

# Library publishing technology services at scale

*or, what goes into running a  
multi-institutional software hosting service.*

**Access Conference 2019**

Kaitlin Newson | [@kaitlinnewson](https://twitter.com/kaitlinnewson)

Digital Projects Librarian, Scholars Portal

# Publishing services at Scholars Portal

# Publishing services at Scholars Portal

---

- Public Knowledge Project (**PKP**)
  - Open Journal Systems (**OJS**)
  - Open Monograph Press (**OMP**)

# Publishing services at Scholars Portal

---

- 11 universities in Ontario, 125+ journals
- 2 librarians & systems admin
- Includes technical support, sponsored CrossRef memberships, analytics, theming & customizations

# Service scaling

---

- 1 -> 11 libraries
- Balancing sustainability, service & workload

# Client support

# Client support

---

- Workload management
- Time tracking
- Defining roles & responsibilities
- External communities

# Code management



# Code updates

---

- 1 repository per instance
- Deciding when to upgrade
  - Security issues?
  - What are the new features?
  - Minor or major release?
  - Known bugs?

# Code customizations

---

- Institutions request individual customizations
- Every code customization has a cost associated with it when you upgrade
- Consider:
  - Could this be useful to the larger project community?
  - How large of a change is this?
  - Where does the code for this live?

# Environments and deployment

# Environments

---

- Local & development environments
- Environment configuration
- Multi-server -> shared server

# Upgrading OJS 3 instances on OJS server

---

Once the code is updated in GitLab and you've verified the submodule commits:

1. Set permissions of the ojs directory to the ojs user:

```
sudo chown -R ojs:users the-ojs-directory/
```

2. `sudo su - ojs` to login as the OJS user which has access to pull from GitLab

3. cd to the OJS directory

4. If existing branch:

```
git pull
```

If new branch:

```
git fetch
git checkout -b stable-3_1_X origin/stable
```

5. Reset the submodule directories so we don't have issues updating them:

```
git submodule foreach git reset --hard
```

6. Update the submodules:

```
git submodule update --init --recursive
git submodule <- check to see everything
```

7. Run composer updates:

```
composer --working-dir=lib/pkp update
composer --working-dir=plugins/paymethod
composer --working-dir=plugins/generic/c
```

8. Get NPM dependencies (run in the main OJS directory):

```
npm install
npm run build
```

# Jenkins

---

- automation server for deploying and automating tasks
- self-hosted
- <https://jenkins.io>



# Jenkins

# Project OJSDEV- Create OJS Dev Instance

This build requires parameters:

INSTANCE\_NAME

Choose Instance to Clone from Prod and Deploy to Test

BRANCH\_TAG

To build a particular Branch use : refs/heads/{branchName}  
Eg. refs/heads/master

To build from a Particular Tag use : refs/tags/{tagName}  
Eg. refs/tags/git-2.3.0

To build from a Particular Commit use : {commitId}  
Eg. 5062ac84

GET\_LATEST\_DB

**Get Latest Database from Production and Restore to DEV**

Note: It will take upto 40minutes to Run this step depending on DB Size!

RUN\_UPGRADE

**Check this if you want to Run Upgrade step**

# Community contribution



# Community contribution

---

- PKP Forums
- Code and documentation
- Working groups & communities
- Financial support
- Supporting broader initiatives, e.g.

<https://investinopen.org/>

# Questions?

[kaitlin@scholarsportal.info](mailto:kaitlin@scholarsportal.info)

[ojs@scholarsportal.info](mailto:ojs@scholarsportal.info)