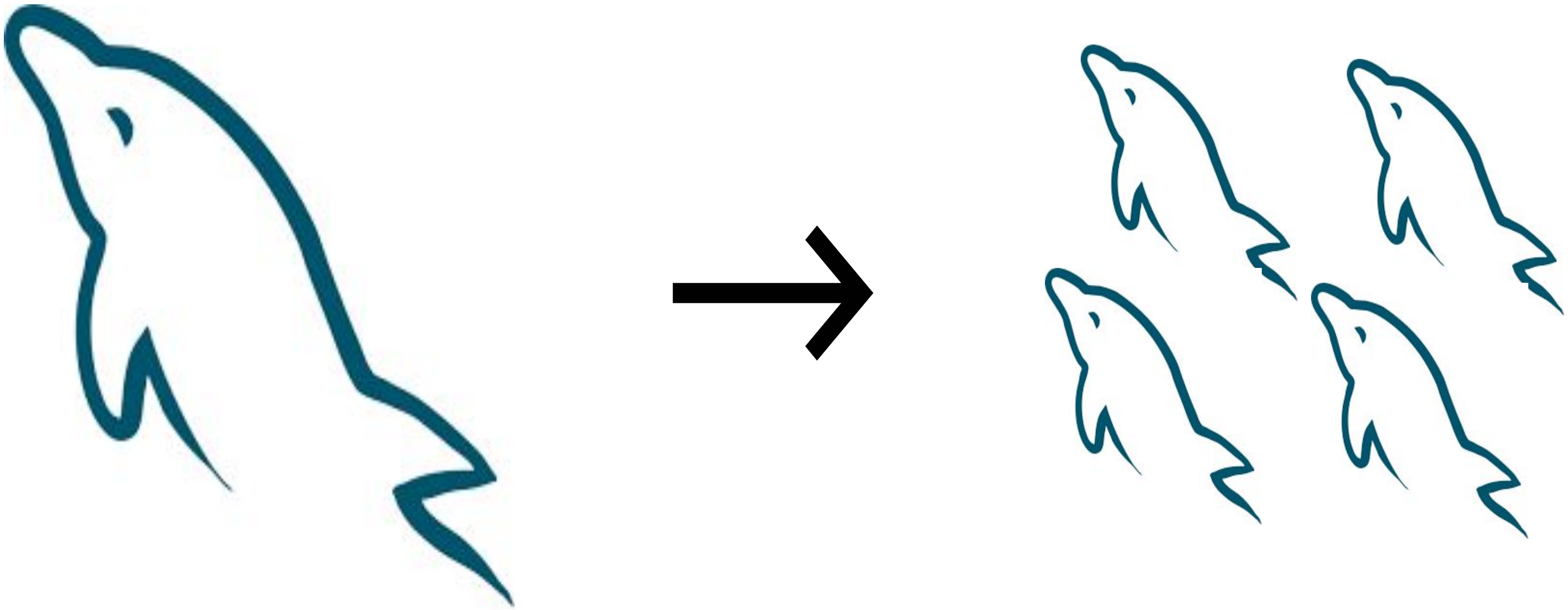


m4.xl

m4.2xl

m4.4xl

MySQL Scalability Woes



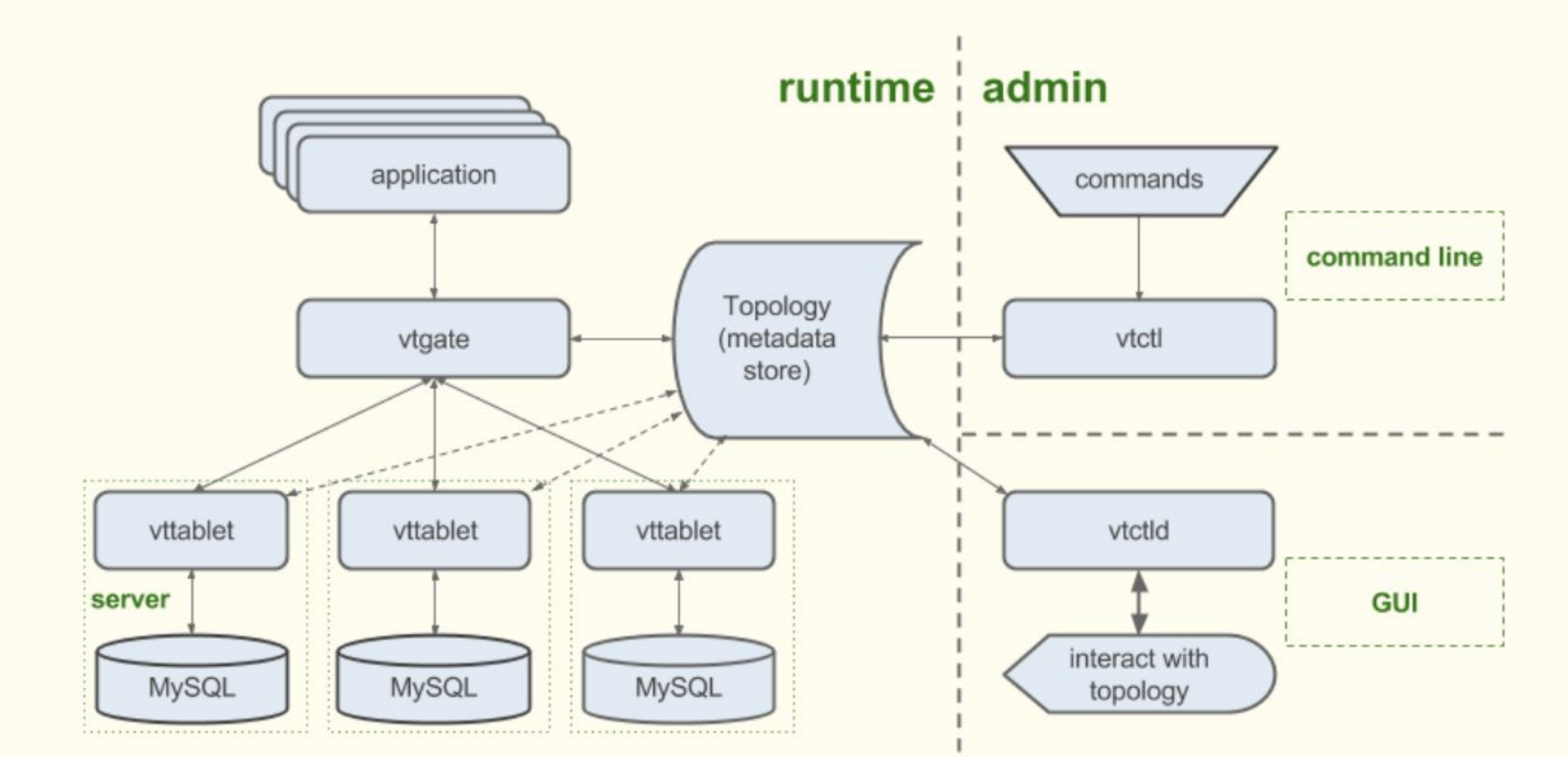
So what now?

So what now?

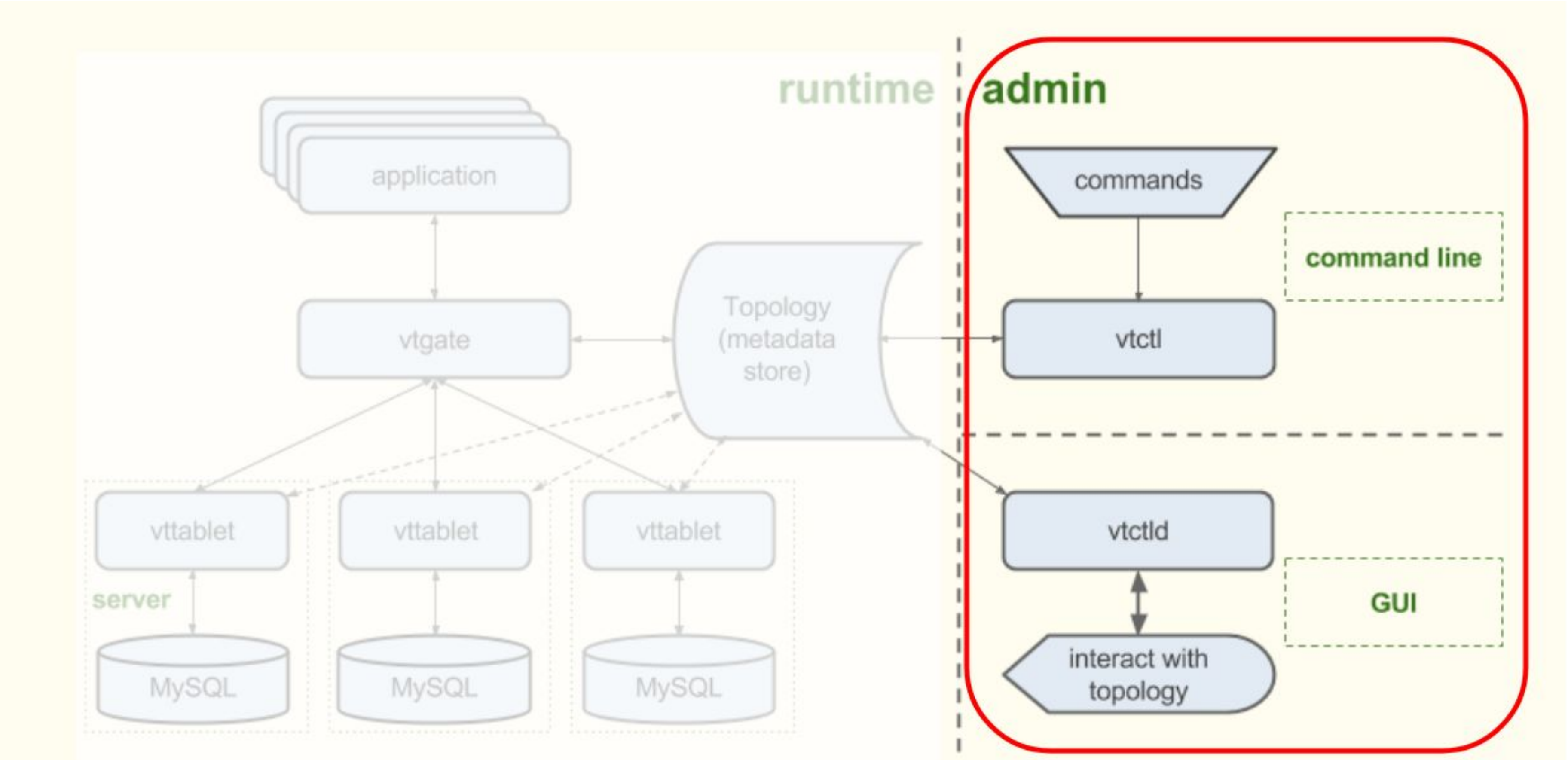
So what now?

So what now?

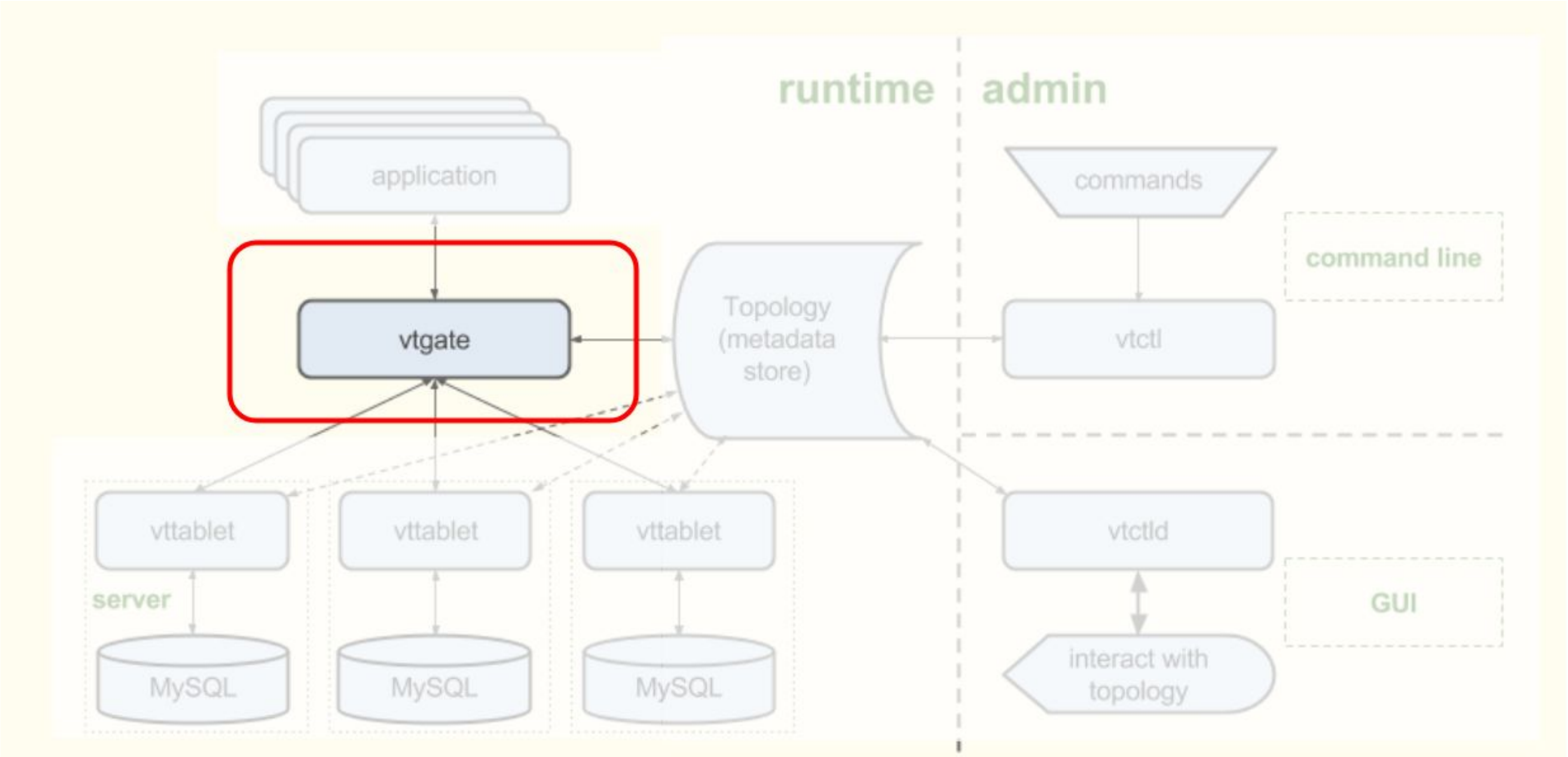
What is Vitess?



