

Open Library Python3 🐸 Game-Plan

FINISHED: Moving remaining issues to [Open Library: Provisioning & Deployment](#)

<https://github.com/internetarchive/openlibrary/projects/30>

See following projects: <https://github.com/>

The Internet Archive is seeking an experienced python contractor to help the Open Library team achieve Python 2 → 3 compatibility within the next two months (before August 31) at a rate of \$xx/h. The project scope includes 3 repositories: the codebase for the [OpenLibrary.org](https://github.com/internetarchive/openlibrary) website (<https://github.com/internetarchive/openlibrary>), the [webpy](https://github.com/internetarchive/infogami)-based [infogami](https://github.com/internetarchive/infogami) (<http://github.com/internetarchive/infogami>) microframework, and a very small private configuration repo called [olsystem](#).

Skills:

- Python 3, pytest, pip, py2to3, virtualenv
- Docker, gunicorn, supervisor, nginx, haproxy

Completion:

The project is successful when both our Docker development environment and also OpenLibrary.org's 2 production web-application servers (ol-web3, ol-web4) are running Python 3 in production Docker.

Progress:

Since Oct 2018:

1. more than [150 pull requests](#) have been merged by [Chris](#) and others towards migrating the OpenLibrary repo.
2. A [1st pass](#) pull request has been completed and merged to migrate infogami to Python 3 by [Albert](#), [Galen](#), and [Mek](#)
3. Our [production Travis CI](#) is configured to run tests both against Python 2 and Python 3

Next Steps:

1. **Dev Feedback Loop: Audit & Answers**
 - a. Q: Does Open Library run on the latest Infogami (in Travis? in docker?)
 - i. This can be manual, it would be great to answer this week
 - b. What happens when we try to run Open Library on docker + py3?
 - i. Are there questions about how to set this up? E.g. installing python3, because Asheesh might be a great resource to unblock us. Otherwise [@Drini](#) + [@Mek](#)
 - ii. Can we add results to <https://github.com/internetarchive/openlibrary/issues/3333>
 - c. When can we do a dev/prod deploy of infogami?

- d. Create issues for the above (to document what's broken -- maybe folks in the community will help)
2. **Infogami Repo.** The first 2 weeks will be dedicated to bringing Infogami to python3 compatibility, including fixing and adding basic unit test coverage.
3. **OpenLibrary Repo.** The 3rd and 4th weeks will be used to clean up our pytests for python 3 in the OpenLibrary codebase.
 - a. Most issues related to bytes-vs-strings
4. **Python 3 Developer Docker.** The 5th and 6th weeks will be used to get Open Library running on Python 3 within our local development [Docker](#) environment and to complete the [transition of the OpenLibrary repository to Python 3](#) (<https://github.com/internetarchive/openlibrary/issues/3333>).
5. **Python 3 Production Docker.** The 7th and 8th week will be used either for padding or for provisioning new servers to migrate our [production systems to use Docker + Python3](#) (<https://github.com/internetarchive/openlibrary/issues/3416> & <https://github.com/internetarchive/openlibrary/issues/3419>). This includes making sure any of the scripts within our configuration repository ([olsystem](#)) have been ported.

Topics -- [Python 3 taskboard](#)

05 April 2021.

Knowledge Transfer

- [] Deploy with Drini so Drini knows the ropes
- [] GitHub Actions knowledge transfer

Todo:

- Data dumps: Run split step with fix
- Remove 3.9 version from .python-version+ update to 3.9.4
- Open PR that switches the default to be 3.9

- Script 20* is blocking mypy; mypy tries to build a syntax tree even for excluded files, and hence errors on these since they are py2 only.

04 Mar. 2021 Wrapping up cron-job loose ends

1. Quick scan of [Open P1s](#)
2. Deploy + [checking Sentry](#)? and [v1.0.0](#)
3. Volume mounts [#4719](#)
4. Closing/merging any remaining work re: crons [#4383](#) + Ferm and Docker discussion
5. Remaining Py3 Unicode bugs from [@Lisa](#)?
6. Reviews backlog: [#4446](#) [#4274](#) [#4593](#) [#4604](#) [#4605](#)

7. Deprecating Py2 from docker and requirements [#4386](#)
8. ol-www0 nginx (@mek) [#4252](#)

01 Mar. 2021 olsystem/bin/cron/oldump.sh cron job now runs end-to-end on ol-home0

Today starts a new Active Sprint

1. Let's focus today's discussion on closing out [olsystem#118](#) → [#4622](#) → [#4383](#)

22 Feb. 2021 Find and fix merge authors with Unicode names

1. Greenlight to upgrade <https://dev.openlibrary.org> to Ubuntu Focal?
2. Talking points for volume mounts vs. repos inside Docker **TL;DR is let's try this!**
3. Find and fix merge authors with Unicode names [#4225](#)

