

sciebo ng

University of Münster

WWU IT, Röntgenstr. 7-13, 48149 Münster

## Sciebo Site Report

- ▶ started in 2015 for NRW

## Sciebo Site Report

- ▶ started in 2015 for NRW
- ▶ serving branded ownCloud to about 40 academic institutions

## Sciebo Site Report

- ▶ started in 2015 for NRW
- ▶ serving branded ownCloud to about 40 academic institutions
- ▶ 235k registered users (including recently expired accounts, 6 month time limit)  
(roughly stable since last CS3)

## Sciebo Site Report

- ▶ started in 2015 for NRW
- ▶ serving branded ownCloud to about 40 academic institutions
- ▶ 235k registered users (including recently expired accounts, 6 month time limit)  
(roughly stable since last CS3)
- ▶ 5k project boxes (transferable group accounts)

## Sciebo Site Report

- ▶ started in 2015 for NRW
- ▶ serving branded ownCloud to about 40 academic institutions
- ▶ 235k registered users (including recently expired accounts, 6 month time limit)  
(roughly stable since last CS3)
- ▶ 5k project boxes (transferable group accounts)
- ▶ 3.8PB of user data including snapshots (30 days)

## Sciebo Site Report

- ▶ started in 2015 for NRW
- ▶ serving branded ownCloud to about 40 academic institutions
- ▶ 235k registered users (including recently expired accounts, 6 month time limit)  
(roughly stable since last CS3)
- ▶ 5k project boxes (transferable group accounts)
- ▶ 3.8PB of user data including snapshots (30 days)
- ▶ moved all production w/ exception of some minor homegrown services to k8s

## Setup

- ▶ owncloud + mariadb galera + keydb (redis) in sentinel mode on kubernetes per instance



## Setup

- ▶ owncloud + mariadb galera + keydb (redis) in sentinel mode on kubernetes per instance
- ▶ GPFS (IBM spectrum scale) Filesystem mounted as hostPath for user data

## Setup

- ▶ owncloud + mariadb galera + keydb (redis) in sentinel mode on kubernetes per instance
- ▶ GPFS (IBM spectrum scale) Filesystem mounted as hostPath for user data
- ▶ central LDAP for users

## Setup

- ▶ owncloud + mariadb galera + keydb (redis) in sentinel mode on kubernetes per instance
- ▶ GPFS (IBM spectrum scale) Filesystem mounted as hostPath for user data
- ▶ central LDAP for users
- ▶ some friendly additional services

## Takeaways for DBs in kubernetes

- ▶ we use openstack helm infra helm chart, not sure if  $\exists$  newer and better charts

## Takeaways for DBs in kubernetes

- ▶ we use openstack helm infra helm chart, not sure if  $\exists$  newer and better charts
- ▶ DB config tuning still necessary: config for a 300GB DB does not work for 128GB or 10GB database

## Takeaways for DBs in kubernetes

- ▶ we use openstack helm infra helm chart, not sure if  $\exists$  newer and better charts
- ▶ DB config tuning still necessary: config for a 300GB DB does not work for 128GB or 10GB database
- ▶ DB performance tuning is trivialized by hardware and ceph rbd

## Takeaways for DBs in kubernetes

- ▶ we use openstack helm infra helm chart, not sure if  $\exists$  newer and better charts
- ▶ DB config tuning still necessary: config for a 300GB DB does not work for 128GB or 10GB database
- ▶ DB performance tuning is trivialized by hardware and ceph rbd
- ▶ DB is still the bottleneck though, but you just can not out-tune huge tables like the dreaded filecache

## Takeaways for DBs in kubernetes

- ▶ we use openstack helm infra helm chart, not sure if  $\exists$  newer and better charts
- ▶ DB config tuning still necessary: config for a 300GB DB does not work for 128GB or 10GB database
- ▶ DB performance tuning is trivialized by hardware and ceph rbd
- ▶ DB is still the bottleneck though, but you just can not out-tune huge tables like the dreaded filecache
- ▶ for reliance the most critical parameters relate to synchronization of terminated nodes



## Takeaways for DBs in kubernetes

- ▶ overall robust experience

## Takeaways for DBs in kubernetes

- ▶ overall robust experience
- ▶ clean shutdown help clean startups; almost all issues are due to SST (complete transfer of state to new node)
- ▶ many smaller cluster definitely feel less terrifying than extremely large clusters

## Takeaways for DBs in kubernetes

- ▶ overall robust experience
- ▶ clean shutdown help clean startups; almost all issues are due to SST (complete transfer of state to new node)
- ▶ many smaller cluster definitely feel less terrifying than extremely large clusters
- ▶ performance is as good (or bad) as native; mainly limited by the synchronization guarantees

## Friends!

Service	level of integration	maturity	production level	deployment via
OnlyOffice	full (app)	feature complete	productive	docker on separate system
Collabora	full (app)	federation not working*	testing	easy helm chart
Overleaf	credentials	federation and file-sharing not working	testing on users	custom helm chart and codebase
IOP/Sciencemesh				helm chart and oC