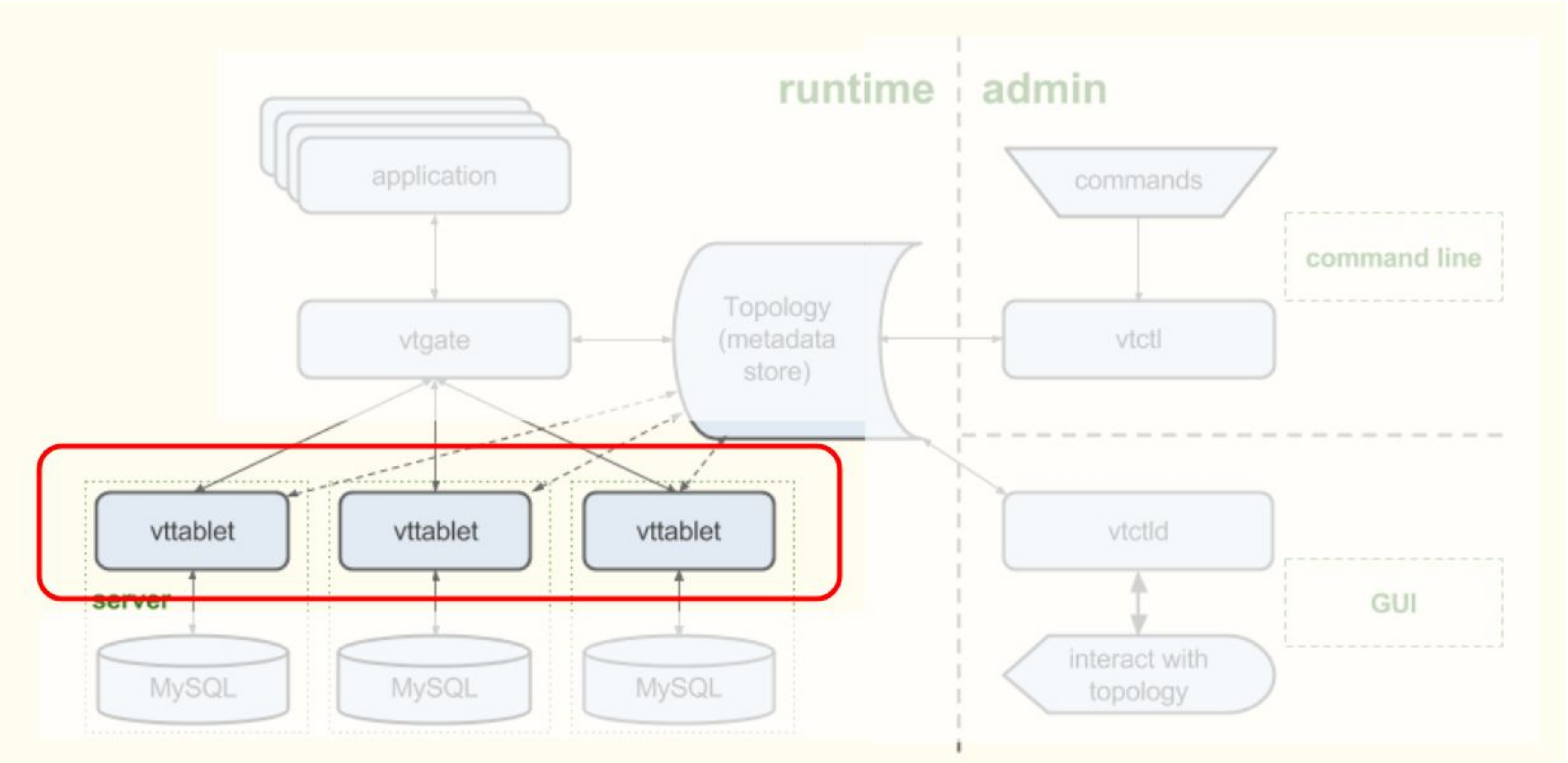
What is Vitess?



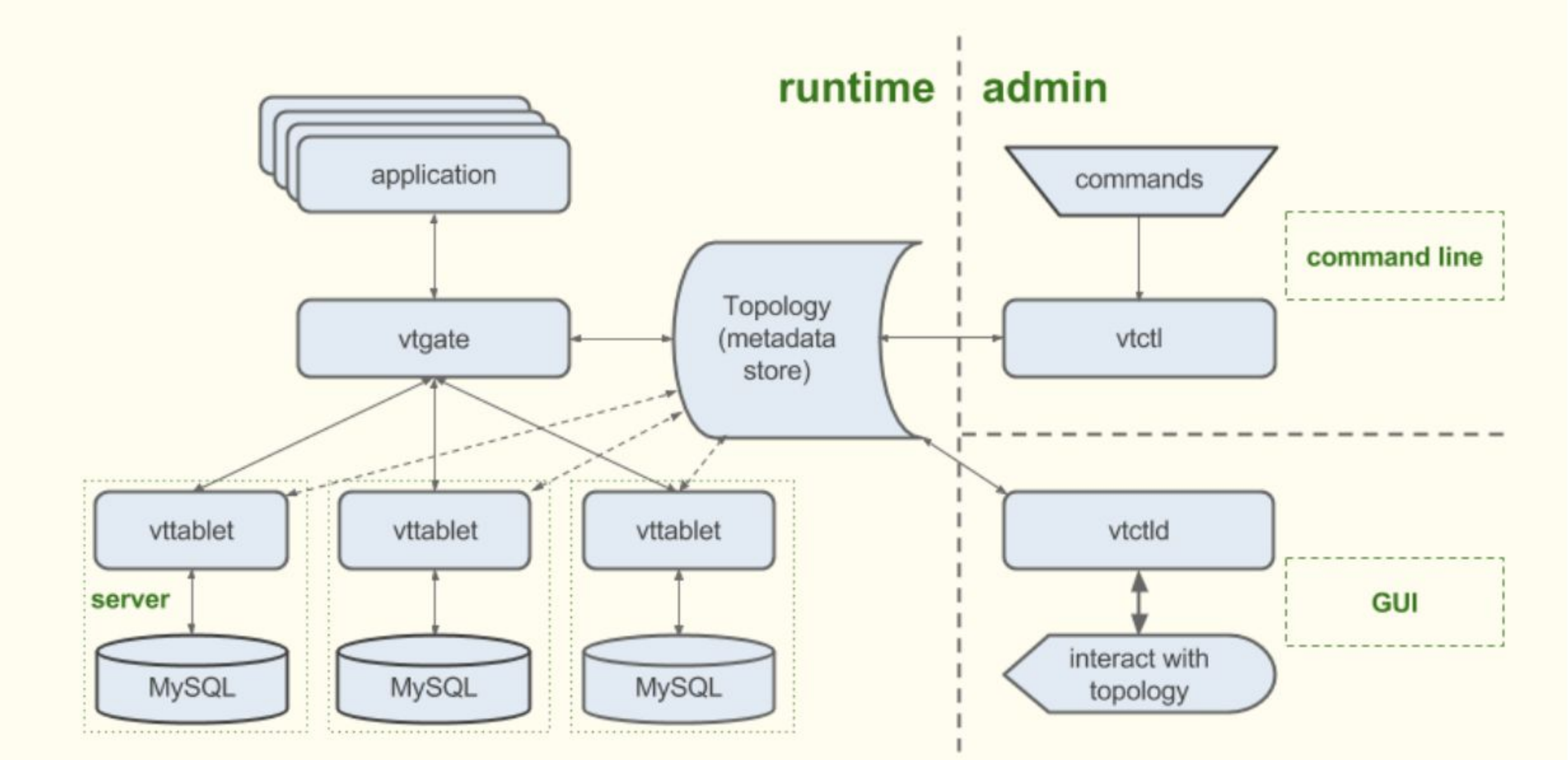
What is Vitess?



What is Vitess?

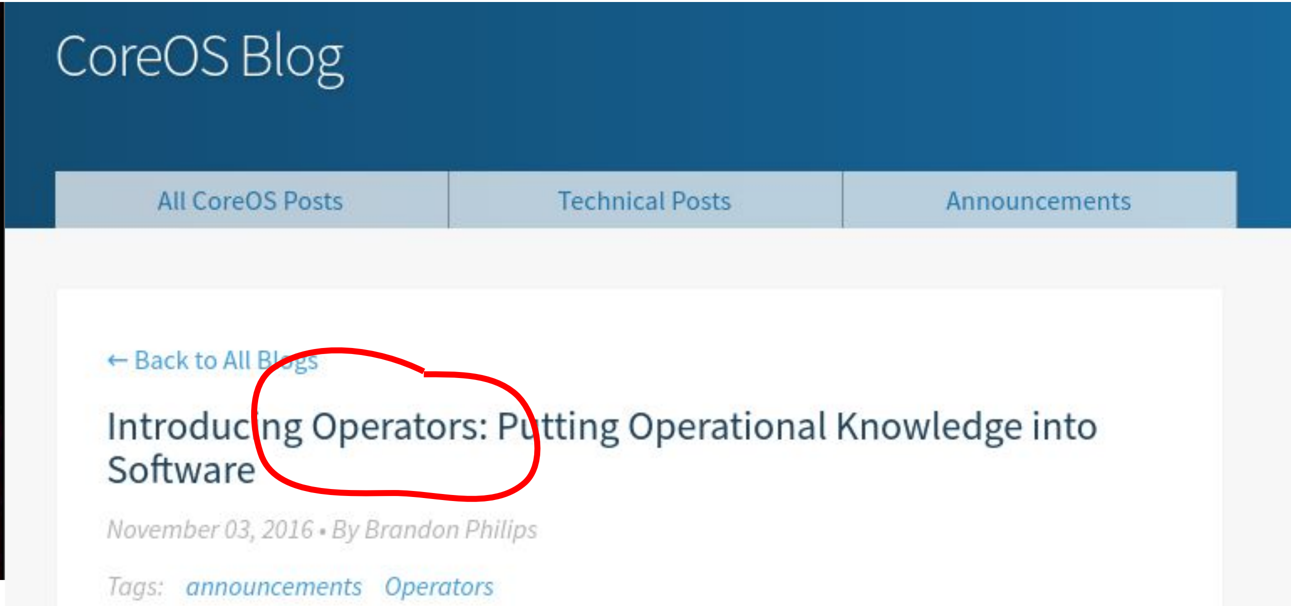


What is Vitess?



But how?

But how?