Talking points for volume mounts vs. repos inside Docker

Pros for moving **openlibrary** and **infogami** repos inside the Docker container:

1. Easier to guarantee that all hosts have exactly the same content
2. Reduce complexity of syncing repos (sync 2 (→ 3) repos instead of 4)
3. Reducing the number of mount points simplifies docker-compose.yml
4. Simplifies rollback

Cons for moving **openlibrary** and **infogami** repos inside the Docker container:

1. On-the-fly editing of content inside containers is tricky (no **sudo**, etc.) **-u root**
2. Need to **docker cp** modifications in/out of container -- only for priority patches
 - a. Patch / apply -- docker restart preserves changes - down/up destroys changes
3. **NOT!!!** Docker containers are larger so deployment (save -> rsync -> load) is slower -- **This code is in ol-base so this means a waiting period between merging to master before deploy!**
4. How do we run i18n? -- This is run by ol-base build process for Production only. Same for npm install, etc.

How: The root docker-compose must be edited to remove the volume mounts and those mount directives into override and/or staging docs.

Performance issue / Cython issue??

17 Feb. 2021 Finalize oldump cron job on Python 3 in Docker

1. Staging and Dev <https://github.com/internetarchive/openlibrary/issues/4229>

2. Quick PRs to review and land:

- a. <https://github.com/internetarchive/openlibrary/pull/4618>
- b. <https://github.com/internetarchive/olsystem/pull/117>
- c. [Removing urlopen\(\)](#)

3. DRAFT: Fix cron job oldump to run in Docker [#118](#)

4. Copydocs issue <https://github.com/internetarchive/openlibrary/issues/4603>

5. PUNT: Merge authors with Unicode names [#4225](#)

- [] @cclauss Talking points for volume mounts vs. repos inside Docker
- [x] @cclauss Determine if bin/ipstats.py is used; [delete if not](#)
- [x] @cclauss Change scripts/ipstats.py shebang + add comments [#4628](#)
- [x] @cdrini Expand the cronjobs issue to have a section for ol-www cron jobs (those points will be blocked by Mek's www1 works)
- [] @cclauss Need to remove the cron jobs that are now running on python3
 - Find the cron file on ol-home, edit to rm the ones that need removing
- [] @cclauss We should stop deploying new code to the old ol-home; let it deprecate. Remove ol-home from deploy targets (no longer getting updates)
- [x] @cdrini Remove py2 tests from Jenkins
- [] @cclauss Add error from oldump to issue
- [x] @cdrini update Nothing>int issue

08 Feb. 2021 Finalize cron jobs on Python 3 in Docker

6. It seems like our top-ish priorities for dev-ops epic are:

Finishing cron migration [#4387](#)

Testing the affiliate-server deploy [#4308](#)

Volume mounts

Provisioning ol-www0 -- open Q:

- o @cdrini, would help be appreciated from @cclauss on the provisioning element of ol-www0? [#4566](#)

@cclauss could use help making sense of next steps for [#4225](#)

7. So for Thursday:

Proposal: Land [olsystem#110](#) (merged) + [#4387](#) (including loan_stats, imports, accounts, general stats) -- w/o ol_dumps (separate PR: [olsystem#118](#))

Done: Move out the first two files (OLDUMP related)

Affiliate code should be tested

To discuss: [#4225](#) merging authors: authors or works have keyvals w/ Unicode characters

8. For next week:

Volume mounts

[olsystem#118](#) ol_dumps cron? (This would allow us to close Python 2?)

Support on ol-www0

14 Jan. 2021 A script-driven Deployment

1. Deployments should be triggered from ol-home0
 - a. [Scripts for production deployments](#) #4395 (Merged)
 - b. [Create importbot docker service](#) #4324 (Merged)
 - c. [DRAFT: Run cron jobs in Docker](#) #4387
 - d. [Deployments should be triggered from ol-home0](#) #4318
2. Low hanging fruit [PRs](#)
 - a. [Remove legacy Python from requirements*.txt and Docker](#) #4386
 - b. [Use requests in solr_client.py](#) #4317
 - c. [Remove duplicate orphaned edition code from addbook.py](#) #4307
 - d. [Docker: Python 3.9.1 runs natively on Apple M1 Macs](#) #4232
 - e. [Olsystem open PRs](#)

