

Experience with taking over a service running ownCloud on virtualized infrastructure, and surviving the experience

SWITCH

Greg Vernon

greg.vernon@switch.ch

Kraków, 29 January 2018



SWITCH

- We are the Swiss National Research and Education Network.
- We network the Institutes of Higher Education and Research to each other, and the rest of the world.
- We provide additional services such as Federated Authentication, Video, and File Sharing to our Educational customers.
- We manage the Top Level Domains for Switzerland (.ch) and Liechtenstein (.li).
- We provide SWITCH-CERT security service.

Our customers



SWITCH community

- Swiss universities on tertiary level (academic sector) and their research institutions



Extended community

- Other organizations involved in research or education



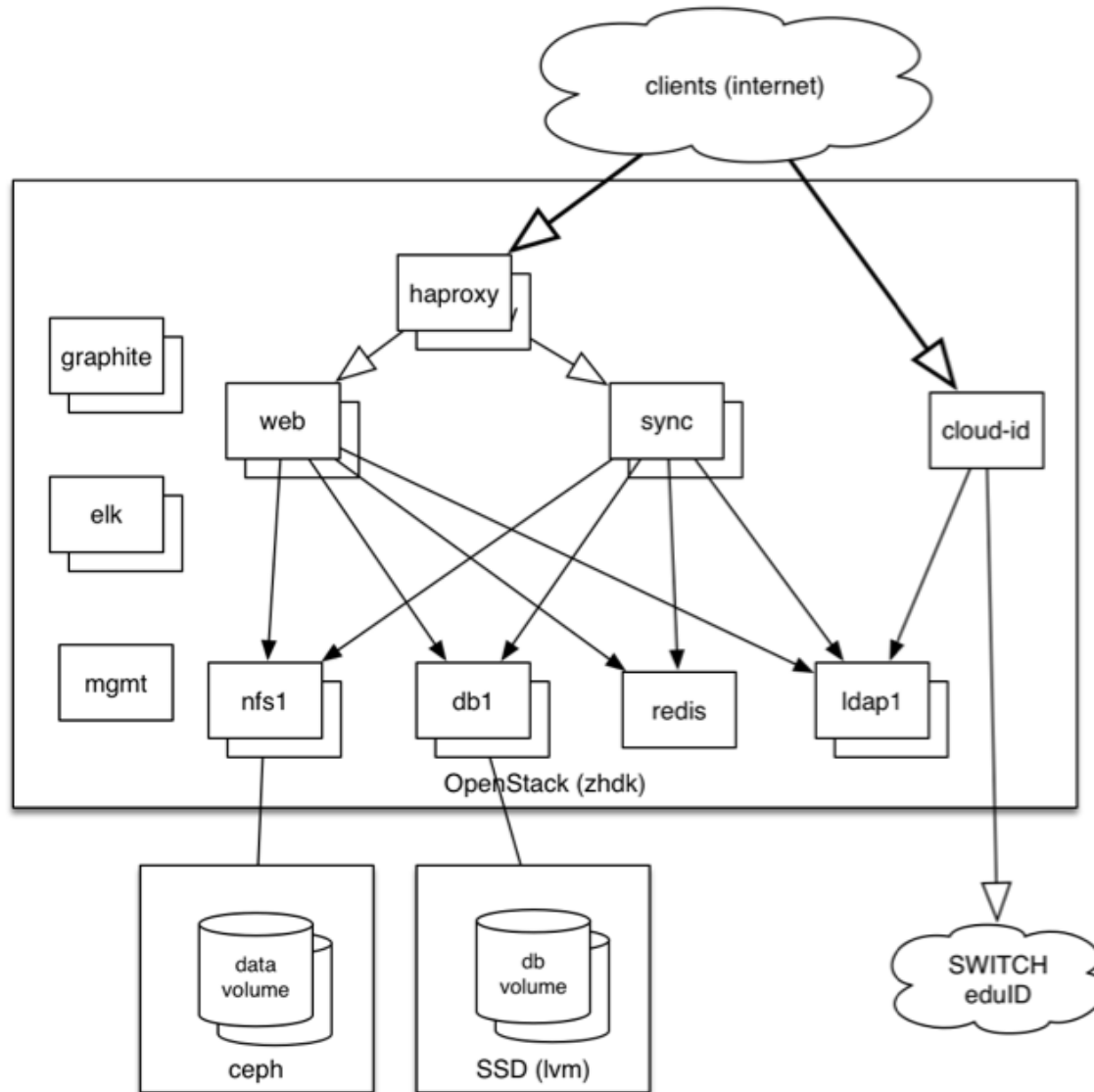
Commercial customers

- Registrars of .ch- and .li-Domain-Names, Swiss financial institutions, research-related industry and government

SWITCHdrive

SWITCHdrive is our branded ownCloud offering. We have the following:

- About 30,000 Users
- 125,000,000 files
- 125,000,000 rows in our oc_filecache table
- 3 Mariadb servers in a Galera cluster
- 9 Apache Servers(4 Sync/4 Web/1 Management)
- Redis
- 3 LDAP Servers
- 5 NFS servers running atop CEPH
- 2 HAproxy load balancers
- Monitoring (Graphite, ELK)
- Runs atop SWITCHengines, our OpenStack offering
- Most services are Docker containers



How I got involved

- I've run big production systems
- With web servers
- and databases
- And I love to troubleshoot things



Ansible

- We use Ansible to manage our infrastructure.
- We have to use something like this.
- It works well, but all of our playbooks are a complex collection.
- If you use Ansible, stick to the best practices when setting up your directory structure. We pretty much had, so it made my life much easier.