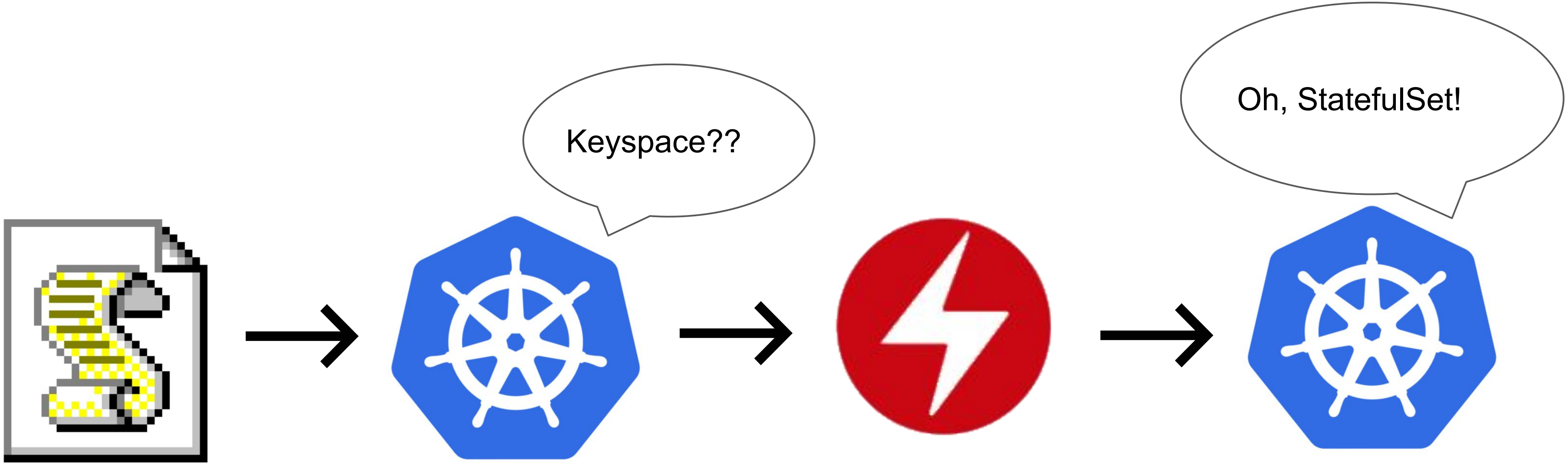


Introducing Operators: Putting Operational Knowledge into Software

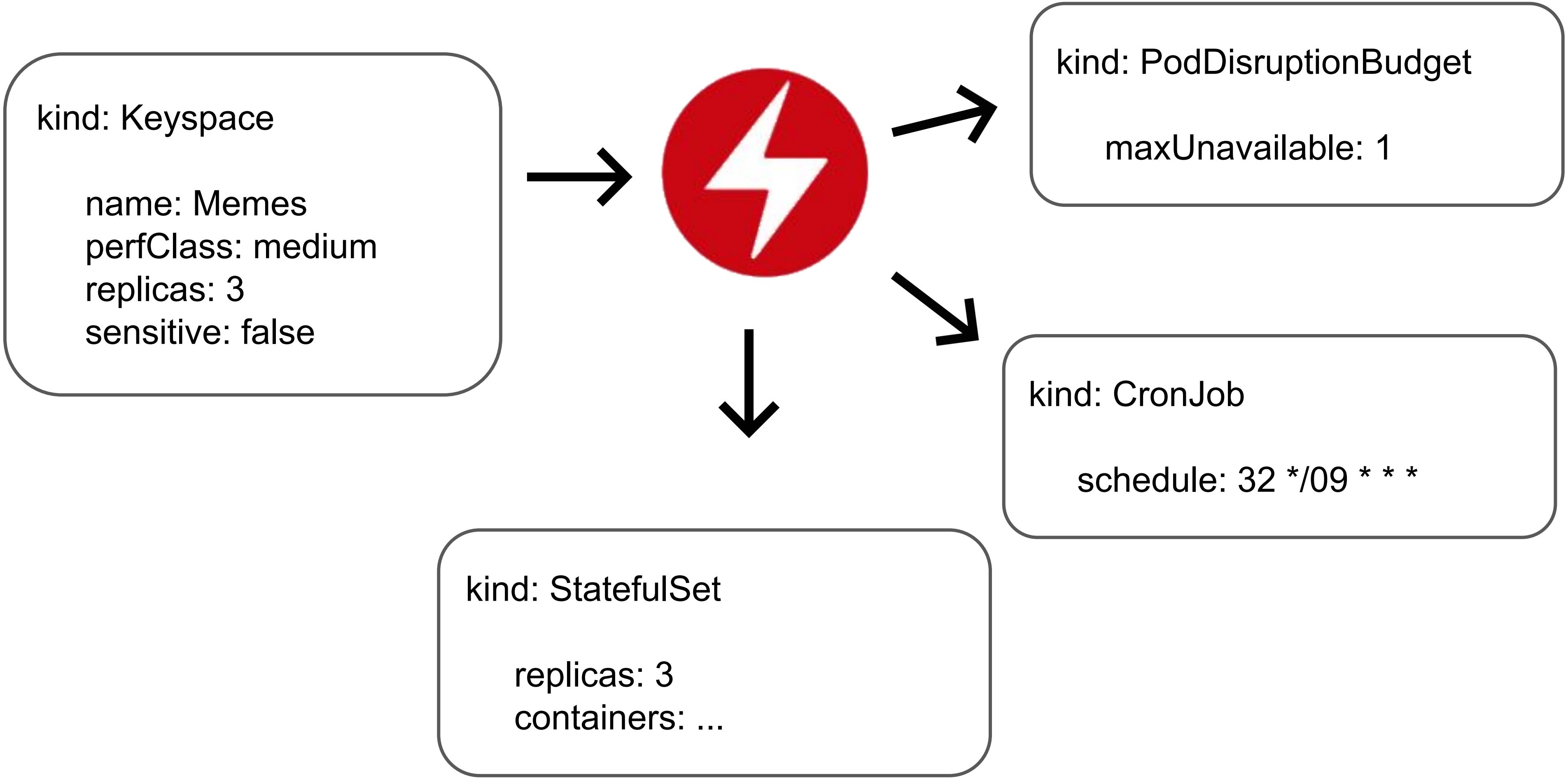
November 03, 2016 • By Brandon Philips

Tags: *announcements* *Operators*

What’s an operator?



Custom Resources



```
kind: Keyspace

    name: Memes
    perfClass: medium
    replicas: 3
    sensitive: false
```

```
kind: PodDisruptionBudget

    maxUnavailable: 1
```

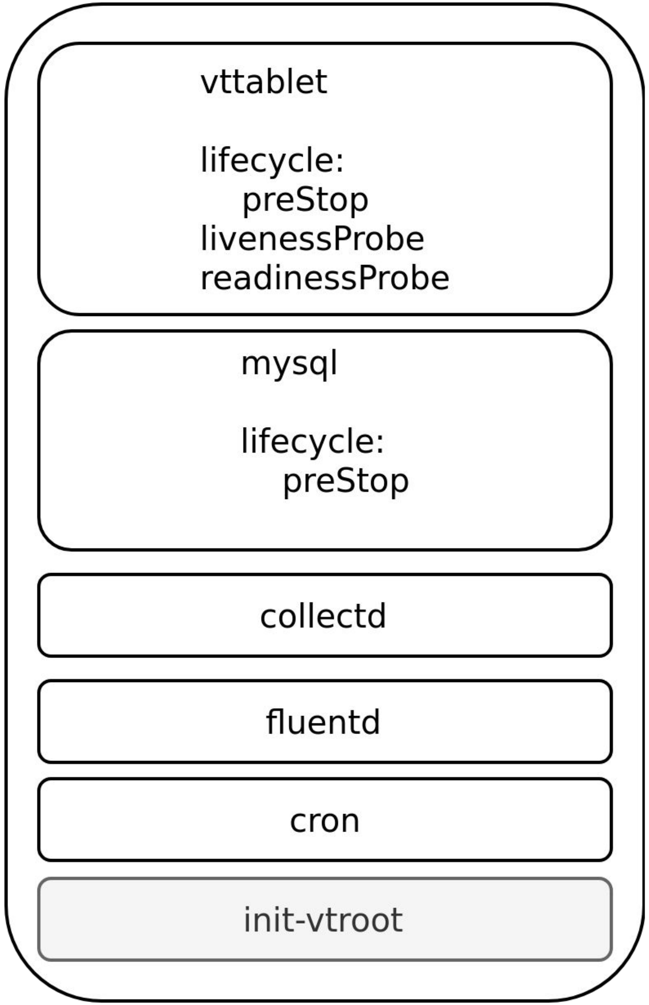
```
kind: CronJob

    schedule: 32 */09 * * *
```

```
kind: StatefulSet

    replicas: 3
    containers: ...
```

Anatomy of a Pod



vttablet

lifecycle:
    preStop
livenessProbe
readinessProbe

mysql

lifecycle:
    preStop

collectd

fluentd

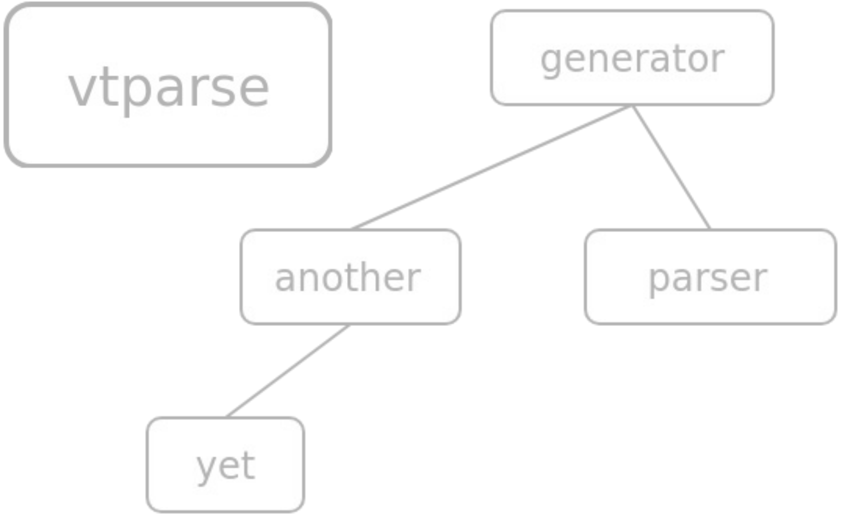
cron

init-vtroot

Lots of other work: VtgatePool CRD

```
kind: VtgatePool

    enableSSL: true
    replicasPerZone: 4
```



vtparse

generator

another

parser

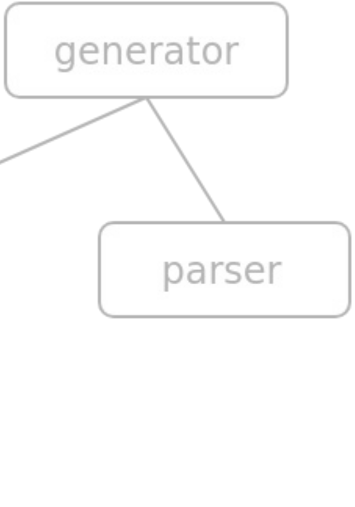
yet

Lots of other work: JDBC driver

```
kind: VtgatePool

    enableSSL: true
    replicasPerZone: 4
```

vtparse



TLS

orchestrator

Lots of other work: TLS certs

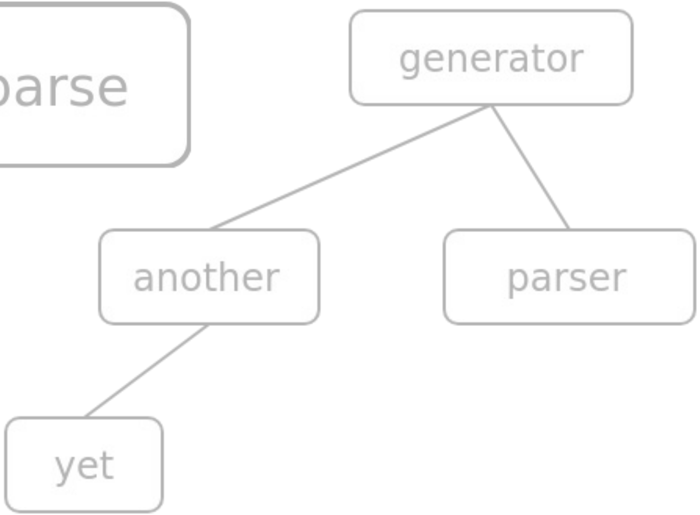
```
kind: VtgatePool

    enableSSL: true
    replicasPerZone: 4
```

Jdbc

TLS

vtparse



orchestrator

Lots of other work: Orchestrator Integration

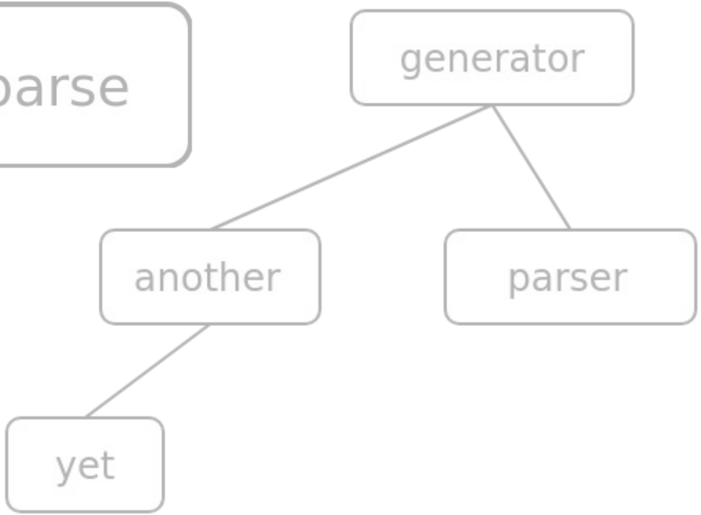
```
kind: VtgatePool

    enableSSL: true
    replicasPerZone: 4
```