11 Jan. 2021 Deployment discussion

3. Debugging is broken <https://github.com/internetarchive/openlibrary/wiki/Debugging>
 - a. Admin password conflict → admin123
 - b. Cookie cleaning -- Done.
 - c. Workers to 1 in which file?
 - d. [Migrate debugging from ptvsd to debugpy -- again](#) #4274
4. Deployments should be triggered from ol-home0
 - a. [Create importbot docker service](#) #4324
 - b. [DRAFT: Run cron jobs in Docker](#) #4387
 - c. [#4318](#)
5. Low hanging fruit PRs
 - a. [Scripts for production deployments](#) #4395
 - b. [Remove legacy Python from requirements*.txt and Docker](#) #4386
 - c. [Add vim and emacs to Dockerfile.olbase](#) #4375
 - d. [Use requests in solr_client.py](#) #4317
 - e. [Remove duplicate orphaned edition code from addbook.py](#) #4307
 - f. [Docker: Python 3.9.1 runs natively on Apple M1 Macs](#) #4232

04 Jan. 2021 Deployment discussion

1. On ol-home0
 - a. `cd /opt/openlibrary`
 - b. `git sync openlibrary, infogami, olsystem, booklending_utils`
 - c. `docker-compose build` the images and `docker save` to a .gz file
 - d. [rsync the .gz image](#) to ol-covers0 ol-solr ol-web1 ol-web2
 - e. `rsync olsystem` to ol-covers0 ol-solr ol-web1 ol-web2 → shift a symlink
2. On each of the hosts
 - a. `cd /opt/openlibrary`
 - b. `git sync openlibrary, infogami, olsystem`
 - c. On ol-web nodes also `git sync booklending_utils`
 - d. `docker load` the .gz image
3. Docker up the images on all hosts.

~~Try to bake in **booklending_utils** in the Docker image. ???~~

Revert... Keep the last three .gz files? Or keep them all on ol-home0 and rsync to hosts.

Deploy based on a SHA

Build and make i18n issues

Node modules issues

OLsystem should always be volume-mounted -- Keep using **fabfile.py** from ol-home0

Booklending_utils should always be volume-mounted

- Try removing /openlibrary, infogami mount from `docker-compose.production.yml`
- Add vim, emacs to `Dockerfile.oldev`

oldev:latest

oldev:e654e45

31 Dec. 2020 Year end discussion

1. [Moving ol-home to python3 node](#) #4060 → PR: [Create importbot docker service](#) #4324
2. [Retire ol-web3 & ol-web4](#) #3992 → ol-web3 is now gone.
3. [Switch Continuous Integration to using Docker](#) #3443 → PR: [GitHub Action to test docker-compose](#) #4242

14 Dec. 2020 office hours

1. Thread proposal for Amazon API #4221 [affiliate-server with queue for amz](#)
2. P1 [Update olsystem ol-covers1 → ol-covers0](#) #4203 (CLOSED)
3. [Moving ol-home to python3 node](#) #4060 → PR: [Create solr-updater docker service](#) #4244 → [Draft: Create solr-updater docker service AGAIN](#) #4287
4. [Orphaned editions throw IndexError](#) #4051 → [Make work from an orphaned edition](#) #4286 (MERGED)
5. [Retire ol-web3 & ol-web4](#) #3992 → ol-web3 is now gone.

6. [Switch Continuous Integration to using Docker](#) #3443 → PR: [GitHub Action to test docker-compose](#) #4242

Final steps in Python 3 migration (07 Dec. 2020) office hours

7. Amazon API requests: #4221 [affiliate-server with queue for amz](#) and [AmazonAPI](#) label
8. Feedback needed on our new [Deployment Guide](#)
9. What is our timeline for dropping Python 2 support? Gradual addition of type hints, etc.
10. Recycling our [Active Sprint](#) -- Priorities for December and for 2021
11. #4060 [Moving ol-home to Python 3 node](#) -- Cron tasks !!
12. #3992 [Retire ol-web3 & ol-web4](#) docker-compose.yml → [kubernetes kompose](#) -- Retire **covers 1 -- find and replace**
13. <https://github.com/internetarchive/openlibrary/pulls/cclauss> 14 open PRs
14. Bookserver and Archive.cd

ol-solr2, ol-covers1, ol-home1

Pyt 3 book sponsorship

Todos: Review Deployment Guide, Review Open PRs.

CCC: Focus on #4060 plus retiring covers1

Covers0 deploy (30 Nov. 2020)

Prep (async):

1. [x] @cdrini: rsync from covers1 → covers0

...

on covers0:

```
sudo chown drini:drini -R /1/var/lib/openlibrary/coverstore/localhost/2020/11
```

on covers1

no longer want --delete since covers could have been uploaded to covers0

```
sudo rsync -aiPh --stats /1/var/lib/openlibrary/coverstore/localhost/2020/11/
drini@ol-covers0:/1/var/lib/openlibrary/coverstore/localhost/2020/11/
```

on covers0:

```
sudo chown 999:999 -R /1/var/lib/openlibrary/coverstore/localhost/2020/11
```

...