Testing environment

- We have a test environment similar to our production environment.
- Of course, with less resources
- When our ansible scripts work with the test environment, they almost always work with the production environment

- We had just upgraded to ownCloud 9.0 from 8.2
- Our DB load was very high
- We decided to add more DB nodes to handle it
- The system needed continual monitoring as:
 - things were hanging
 - storage volumes would grow out of control
 - Cron jobs would hang, or many would queue up.

DB problems (looping, corruption, performance)

- We had to add a number of extra DB read nodes to our Galera cluster, as our DB load became very high
- We had an issue where there was an infinite loop in the oc_filecache structure when changes were being propagated.
- There were upserts that weren't quite working right
- The high load and the looping problem is the likely cause of corruption that we were seeing in the DB
- Adding a missing index made things **MUCH** better! (Come to my talk tomorrow)

Broken cron and how we solved it

- We noticed that our ownCloud cron jobs were backing up.
- So many, that when the semester started, we would not be able to keep up.
- Cron would go into a 'scanfiles' loop
- ownCloud provided logging patches, and we started stack traces whenever we saw cron going into 'scanfiles' loops
- We also dropped the CardDAV cron job, which helped quite a bit.
- After adding a missing index, we have not noticed the issue again.

Storage

- Our storage is a bit unusual
- We use NFS servers that are built from Ceph volumes.
- We need many servers, and even more volumes.
- But we can change storage backends transparently.
- Our storage is evolving, and is continually improving.
- It has challenges, and learn more at my talk on Wednesday!

Meteors

- A meteor is what I named our volumes filling up out of control.
- The grafana storage graph showed a line sharply going down, a bit like a missile in missile command
- Many of the files causing this were Outlook .pst files, or database type files that would cause versions to be generated often.
- The cron jobs `_should_` take care a lot of this, but they were getting hung up, as mentioned earlier, fixing the DB problems has fixed this.

NFS breaking/Ceph

- Our storage is currently five NFS servers, with about 20 2TB Ceph volumes each that are exported out to all of the web/sync servers
- We suspected our bottleneck was Ceph, as it was slower in earlier versions, but it isn't so slow any longer
- We used fio to break our NFS servers, showing that Ceph wasn't our main bottleneck, but NFS seems to be, or perhaps XFS is?
- The problem now, is how do we improve our NFS performance, while still preserving data integrity. We're still working on this.

MoveUserDirs

- With 5 NFS servers having 20 2 TB volumes, these volumes will eventually fill up.
- Some users won't use much, others will use a lot
- So, we move users from full to empty volumes.
- Issues:
 - Trash files aren't in the quota, so sometimes the users have WAY more data than we think they do.
 - The script doesn't yet account for the case where all users have used up their quota
 - OpenStack doesn't handle server keys automatically, so we always forget to check these until the script stop working.

Load balancers and Keepalived

- We use HAProxy as our load balancer in front of the web/sync servers.
- When we enabled IPv6, we happened into a bug where restarting the DHCP serving the load balancer would cause one of the Keepalived daemons to hang.
- As a temporary fix we assigned hardcoded IP addresses to the Virtual IP addresses in Keepalived.

8192 errors

- We had a problem where users were complaining that they would get an error: “Bad Request (expected filesize SOME_NUMBER got 8192)”.
- This is an error you get when a network connection drops, for example, when I close my laptop.
- It only happened to some sync users.
- And only sometimes....
- It turned out that the logs would fill up, logrotate would delete them, and then all would be good.
- So...

Log files filling up

- We have ownCloud set up to do quite a bit of logging. We had stored our logs on ramdisk to make things faster.
- The problem with storing logs in the ramdisk is that when you run out of space there, you're also out of ram.
- We solved this by running logrotate hourly and compressing the rotated logs

MaxScale

- We install MaxScale on our Web/Sync/Management servers to enable the read/write split to our DB cluster.
- It mostly works fine.
- But sometimes it starts to favor a server.
- New version licensing is not attractive to us.
- We are thinking that ProxySQL might work better for us.



Summary

- Modern RDBMSs are actually pretty amazing.
 - You can put them in Docker containers and they actually work quite well.
- Keep track of what's happening with your system, as it will change, sometimes greatly, over time.
- Storage is a challenge.

Questions

Email: greg.vernon@switch.ch