Jdbc

TLS

vtparse



orchestrator

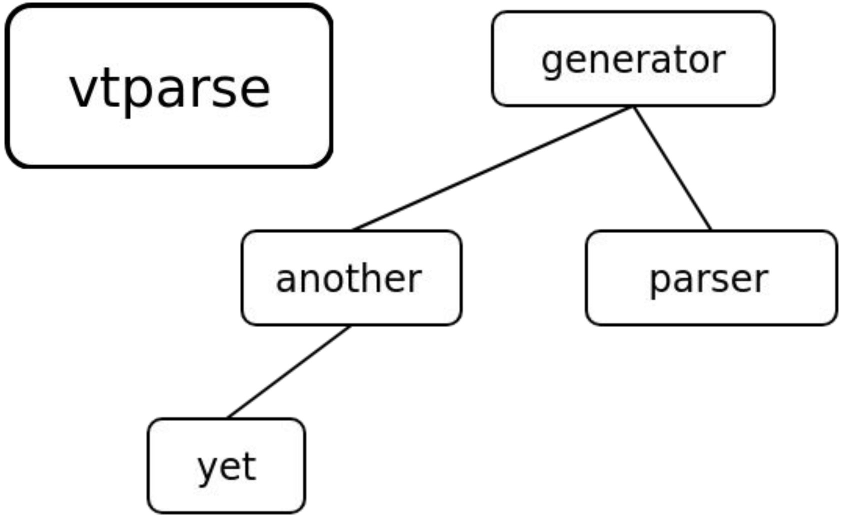
Lots of other work: SQL grammar

```
kind: VtgatePool

    enableSSL: true
    replicasPerZone: 4
```

