Moving Shopify Core to the 🌩️

Scott Francis
scott.francis@shopify.com
@planetscott
Scott Francis
Senior Production Engineer Lead

Location: Ottawa - Elgin, Canada
Discipline: Engineering

Joined: about 5 years ago
97.37% of Shopifolk joined afterwards
80k requests per second peak

600k+ active merchants

$26b gmv (2017)

800+ developers

40+ daily deploys
Part One: Pre-
Sharding - 2013

Web 1 → Database Shard 1 (1/2 shops)
... → ...
Web N → Database Shard 2 (1/2 shops)
Job 1 → ...
Job N → ...

Database Shard 1
(1/2 shops)

Database Shard 2
(1/2 shops)
A (Shopify) pod is an isolated unit of one or more shops.
Pod Routing

Data Centre 1
23.227.38.64

Pod 1
Pod 2
Pod 6
Pod 8

Pod 3
Pod 5
Pod 7
Pod 4

Data Centre 2
23.227.38.64
Part Two: 🌧️
Building for the long term.
Compliance
Performance
Consistency
Execution
“We’re going to move to the cloud one shop at a time.”

- Nameless Shopify Production Engineer
Shop Mover

• Moving a shop is NOT trivial

• Data migration uses Ghostferry

• Randomized shop selection
Shop Mover

1. New Shop Move
2. Perform Sanity Checks
3. Begin binlog Copy
4. SELECT and INSERT
5. Wait for binlog
6. Select Shop Writes
7. Switch Shard ID
8. Complete Replication
9. Stop Shop Writes
10. Wait for Jobs
11. Allow Shop Writes
12. Shop Move Complete
active shops: 2017-08-17

Cloud Shards

Number of shops

Shard ID

1. The chart represents the number of active shops as of 2017-08-17.
2. The x-axis is labeled "Shard ID," and the y-axis is labeled "Number of shops.
3. The chart shows a bar graph with bars indicating the number of shops across different shard IDs.
4. The highest number of shops is around the Shard ID 0, with a significant drop afterward.
5. The chart suggests a distribution of shops across various shards, with a concentration in the first shard.
spy failover shopify pod 1 to ash

spy 12:29 PM

Failing over pod 1 to ash.

Please monitor the failover in #pod-failovers and datadog:

pod 1: 

Tommy is running Failover a pod for activefailover/production TARGET=ash POD_ID=1 (logs)

Failover a pod for activefailover/production succeeded in 2m38s
Activefailover

1. Start Pod Failover
2. Perform Sanity Checks
3. Update Pod Location
4. Pause Requests
5. Failover MySQL
6. Pod Failover Complete
7. Transfer Jobs
8. Clean Up
9. Resume Requests
10. Perform Sanity Checks
11. Update Pod Location
12. Pause Requests
13. Failover MySQL
14. Pod Failover Complete
15. Transfer Jobs
16. Clean Up
17. Resume Requests
Problems
Fear, Uncertainty, Doubt

- Database Performance
- Kubernetes
- Cloud Infrastructure
Scaling Kubernetes Clusters
Shopify Core k8s v1
k8s Scaling Problems

• How to communicate across clusters?? 😞

• Running out of routes??

• Cross-cluster service discovery??
k8s Scaling Solutions

• Running out of routes??
  • IP Aliasing

• Cross-cluster service discovery??
  • External kube-dns ILB
Shopify Core k8s v2
kubectl is a really big
kubernetes-deploy

- Watches changes to make sure they’re successful
- Pre-deploys dependent resources
- Runs tasks at the beginning of the deploy
kubernetes-deploy

```
$ tree

  ├── configmap-data.yml
  │    └── redis.yml
  │    └── secrets.json
  │    └── unmanaged-pod.yml.erb
  └── web.yml.erb

0 directories, 5 files
```

```
$ tree

```
Lessons Learned
Maintaining two separate infrastructures is really hard.
Lift and Shift, or: don’t do too much at the same time
Iterate
Automate
Thanks!

p.s. we’re hiring
Links

- **Ghostferry**: https://github.com/Shopify/ghostferry
- **kubernetes-deploy**: https://github.com/Shopify/kubernetes-deploy
- **Shipit**: https://github.com/Shopify/shipit-engine
- **Chat Ops at Shopify**: https://engineering.shopify.com/blogs/engineering/implementing-chatops-into-our-incident-management-procedure
Image Credits

- https://i.ytimg.com/vi/u19mQIqEmE/maxresdefault.jpg