3. [x] @cclauss: Clear up an unnecessary local mods
4. [x] @cclauss: Covers0: Checkout latest olsystem
5. [x] @cclauss: Covers0: Checkout latest openlibrary
6. [x] @cclauss: Covers0: Checkout latest infogami
7. [] @cclauss: Covers0: build/pull docker images

...

sudo bash

```
export
COMPOSE_FILE="docker-compose.yml:docker-compose.infogami-local.yml:docker-com
pose.production.yml"
https://sentry.archive.org/sentry/ol-web/?environment=production
```

```
docker-compose pull covers covers_nginx memcached
docker-compose build --pull covers covers_nginx memcached
```

```
# Cleanup
docker container prune
docker volume prune # Shouldn't usually run this!
...
```

Actual transfer

```
1. [x] @cdrini Clear up an unnecessary local mods
2. [SKIP] @cdrini rsync over (can skip if done very recently)
...
```

on covers0:

```
sudo chown drini:drini -R
/1/var/lib/openlibrary/coverstore/localdisk/2020/11/
```

on covers1 -- no longer want --delete since covers could have been uploaded to covers0

```
sudo rsync -aiPh --stats /1/var/lib/openlibrary/coverstore/localdisk/2020/11/
drini@ol-covers0:/1/var/lib/openlibrary/coverstore/localdisk/2020/11/
```

on covers0:

```
sudo chown 999:999 -R /1/var/lib/openlibrary/coverstore/localdisk/2020/11/
...
```

```
3. [x] @cclauss Start up on covers0
...
```

```
sudo bash
export
COMPOSE_FILE="docker-compose.yml:docker-compose.infogami-local.yml:docker-com
pose.production.yml"
docker-compose up -d --scale covers=2 covers_nginx memcached
# no scaling: docker-compose up -d covers covers_nginx memcached
...
```

```
4. [x] @cdrini: Test staging
...
```

```
# Edit to point to http://ol-covers0.us.archive.org:8081/
vim /opt/olsystem/etc/openlibrary.yml
sudo docker restart openlibrary_web_1 openlibrary_memcached_1
...
```


Got: pyscop2 too many connections; tried stopping one of the old covers

...

```
sudo supervisorctl stop coverstore-7076
```

...

5. [x] @cdrini: Test dev.openlibrary.org

...

Edit to point to <http://ol-covers0.us.archive.org:8081/>

```
vim /opt/olsystem/etc/openlibrary.yml
```

...

>> Failed: Uploading doesn't work; PR ready

6. [] Set dev1/dev0 to use prod covers

7. Rsync the upload on covers0 to covers1

on covers0:

```
sudo chown drini:drini -R
```

```
/1/var/lib/openlibrary/coverstore/localdisk/2020/11/25
```

on covers1:

```
sudo chown drini:staff -R
```

```
/1/var/lib/openlibrary/coverstore/localdisk/2020/11/25
```

on covers1 -- **no longer want --delete** since covers could have been uploaded to covers0

```
rsync -aiPh --stats /1/var/lib/openlibrary/coverstore/localdisk/2020/11/25/
```

```
drini@ol-covers0:/1/var/lib/openlibrary/coverstore/localdisk/2020/11/25/
```

on covers0 -- **no longer want --delete** since covers could have been uploaded to covers0

```
rsync -aiPh --stats /1/var/lib/openlibrary/coverstore/localdisk/2020/11/25/
```

```
drini@ol-covers1:/1/var/lib/openlibrary/coverstore/localdisk/2020/11/25/
```

on covers1:

```
sudo chown openlibrary:openlibrary -R
```

```
/1/var/lib/openlibrary/coverstore/localdisk/2020/11/25
```

on covers0:

```
sudo chown 999:999 -R /1/var/lib/openlibrary/coverstore/localdisk/2020/11/25
```

...

8. [x] down covers0

NOTE: Not restarting 7076 on covers1; seems unlikely to cause an issue? 🙌

5. [x] @mek: Notify of potential downtime - @mek did this on the All-hands

6. [x] @cdrini: Covers1: Edit nginx to point to ol-covers0 at port 8081

...

```
sudo nginx -t -c /etc/nginx/nginx.conf
sudo service nginx reload
...
```

7. [x] @cdrini: Stop covers1
...

```
sudo supervisorctl status
# Then stop them
...
```

8. [x] Monitor:

- Grafana:
https://mtrcs-db0.us.archive.org/dashboard/db/ol-solr0?from=now-6h&to=now&var-fqhos_tname=ol-covers0_us_archive_org&var-resolution=30m
- Sentry: <https://sentry.archive.org/sentry/ol-covers/>

Next steps covers:

- [x] @cdrini: Test + merge PR to fix uploading from py2-web to py3-covers
- [x](Mon) @mek Deployed latest code to web nodes before we try going to py3-covers
- [x] @cdrini: Try mounting olsystem default nginx to docker covers
- [x] @cclauss: replacing the print/web.debug in coverstore/code & coverstore/coverlib with logger.exception so we can see them in sentry
- Monday: We'll deploy covers for good 🙌

Next steps infobase:

- [x] @cclauss: Open Sentry PR
 - Move sentry binding here if possible:
<https://github.com/internetarchive/openlibrary/blob/ede14f159c6a2ea57090956a718f4934c07f5d7a/scripts/infobase-server#L39-L59>
 - Let's see if we can get a DivZero error in sentry from local
- [x] Nginx - @cdrini review: <https://github.com/internetarchive/openlibrary/pull/4140/files>
 - Also: Mounting ia.ini

Notes:

- Might need to change docker-compose.production.yml covers_nginx to export port 80
- Manual change to olsystem coverstore.yml to point to local memcached at memcached:11211
- Ask Andy about /opt/.petabox/seed not existing on new VMs

Home (infobase on **ol-home0**) deploy (01 Dec 2020)

- Highest priority: ol-home must not be publicly accessible

Tracking issue is [#4059](#) is the tracking issue under [#4060](#).

Risks:

1. Two infobases running on same cache or db vs. read-only cache or db for one
2. Need tight process reversal steps.
3. Are these [#2755](#) [#4045](#) problem? -- Don't think related.
4. Any worries/mitigations on the flup upgrade?

Cheat sheet for starting infobase:

- https://github.com/internetarchive/openlibrary/blob/master/scripts/run_olserver.sh

1. [x] @mek: Make sure ol-home0 not accessible publicly

...

vim /etc/firm.conf # Remove non-cluster

...

2. [x] @cclauss: Clear up an unnecessary local mods

3. [x] @cclauss: ol-home0: Checkout latest olsystem

4. [x] @cclauss: ol-home0: Checkout latest openlibrary

5. [x] @cclauss: ol-home0: Checkout latest infogami

6. [x] @cclauss: Update infobase.yml with any needed modifications

7. [] @cclauss: ol-home0: build/pull docker images

...

sudo bash

export

COMPOSE_FILE="docker-compose.yml:docker-compose.infogami-local.yml:docker-compose.production.yml"

docker-compose pull infobase infobase_nginx

docker-compose build --pull infobase infobase_nginx

Cleanup

docker container prune

...

Key files:

<https://github.com/internetarchive/openlibrary/blob/master/docker/ol-home-start.sh>

<https://github.com/internetarchive/openlibrary/blob/master/docker/ol-infobase-start.sh>

<https://github.com/internetarchive/openlibrary/blob/master/conf/infobase.yml>

<https://github.com/internetarchive/openlibrary/blob/master/scripts/infobase-server>

[] Fire up: <https://sentry.archive.org/sentry/ol-infobase/>

The switch:

1. [x] @mek Notify #openlibrary #openlibrary-g of potential outages

2. [x] Set up an nginx reverse-proxy from ol-home to ol-home0

[x] @cclauss: Start on ol-home

[x] @mek: Reload nginx

[x] Quick testing running on 7000

[x] @cdrini: Stop infobase on ol-home

1. [] @cclauss Stop infobase on ol-home

2. [] @cclauss Start infobase on ol-home0

...

sudo bash

export

COMPOSE_FILE="docker-compose.yml:docker-compose.infogami-local.yml:docker-compose.production.yml"

docker-compose up -d --no-deps infobase infobase_nginx

...

3. [] @cclauss Test it's running:

...

ssh -L 7000:ol-home0:7000 -qAy YOURNAME@sshgw-sf.us.archive.org

...

3. [] @cclauss Monitor:

- Sentry: <https://sentry.archive.org/sentry/ol-infobase/>

- Grafana:

https://mtrcs-db0.us.archive.org/dashboard/db/ol-solr0?from=now-6h&to=now&var-fqhostname=ol-home0_us_archive_org&var-resolution=30m

Notes:

- We needed to chown the the directories related to logs; this was a manual step (which we likely won't have to repeat since they're now named volumes; unless the openlibrary user's UID changes in the Dockerfile for some reason).