Jdbc

TLS



vtparse

generator

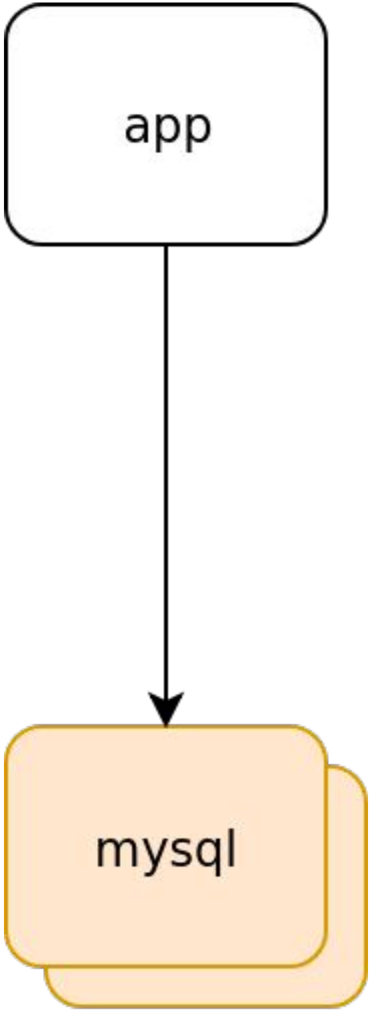
another

parser

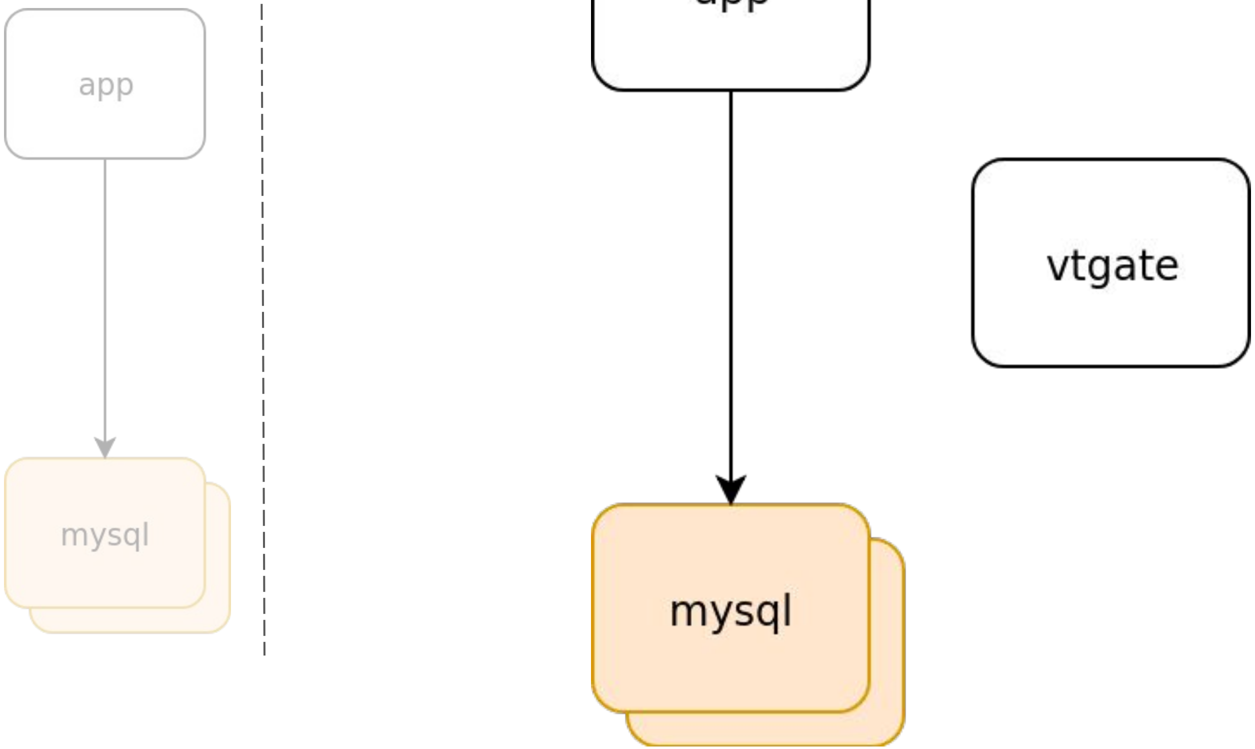
yet

orchestrator

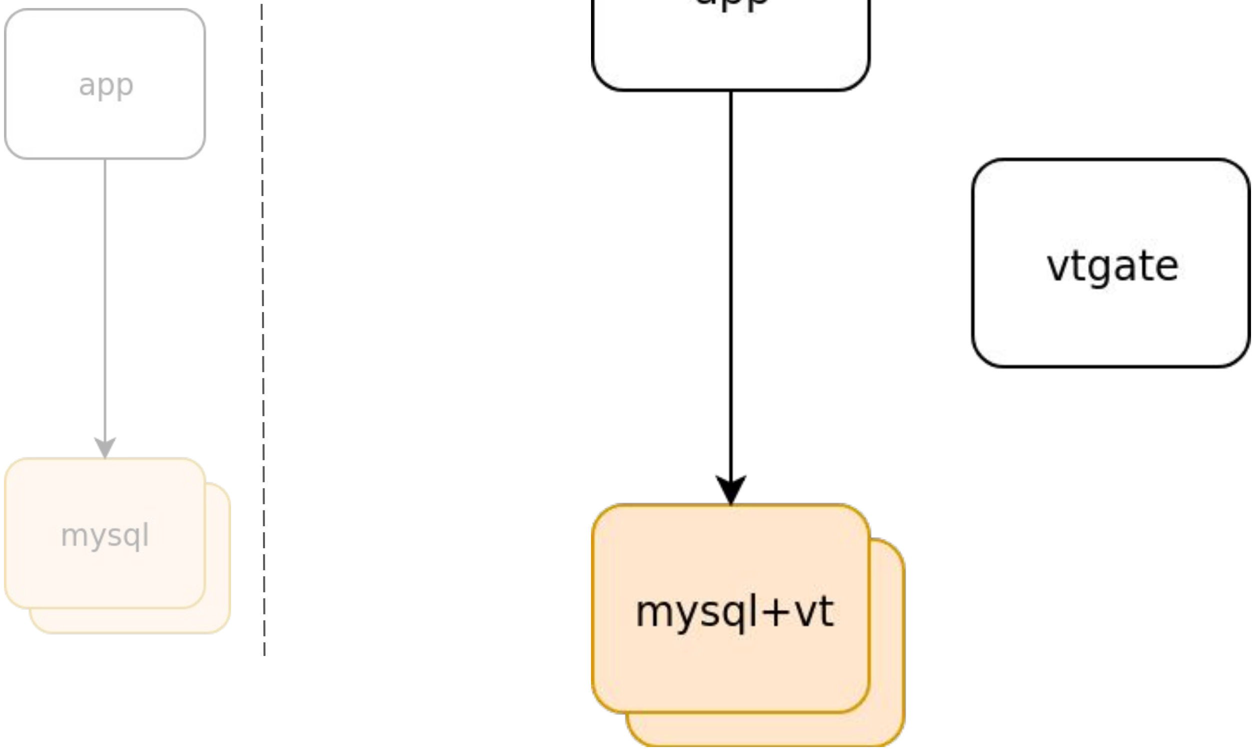
Migration



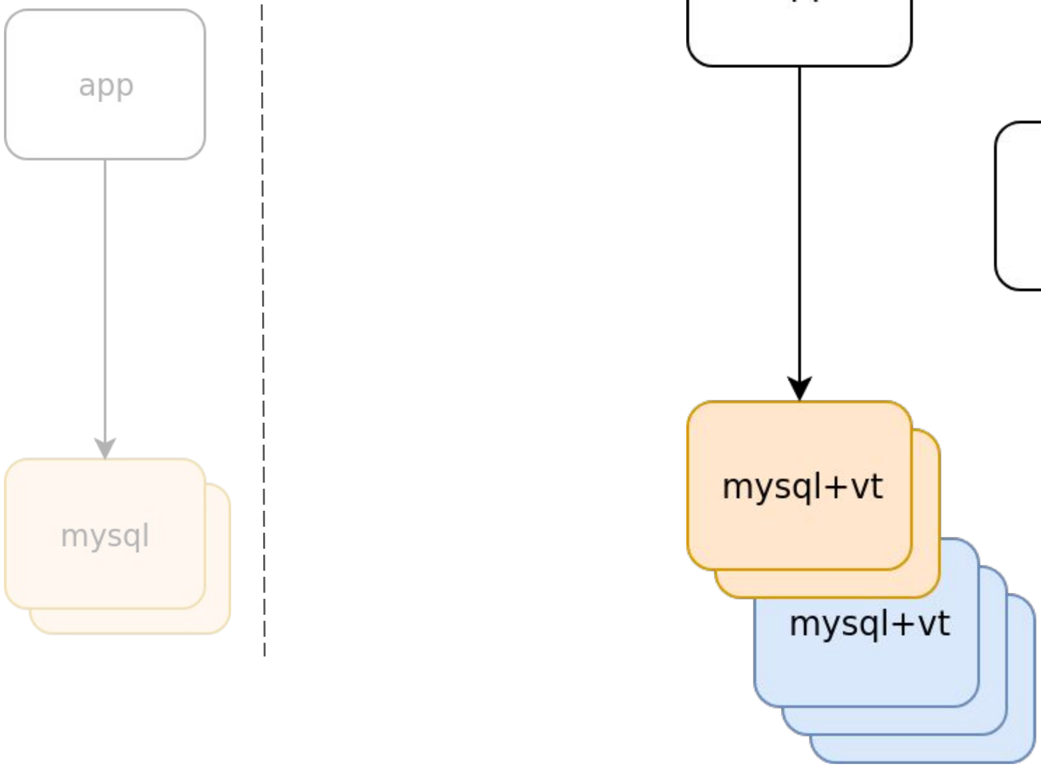
Migration



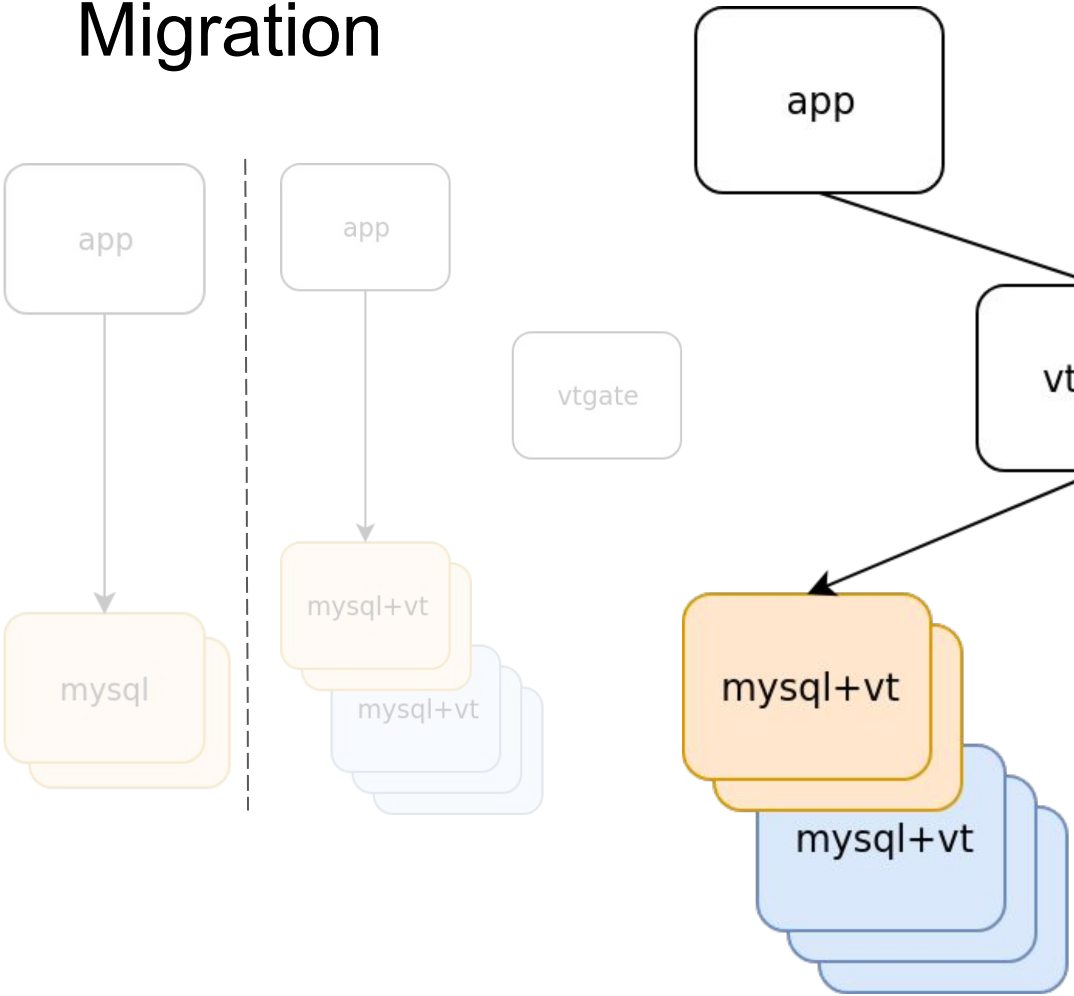
Migration



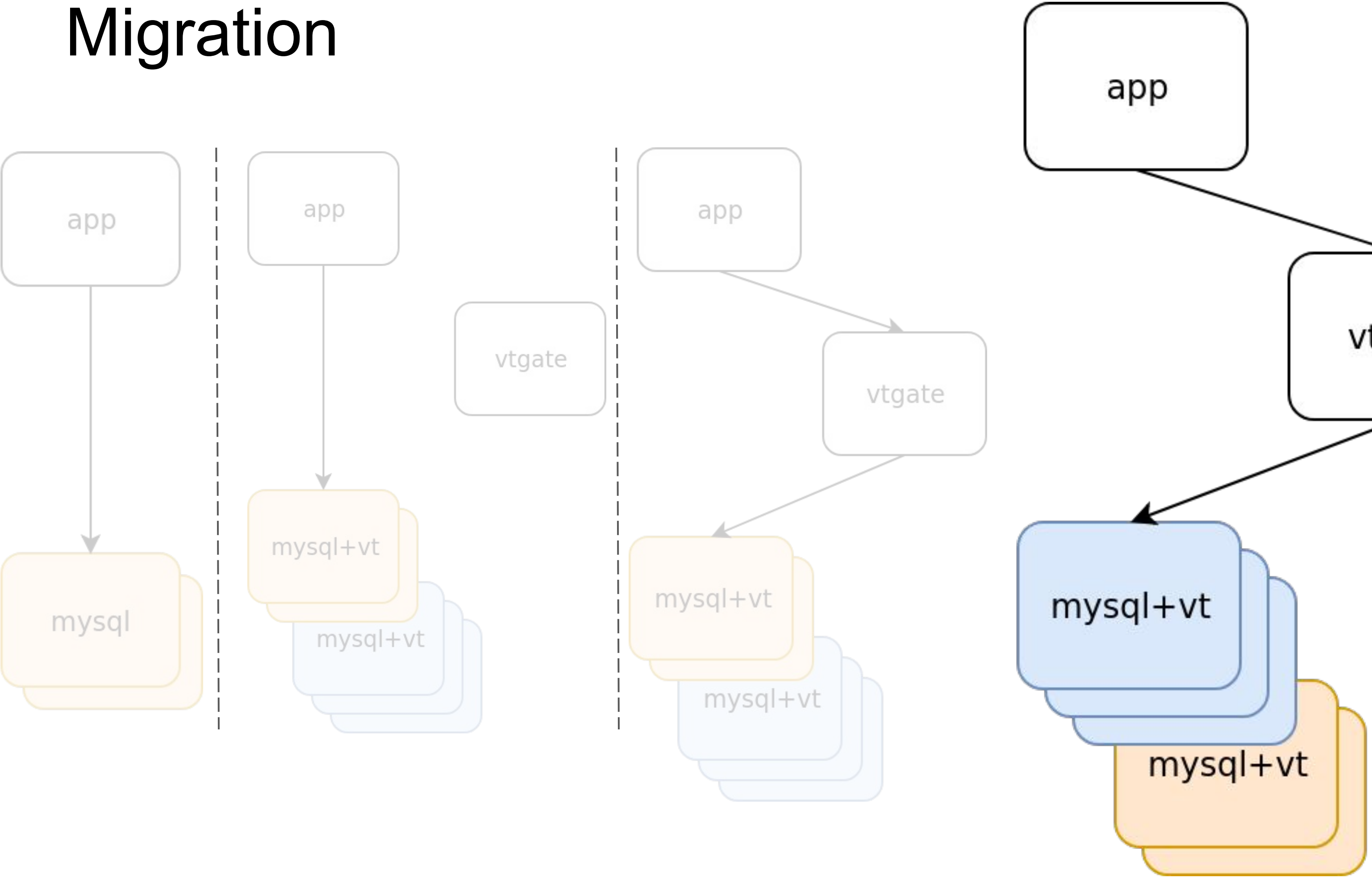
Migration



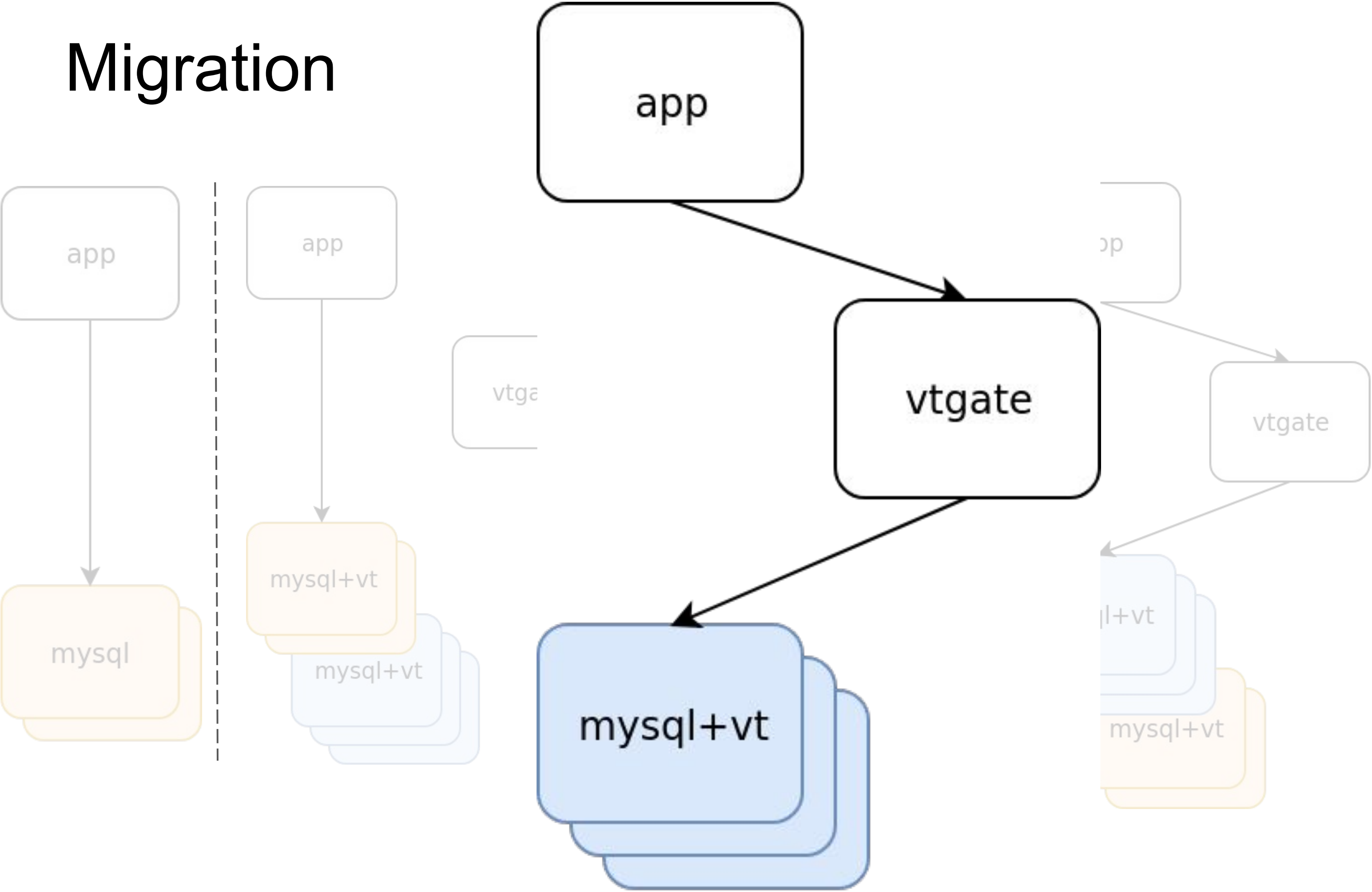
Migration


app
mysql
app
mysql+vt
mysql+vt
vtgate
app
vtgate
mysql+vt
mysql+vt

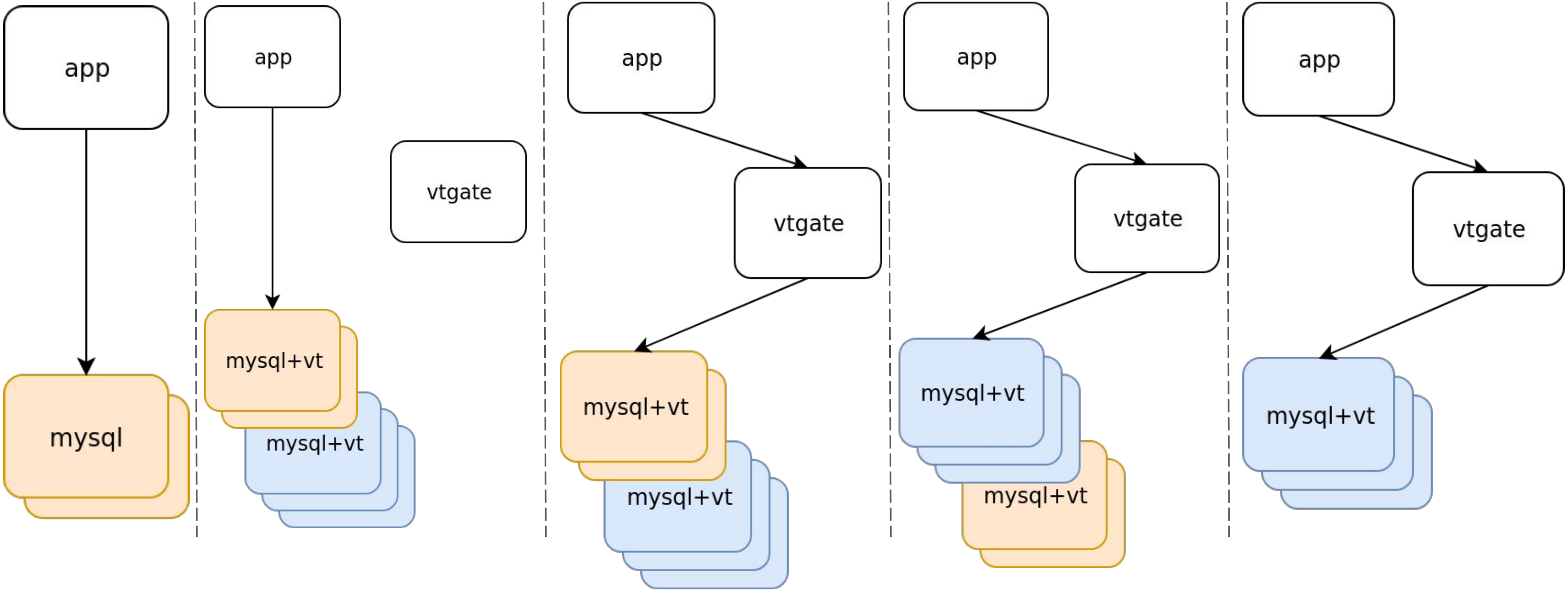
Migration


app
mysql
app
mysql+vt
mysql+vt
vtgate
app
vtgate
mysql+vt
mysql+vt
app
vtgate
mysql+vt
mysql+vt