\*last checked mid 2022

## Friends!

Service	level of integration	maturity	production level	deployment via
OnlyOffice	full (app)	feature complete	productive	docker on separate system
Collabora	full (app)	federation not working*	testing	easy helm chart
Overleaf	credentials	federation and file-sharing not working	testing on users	custom helm chart and codebase
IOP/Sciencemesh				helm chart and oC

\*last checked mid 2022

TODO: Install OnlyOffice with helm

## Helm über alles

- ▶ we find helm is the way to go for any deployment

## Helm über alles

- ▶ we find helm is the way to go for any deployment
- ▶ deployments are easily automated with ArgoCD

## Helm über alles

- ▶ we find helm is the way to go for any deployment
- ▶ deployments are easily automated with ArgoCD
- ▶ hacks can be made with kustomize or just shell scripts



## Co-Developing helm charts

- ▶ k8s replaced us admins with a helm chart

## Co-Developing helm charts

- ▶ k8s replaced us admins with a helm chart
- ▶ BUT we suddenly need to write down things computers do in a repeatable fashion, ...  
is this coding?

## Co-Developing helm charts

- ▶ k8s replaced us admins with a helm chart
- ▶ BUT we suddenly need to write down things computers do in a repeatable fashion, ... is this coding?
- ▶ also charts still need to be deployed by other people, ... this feels like package management

## Co-Developing helm charts

- ▶ k8s replaced us admins with a helm chart
- ▶ BUT we suddenly need to write down things computers do in a repeatable fashion, ... is this coding?
- ▶ also charts still need to be deployed by other people, ... this feels like package management
- ▶ charts need to somewhat track the actual application

## Co-Developing helm charts

- ▶ k8s replaced us admins with a helm chart
- ▶ BUT we suddenly need to write down things computers do in a repeatable fashion, ... is this coding?
- ▶ also charts still need to be deployed by other people, ... this feels like package management
- ▶ charts need to somewhat track the actual application
- ▶ we do not want to bog down the actual maintainers with the additional burden of managing charts

## Practices

Not best. Possibly not worst.

## Practices

Not best. Possibly not worst.

- ▶ helm charts and container image definitions in the same repo works very well for small applications

## Practices

Not best. Possibly not worst.

- ▶ helm charts and container image definitions in the same repo works very well for small applications
- ▶ kinda becomes annoying in applications with lots of dependencies and microservices or during large refactorings of the architecture



## Practices

Not best. Possibly not worst.

- ▶ helm charts and container image definitions in the same repo works very well for small applications
- ▶ kinda becomes annoying in applications with lots of dependencies and microservices or during large refactorings of the architecture
- ▶ `helm template` is good for tests, e.g. `helm template . | kubeval`

## Practices

Not best. Possibly not worst.

- ▶ helm charts and container image definitions in the same repo works very well for small applications
- ▶ kinda becomes annoying in applications with lots of dependencies and microservices or during large refactorings of the architecture
- ▶ `helm template` is good for tests, e.g. `helm template . | kubeval`
- ▶ `{{- fail "Some message."}}` can be used for compability tests inside the chart itself

## Practices

Not best. Possibly not worst.

- ▶ helm charts and container image definitions in the same repo works very well for small applications
- ▶ kinda becomes annoying in applications with lots of dependencies and microservices or during large refactorings of the architecture
- ▶ `helm template` is good for tests, e.g. `helm template . | kubeval`
- ▶ `{{- fail "Some message."}}` can be used for compability tests inside the chart itself
- ▶ automated deployments only really fun when you have helm chart in some helm repo

## Practices

Not best. Possibly not worst.

- ▶ helm charts and container image definitions in the same repo works very well for small applications
- ▶ kinda becomes annoying in applications with lots of dependencies and microservices or during large refactorings of the architecture
- ▶ `helm template` is good for tests, e.g. `helm template . | kubeval`
- ▶ `{{- fail "Some message."}}` can be used for compability tests inside the chart itself
- ▶ automated deployments only really fun when you have helm chart in some helm repo
- ▶ use Secrets for secrets; allow user provided secrets so the values YAML can be clean and committed unencrypted to git!

## Eliminating friction

**Todo** setup Github runners to make development more immediate

## Eliminating friction

**Todo** setup Github runners to make development more immediate

**Wish** declarative management of gitlab/github projects, ideally across git\* providers

## Eliminating friction

- Todo** setup Github runners to make development more immediate
- Wish** declarative management of gitlab/github projects, ideally across git\* providers
- Wish** personal Issue boards across git\* providers and projects

## Eliminating friction

**Todo** setup Github runners to make development more immediate

**Wish** declarative management of gitlab/github projects, ideally across git\* providers

**Wish** personal Issue boards across git\* providers and projects

**LEARNING** Licenses and Contributor Agreements are necessary and not too bad; they *enable* some people to contribute and use the product



## Eliminating friction

**Todo** setup Github runners to make development more immediate

**Wish** declarative management of gitlab/github projects, ideally across git\* providers

**Wish** personal Issue boards across git\* providers and projects

**LEARNING** Licenses and Contributor Agreements are necessary and not too bad; they *enable* some people to contribute and use the product

**Resolution** Upstream first; try fixing issues upstream instead of quick workarounds

**Thanks!**

Thanks!