[] Whenever we provision a new machine Copy seed to /opt/.petabox/seed

[] Infobase logs are now stunted; had to restart solr-updater. Likely ~15 min will be missing from solr

[] We're still proxying through the old ol-home

[] We want to do a find-and-replace of all ol-home:7000 -> ol-home:0700

[] Eventually remove the reverse proxy nginx and see what breaks

[] Need to set up Nagios

Web 1 & 2 deploy -- 3rd run on 02 Dec 2020

Resources:

- <https://github.com/internetarchive/openlibrary/wiki/Deployment-Guide#ol-web1>
- https://github.com/internetarchive/openlibrary/blob/master/scripts/run_olserver.sh

Game plan:

- [] @mek Ping openlibrary-g, openlibrary, ops
- [x] @mek Add web2 to pg hba conf to olsystem
- [x] `/etc/init.d/postgres reload` on ol-db1 (and ol-db2 to be safe)
- [x] @mek put pg hba conf on [x] db1, [x] db2, [x] web3, [x] web4
- [x] @cdrini Change olsystem openlibrary yml to use ol-solr1

- [x] @cdrini web1 ready
 - [x] web1 pull latest master
 - <https://github.com/internetarchive/openlibrary/wiki/Deployment-Guide#ol-web1>
 - [x] sudo git pull origin master for olsystem
 - [x] web1 build
 - [] web1 down; up
 - [] Confirm running ol-web1.us.archive.org:8080/status
 - [] Confirm software version on /status page: 488fb81
 - [] confirm hostname on /status
 - [] Disconnect web1 from sentry - `sudo vim /opt/olsystem/etc/openlibrary.yml`

- [x] @chris Web2 ready
 - <https://github.com/internetarchive/openlibrary/wiki/Deployment-Guide#ol-web1>
 - [x] web2 pull latest master
 - [x] sudo git pull origin master for olsystem
 - [x] web2 build
 - [] web2 down; up
 - [] Confirm running ol-web2.us.archive.org:8080/status
 - [] Confirm software version on /status page: 488fb81
 - [] confirm hostname on /status

- ~~— [] @cdrini Add sentry quota for OL so it doesn't creak~~ Don't connect sentry to web1

- [] WAIT web1, web2 ready
 - [] @mek Add web1 to haproxy pool

- [] @mek Add web2 to haproxy pool
- [] @mek Remove web3 from haproxy pool

IMPORTANT: or <https://github.com/internetarchive/openlibrary/issues/4209> will happen

- [] @all: ol-web3: sudo supervisorctl status ; sudo supervisorctl stop openlibrary
- [] @all: ol-web4: sudo supervisorctl status ; sudo supervisorctl stop openlibrary

- [] @all: Monitor on sentry
- [] Monitor performance/errors: <https://mtrcs-db0.us.archive.org/dashboard/db/open-library-dev>
- [] @all: Monitor on haproxy

[] Drini: register

[] Mek: Borrow a book

[x] Chris [TypeError hmac in __init__](#)
in hmac.py

key: expected bytes or bytearray, but got 'str'

LO-WEB-M1 & OL-WEB-M2

a few seconds ago — 10 days old

Found issues:

- 1. OL-WEB-M2 - hmac error when trying to build ia token for “backdoor” ia login
- 2. OL-WEB-KV - Going to https://openlibrary.org/books/OL30884340M/Stamtavle_over_sl%C3%A6gterne_Stampeog errors when it tries to redirect because of the non-ascii char
- 3. OL-WEB-QD float/int

Possible issues:

- [x] infobase has a copy of memcached; if we're not starting it might error
-> They did not!
- [x] web3/web4 might get kicked out after we add web1 to prod memcache pool
-> And they did!

Todo later (unrelated):

[] Make sure no staged olsystem changes on home

[] Got sent to archive.org after logging in with new account!

[] Borrowing does not auto login on IA?

[] Orphaned editions error:

https://openlibrary.org/books/OL8002423M/Hang_in_There!_Inspirational_Art_of_the_1970s?debug=true OL-WEB-2

~~__class__ assignment failing silently~~

Colab: <https://colab.research.google.com/drive/1lwiqb3QNsGSDIa8N3tpCtZ6Clhb1zbsw>

1. Open Library uses Infogami as its wiki and ORM layer
2. We have a set of models which may be used in templates which are defined in `openlibrary.plugins.upstream.models` and these models extend `openlibrary.core.models`
3. When we're in html and we're call
 - a. After we call `edition.works[0].title` on Python3, `editions.works[0]` type is Thing. On Python2 is Work.
 - b. `edition.works[0].get_sorted_editions()`
 - c.
4. In infogami (which file)

Next steps

- [] Ping Asheesh
- [] Create test so we can quickly see when this is failing
- [] Find minimal example that reproduces the bug
 - Maybe the fact that we have 2 levels of inheritance?
 - Maybe one of the classmethods? static methods? Property methods?
- [] Maybe ping on python IRC to find a reproducible example?