Migration

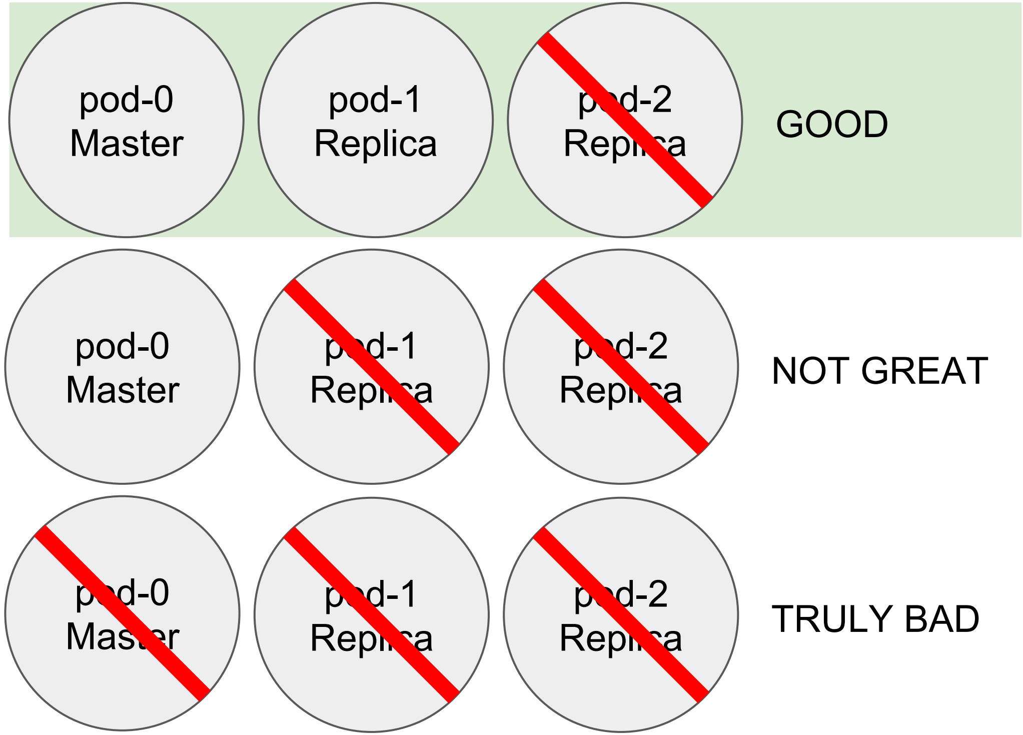

app
mysql
app
mysql+vt
mysql+vt
vtga
app
vtgate
mysql+vt
pp
vtgate
l+vt
mysql+vt

Migration



zero downtime

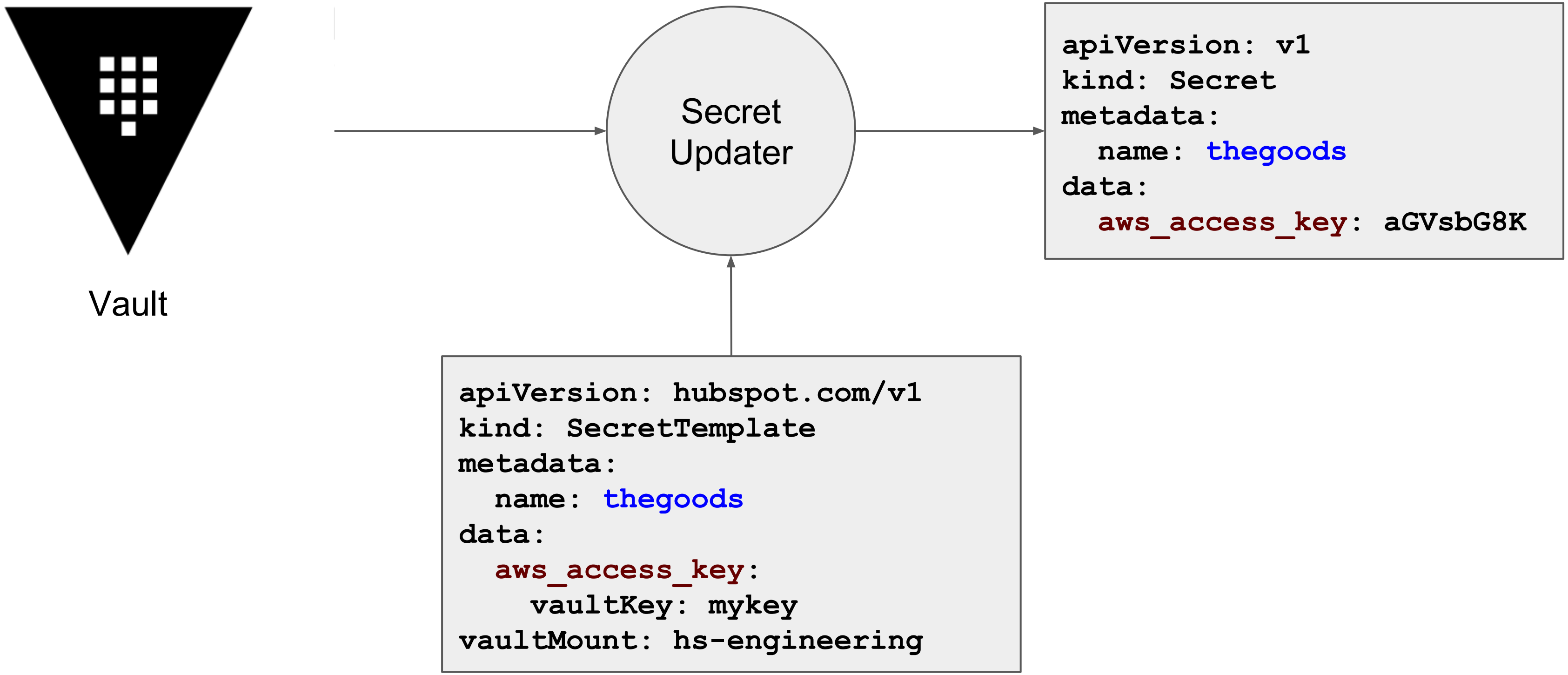
Protecting against cluster maintenance



```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
spec:
  maxUnavailable: 1
  selector:
    matchLabels:
      app: vitess
      component: vttablet
      keyspace: Memes
      role: serving
      shard: "0"
```

DELETION IS NOT EVICTION

Injecting Vault secrets into Pods



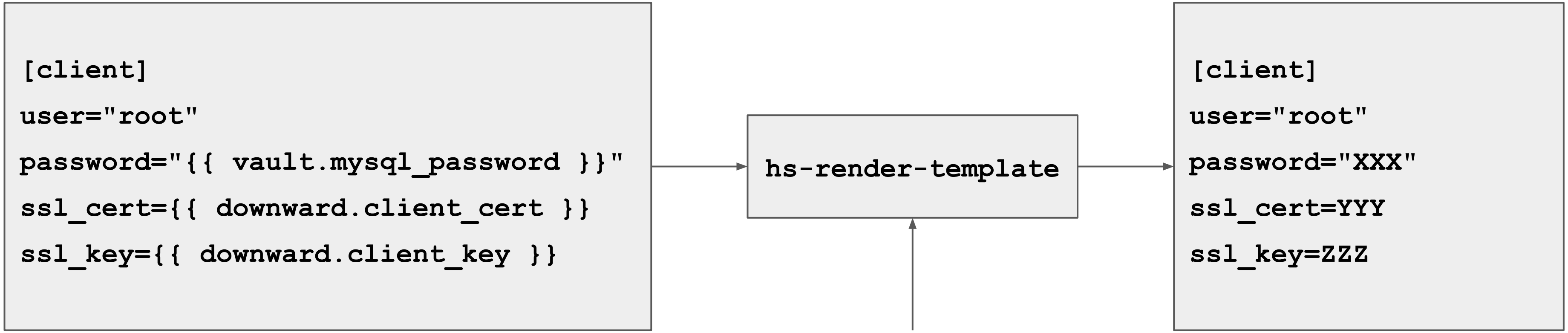
Vault

Secret Updater

```
apiVersion: v1
kind: Secret
metadata:
  name: thegoods
data:
  aws_access_key: aGVsbG8K
```

```
apiVersion: hubspot.com/v1
kind: SecretTemplate
metadata:
  name: thegoods
data:
  aws_access_key:
    vaultKey: mykey
vaultMount: hs-engineering
```

Parameterizing configuration in Docker images



```
[client]
user="root"
password="{{ vault.mysql_password }}"
ssl_cert={{ downward.client_cert }}
ssl_key={{ downward.client_key }}
```

/etc/templates/my.cnf.tmpl

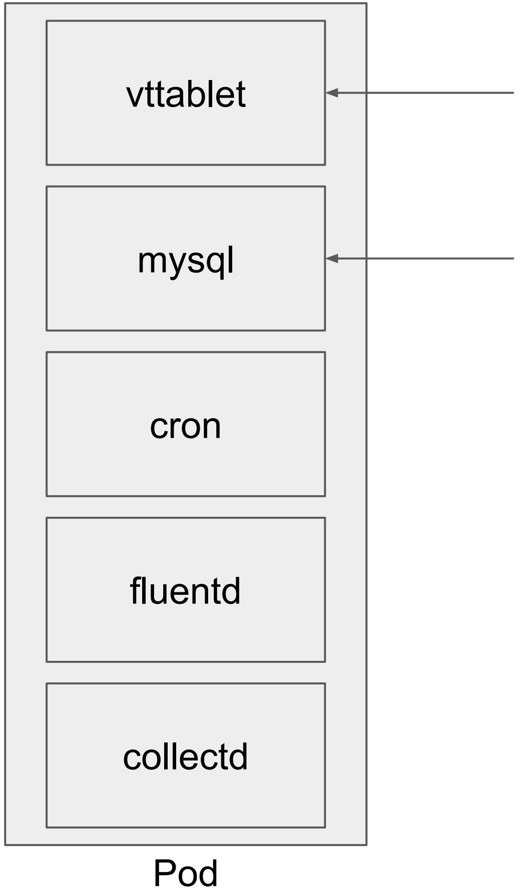
`hs-render-template`

Downward API
Node metadata
YAML files
Env vars

```
[client]
user="root"
password="XXX"
ssl_cert=YYY
ssl_key=ZZZ
```

/etc/my.cnf

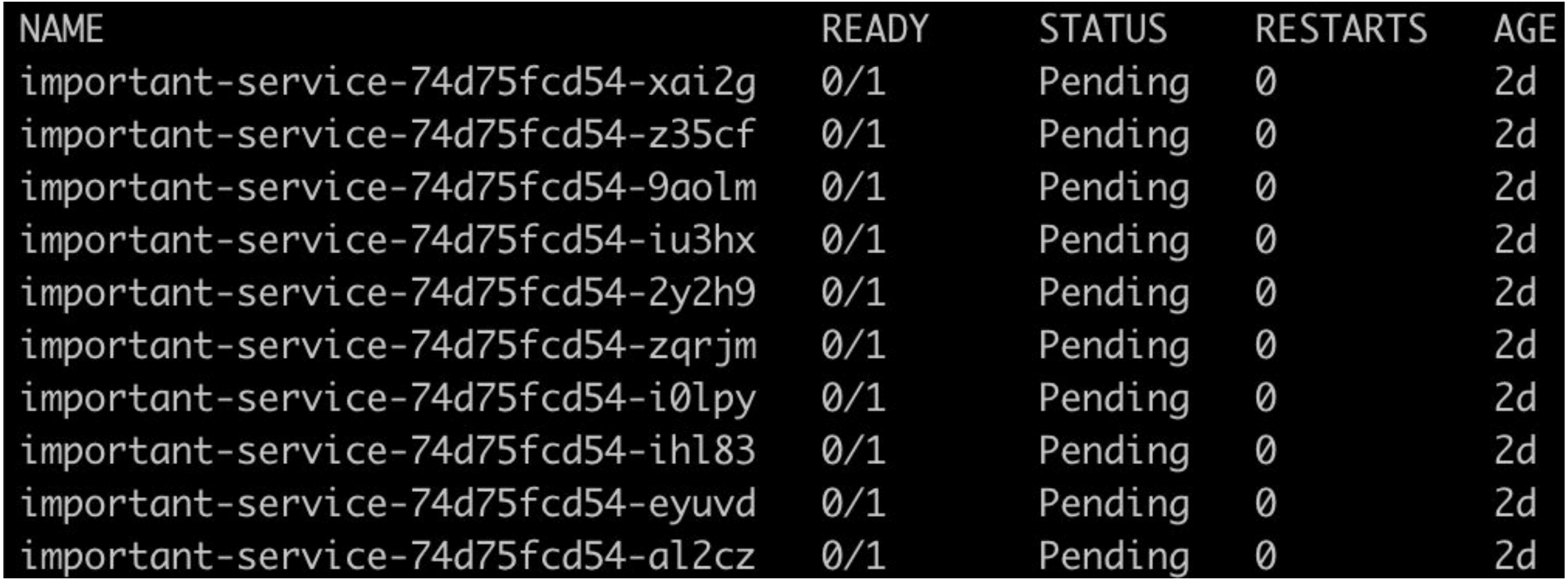
Terminating Pods gracefully



```
if [[ $TABLET_TYPE == “master” ]]; then
  vtctl PlannedReparentShard ...
fi
```

```
while tablet-is-running; do
    sleep 5
done
```

Tapping into (nearly) infinite resources



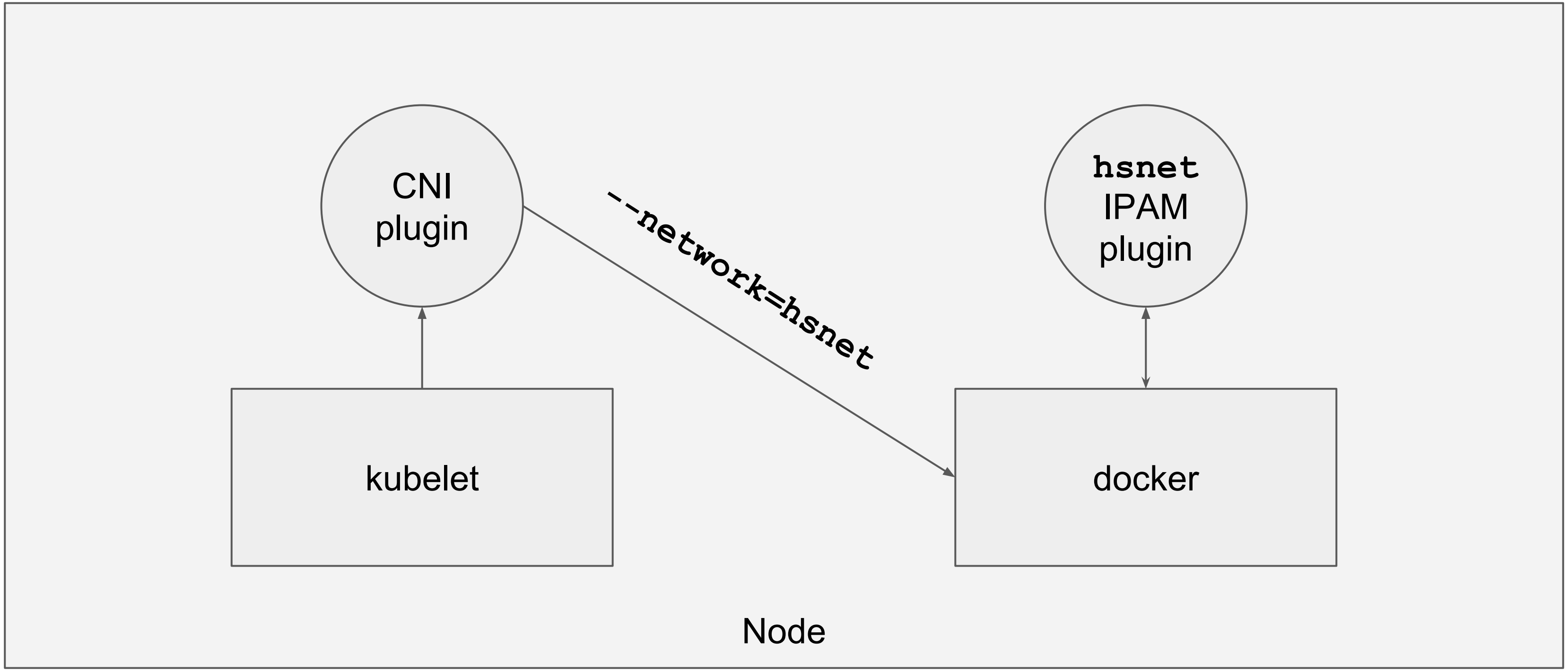
```
NAME                                   READY     STATUS    RESTARTS   AGE
important-service-74d75fcd54-xai2g     0/1       Pending   0          2d
important-service-74d75fcd54-z35cf     0/1       Pending   0          2d
important-service-74d75fcd54-9aolm     0/1       Pending   0          2d
important-service-74d75fcd54-iu3hx     0/1       Pending   0          2d
important-service-74d75fcd54-2y2h9     0/1       Pending   0          2d
important-service-74d75fcd54-zqrjm     0/1       Pending   0          2d
important-service-74d75fcd54-i0lpy     0/1       Pending   0          2d
important-service-74d75fcd54-ihl83     0/1       Pending   0          2d
important-service-74d75fcd54-eyuvd     0/1       Pending   0          2d
important-service-74d75fcd54-al2cz     0/1       Pending   0          2d
```

Hacking around bugs

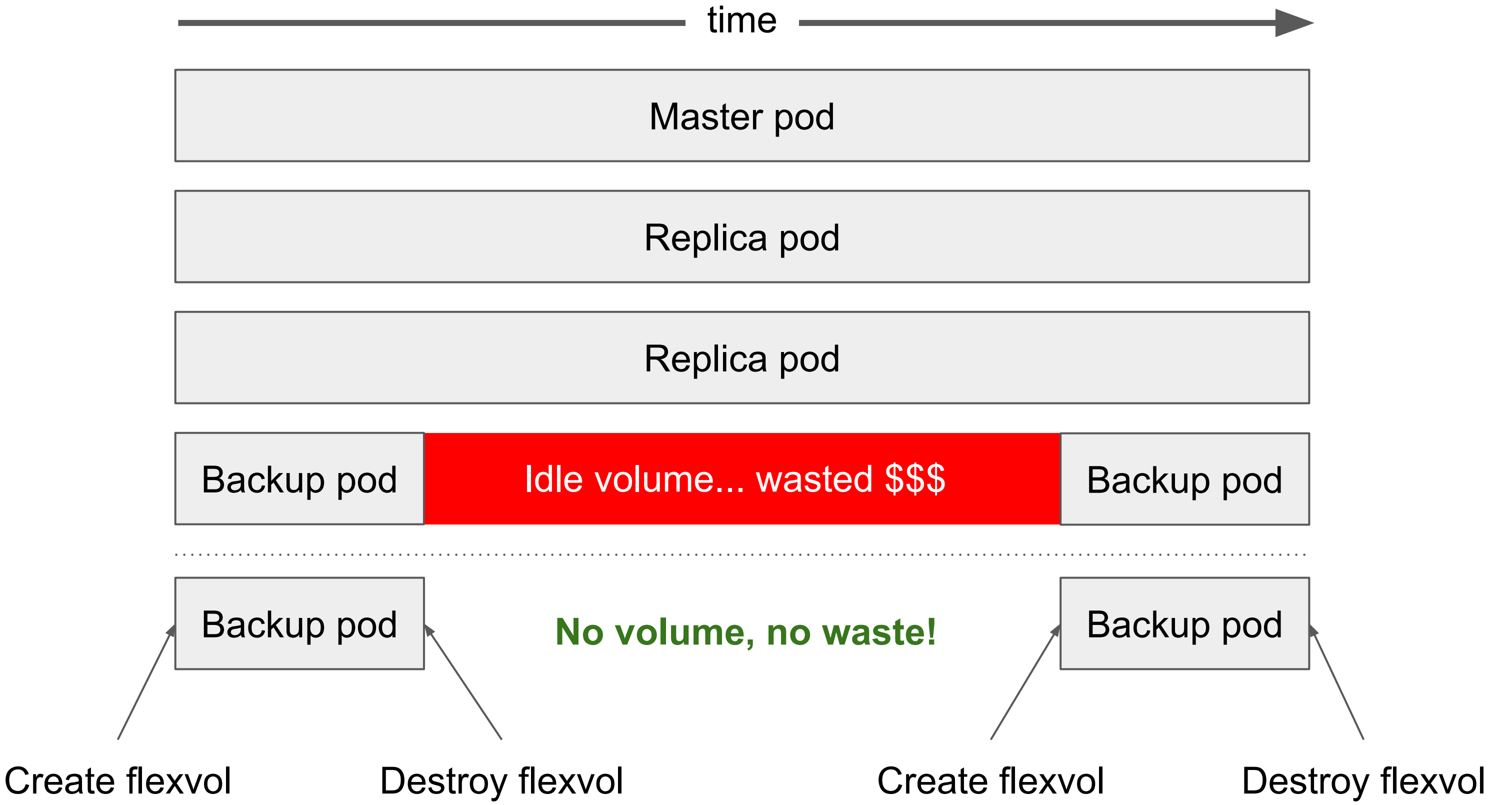
```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gp2-xfs
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
  fsType: xfs
```

Fixed in #45345

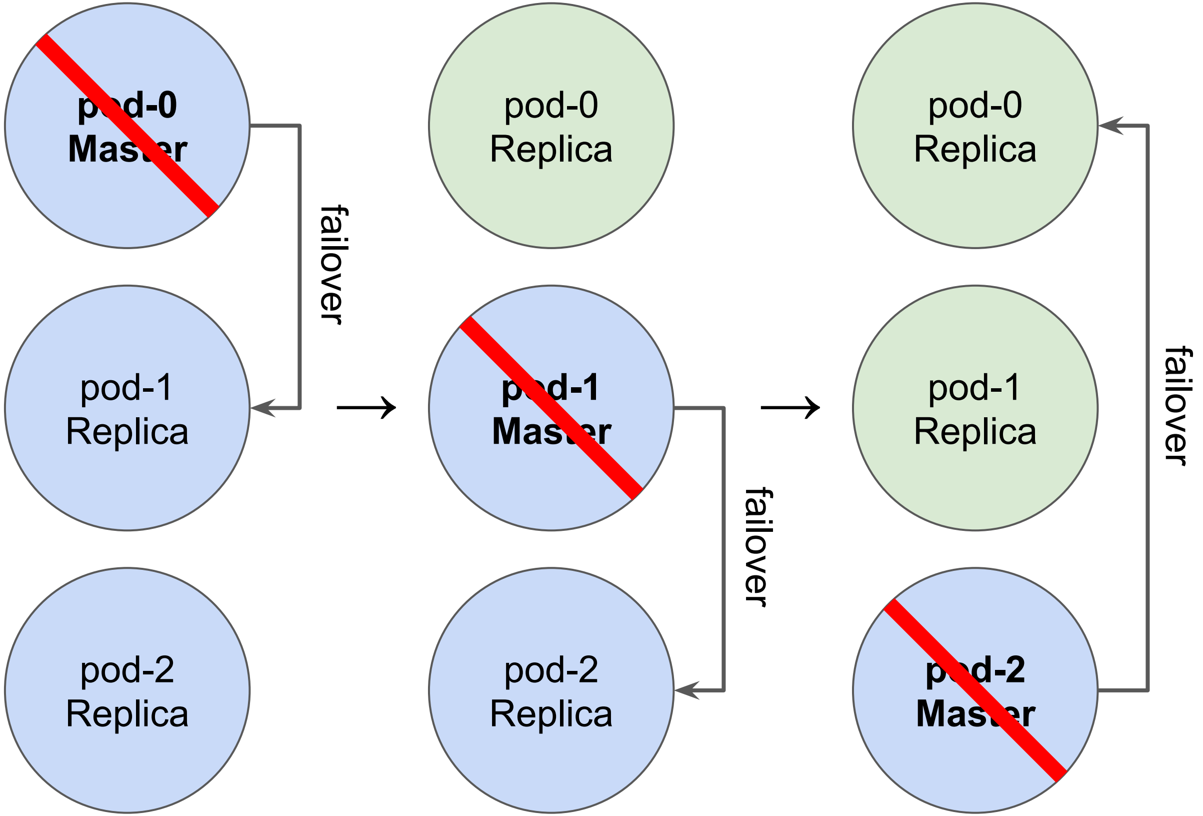
Protecting against rapid IP address reuse