Actions

- [x] @mek Grant access to ol-web3 (goal is to deploy dockerized subset of OL)
- [x] @cclauss Fix Python 3: Add Cover Image hangs [#3733](#) <- last item on UI
 - Put on dev + staging @mek to merge
- [x] @cclauss look at olsystem
 - <https://github.com/internetarchive/olsystem/pull/89> awaiting review
- [x] Chris: StringIO undo <https://github.com/internetarchive/openlibrary/issues/3806>
- [x] Chris: Stress testing of staging.openlibrary.org
- [x] Chris: staging "deploy"

Minutes

2020-11-23

Python 3 in Production -- <https://github.com/internetarchive/openlibrary/wiki/Deployment-Guide-on-Python-3>

- ol-home0 runs infobase
- ol-covers0 runs covers
- ol-web{1, 2} run web
- ol-mem{3, 4, 5} run memcached
- ol-solr{0, 1} run solr
- ol-www1 runs the web head

PRs to review

- <https://github.com/internetarchive/olsystem/pulls> 102, 100
- <https://github.com/internetarchive/openlibrary/pulls> 4128, 4140
 - Also: 4052, 4142, 4143, 4144, 4146

Issues to review

- <https://github.com/internetarchive/openlibrary/issues>
 - 4130, 4115, 4061, 4060, 4059, 4051, 4017, 3993, 3992
 - Also: 4121

2020-11-16

Philosophy

- Covers, infobase, forcing function to set priority on most important pieces

Actionables

- Mek to manually cli try Amazon API, sponsorship
- @cdrini to attach ol-covers0 to sentry
- When we redeploy, use the keys

Proposals

- Have ol-home coordinate Docker building and keys

Hit-list PRs

- [x] [4083](#) - covers
- [-] 4052
- 4102
- 4097 - rm get_items_with_retry amz

This week

- **Covers**
 - Move content from /1 on ol-covers1 → ol-covers0
 - Can we point staging to ol-covers0?
 - Update coverstore.yml + openlibrary.yml as needed to point staging.openlibrary.org to covers0

- Test that some of these books that we copied over are accessible via staging + covers0
- **Infobase**
 -

2020-11-12

Suggestion: Add Sentry URLs to issues and PRs as we create them but also assign yourself to Sentry issues and add relevant GitHub URLs to the Sentry issue comments.

Python 3 has been in production for about 12 hours....

- [PRs for review](#): #4052, #4076, #4078, #4084
- Priority: Covers on Py3 <https://github.com/internetarchive/openlibrary/issues/4054>
- Pair programming: Automatically link Sentry issues to GitHub: [#3830](#)
- <https://github.com/internetarchive/openlibrary/issues/assigned/cclauss>

2020-11-02 (Office Hours)

1. Reverting a mistaken push to master
2. The dev-merged process
3. olsystem <https://github.com/internetarchive/olsystem/pulls>
4. Infogami <https://github.com/internetarchive/infogami/pull/119>
5. Testing: <https://travis-ci.org> → <https://travis-ci.com> vs. GitHub Actions
6. <https://github.com/internetarchive/openlibrary/issues/3991#issuecomment-720131377>
7. <https://github.com/internetarchive/openlibrary/pull/3973> vs. #3970 vs. #3955

2020-10-29

8. <https://sentry.archive.org> Is down again. Still struggling with how to get Sentry to ONLY show Python 3.8.6 events?
9. [On Dev](#) pull requests
10. Testing: [test_compose_ia_url\(\)](#) mocking out **db_parameters**
11. <https://github.com/internetarchive/openlibrary/pull/3733> can we make breaking change?

Ol-web1

- [] Remove nginx from the PR
- [] Confirm if cover uploading is broken on staging
- [] Get web-1 working then add a web-2

Problem URLs on Python 3:

- <http://ol-web1.us.archive.org:8080/subjects&debug=true>

- <http://ol-web1.us.archive.org:8080/sponsorship&debug=true>
- <http://ol-web1.us.archive.org:8080/collections/k-12&debug=true>
- http://ol-web1.us.archive.org:8080/subjects/art#sort=date_published&ebooks=true&debug=true
- <http://ol-web1.us.archive.org:8080/works/OL59728W?edition=moonisharshmistr00robe&debug=true>

2020-10-26 (Office Hours)

12. <https://openlibrary.org/admin?stats> is `red` a problem? ;-)
13. <https://sentry.archive.org/sentry/ol-web/?environment=production&query=runtime%3A%22CPython+3.8.6%22> How do I get Sentry to ONLY show Python 3.8.6 events?
14. <https://github.com/internetarchive/openlibrary/pull/3955> raise ValueError vs. <https://github.com/internetarchive/openlibrary/pull/3950> refactor `compose_ia_url()`
15. <https://github.com/internetarchive/openlibrary/pull/3733> can we make breaking change?

2020-10-21

ol-web1 is now FINALLY running on Python 3.8.6 on Docker...