Optimizing Vitess Backups

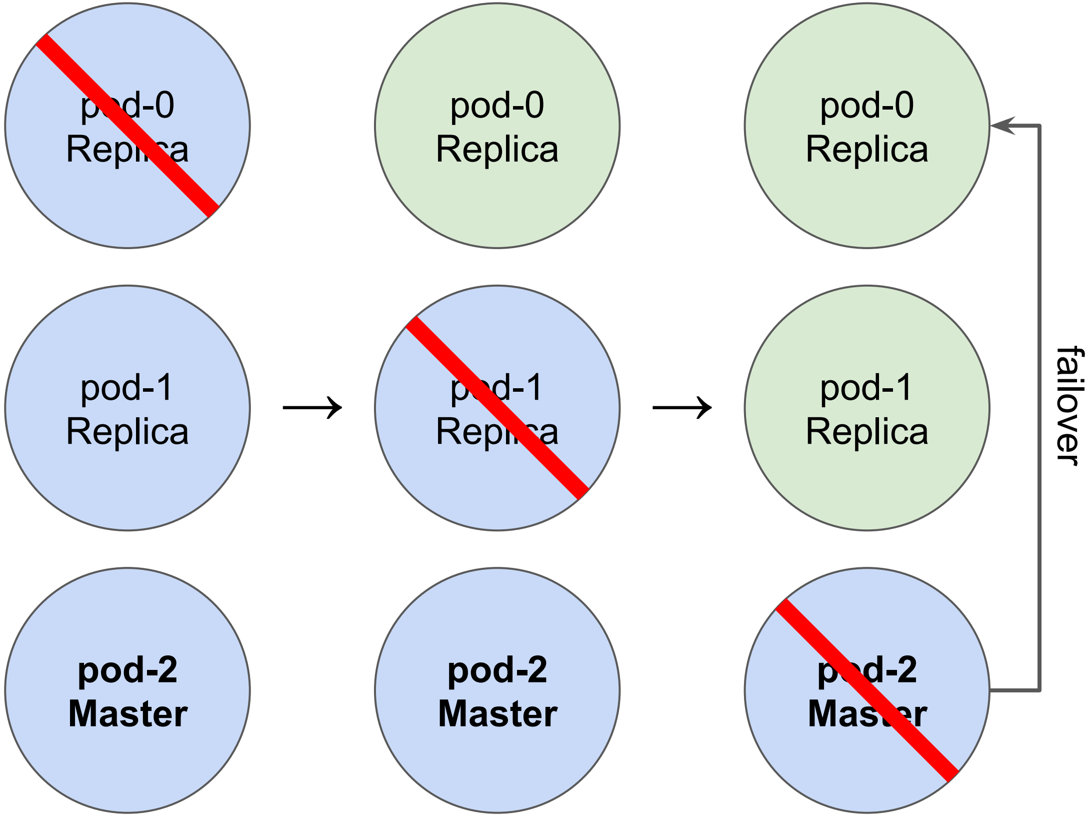


Controlling StatefulSet rollouts



(worst-case scenario)

Controlling StatefulSet rollouts

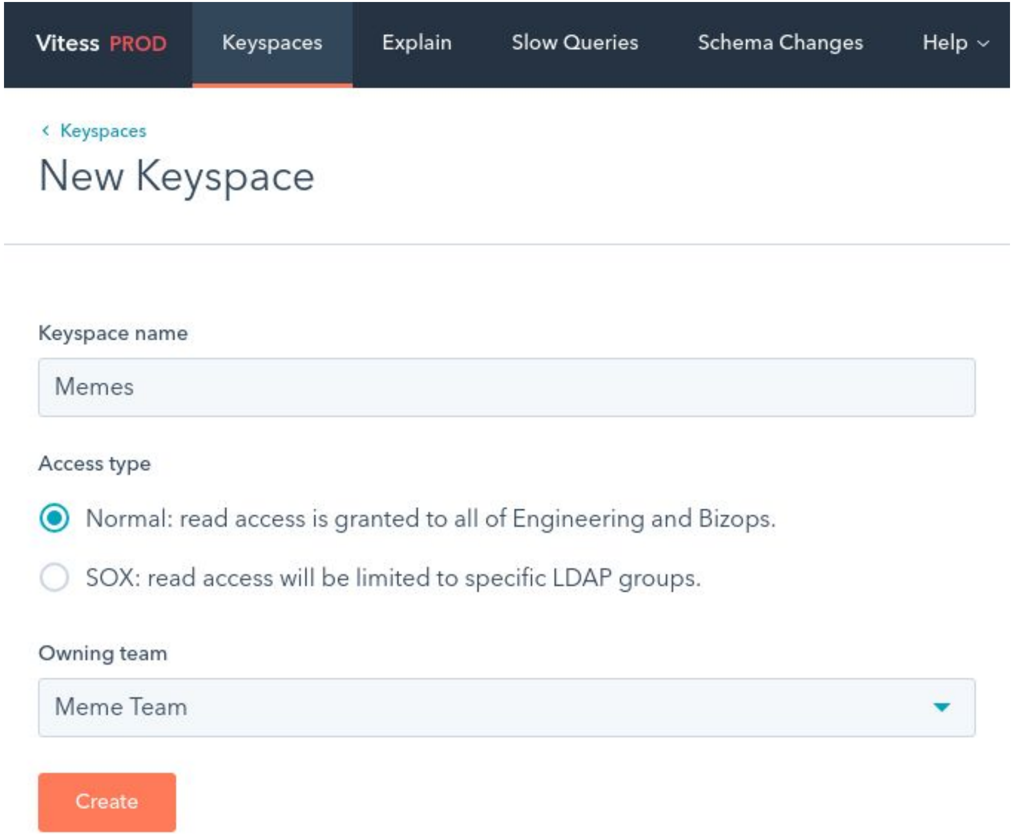


DeletePropagationOrphan FTW

Lessons learned: It’s a transplant

COS Certified. Highly Recommended.
COS Certified. Highly Recommended.

Results: DBaaS



New Keyspace

Keyspace name

Memes

Access type

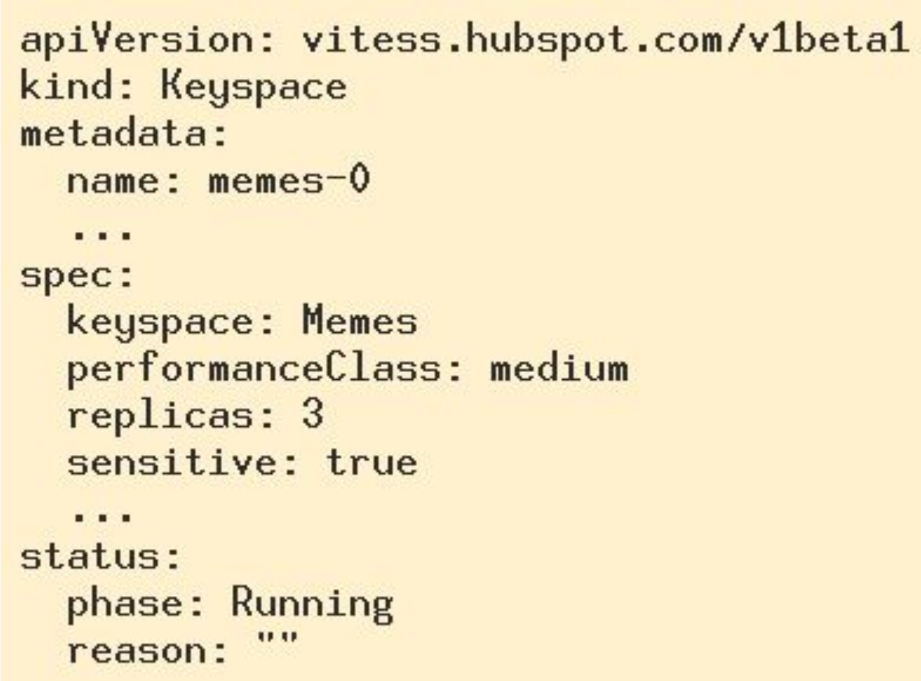
● Normal: read access is granted to all of Engineering and Bizops.

○ SOX: read access will be limited to specific LDAP groups.

Owning team

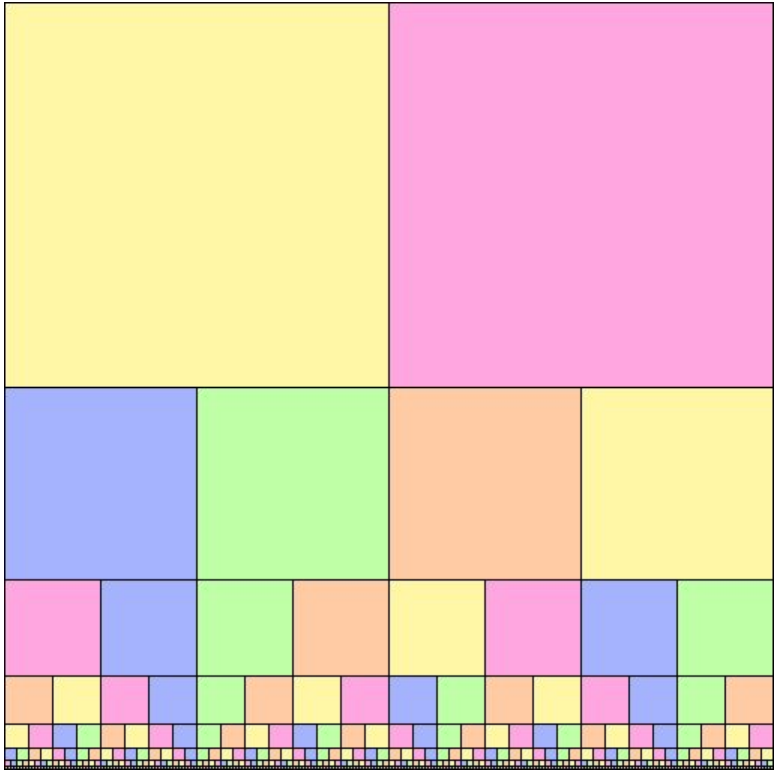
Meme Team

Results: Trivial Read Scaling

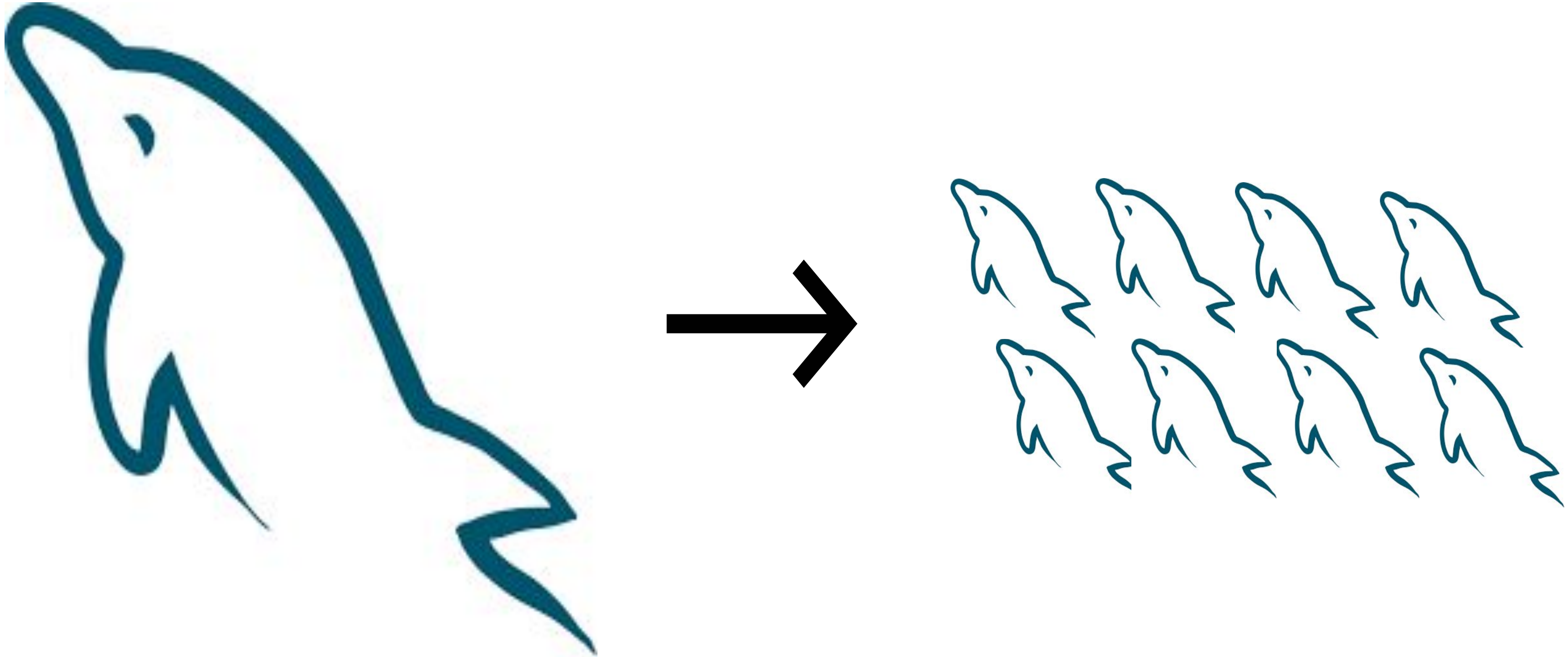


```
apiVersion: vitess.hubspot.com/v1beta1
kind: Keyspace
metadata:
  name: memes-0
  ...
spec:
  keyspace: Memes
  performanceClass: medium
  replicas: 3
  sensitive: true
  ...
status:
  phase: Running
  reason: ""
```

Results: Bin Packing



Results: Sharding



Results: Automated Impairment Handling

Results: Finally I can sleep

Thanks and Q&A

vitess.io

hubspot.com