- <https://github.com/internetarchive/openlibrary/pull/3938>
- `ol-www1% sudo vi /olsystem/etc/haproxy/haproxy.cfg`
- `ol-www1% sudo service haproxy reload`

2020-10-12

- `ol-web1` runs on 207.241.231.16
- Running docker on `ol-web1`
 - **`USE_NGINX=true sudo docker-compose -f docker-compose.yml -f docker-compose.infogami-local.yml -f docker-compose.production.yml up --no-deps -d web`**
- How to check if/why docker is failing?
 - **`sudo docker-compose logs --tail=100 -f web`**
 - It looks like `olsystem` is currently empty (git clone command failing? tokens?)
- Next steps
 - Check milestone --
 - Few open issues & PRs (sentry + Python3)
 - StringIO
 - Changing covers
 - Book Imports ??
 - PR for `olsystem` haproxy -- add `ol-web1` to the pool

```

web_1      | File
"/home/openlibrary/.pyenv/versions/3.8.5/lib/python3.8/site-packages/gunicorn/arbiter.py",
line 583, in spawn_worker
web_1      | worker.init_process()
web_1      | File
"/home/openlibrary/.pyenv/versions/3.8.5/lib/python3.8/site-packages/gunicorn/workers/base.py"
, line 129, in init_process
web_1      | self.load_wsgi()
web_1      | File
"/home/openlibrary/.pyenv/versions/3.8.5/lib/python3.8/site-packages/gunicorn/workers/base.py"
, line 138, in load_wsgi
web_1      | self.wsgi = self.app.wsgi()
web_1      | File
"/home/openlibrary/.pyenv/versions/3.8.5/lib/python3.8/site-packages/gunicorn/app/base.py",
line 67, in wsgi
web_1      | self.callable = self.load()
web_1      | File "scripts/openlibrary-server", line 82, in load
web_1      | load_infogami(self.config_file)
web_1      | File "scripts/openlibrary-server", line 57, in load_infogami

```

2020-10-08

<https://github.com/internetarchive/openlibrary/pull/3901>

1. Import + edit have known python3 issues @cclauss to create issues for both, please :)
2. Update setup_sh to include the `make git` and infogami pull
3. Create a gist for the latest setup_sh and remove from openlibrary repo
 - a. @mek we still need to finesse w/ @abezella to see either what of setup_web1.sh goes into ansible v. some olsystem script which gets run.
4. Haproxy add to pool w/ highly reduced settings (after #1 is in good shape)
5. Can we do a short test on Sunday? 9:30am? @mek to send invite

Here's a tracker!

<https://sentry.archive.org/sentry/ol-web/?environment=production&query=is%3Aunresolved%20runtime%3A%22CPython%203.8.5%22>

2020-10-01

- [x] Have merged some simpler PR ([#3857](#), [#3841](#)) and can test on dev0 and dev1
- [x] Updated <https://github.com/internetarchive/openlibrary/wiki/Deployment-Guide>
- [x] Drini: ./scripts/make-integration-branch.sh improvements

[internetarchive/openlibrary#3880](#) does 1. thru 7. except for 6. below.

TODOs @cclauss

1. ssh into ol-web1
2. Create a setup.sh file (because we're going to have to do this again, and some of these steps are just going to get pulled out into ansible + done for us)
3. Getting docker + docker-compose on `ol-web1` + whatever other dependencies are requires (let's please track them)
 - a. Add these installation steps for provenance into setup.sh
4. Git clone olsystem /opt/olsystem + openlibrary /opt/openlibrary
 - a. Add to setup.sh
5. Getting a production docker-compose file (refer to: docker-compose.staging.yml)
 - a. Getting nginx running -- Ask @cdrini for Q's
 - i. [x] nginx is installed on docker
 - ii. Symlink nginx.conf & sites-available from olsystem into Docker
 1. /olsystem/etc/nginx/nginx.conf → Docker:/etc/nginx/nginx.conf
 2. /olsystem/etc/nginx/sites-available → Docker:/etc/nginx/sites-available
 3. Start nginx
 - b. Configuration settings for start/restart
 - i. # workers, etc (See /olsystem/bin/upstart-service , openlibrary-server)
6. @mek (owns) check network access rules + add notes to setup.sh
 - a. Ferm rules
 - i. Ol-web1 is surfaced through some IP (port 443 / 80) -- mek (for testing: ssh -L 8984:ol-solr2:8983 -qAy cclauss@sshgw-sf.us.archive.org)
 - b. Hb_pga.conf
7. Add ol-web1 to Haproxy on ol-www1 (the haproxy is symlinked to /olsystem/etc/haproxy.cfg) -- ask @mek for Q's
8. Deploys + restarting docker service? Manual deploys for Thursday, then revisit

ol-www1% **sudo vi /olsystem/etc/haproxy/haproxy.cfg**

ol-www1% **sudo service haproxy reload**

- <https://openlibrary.org/admin?stats>
- <https://sentry.archive.org/sentry/ol-web>

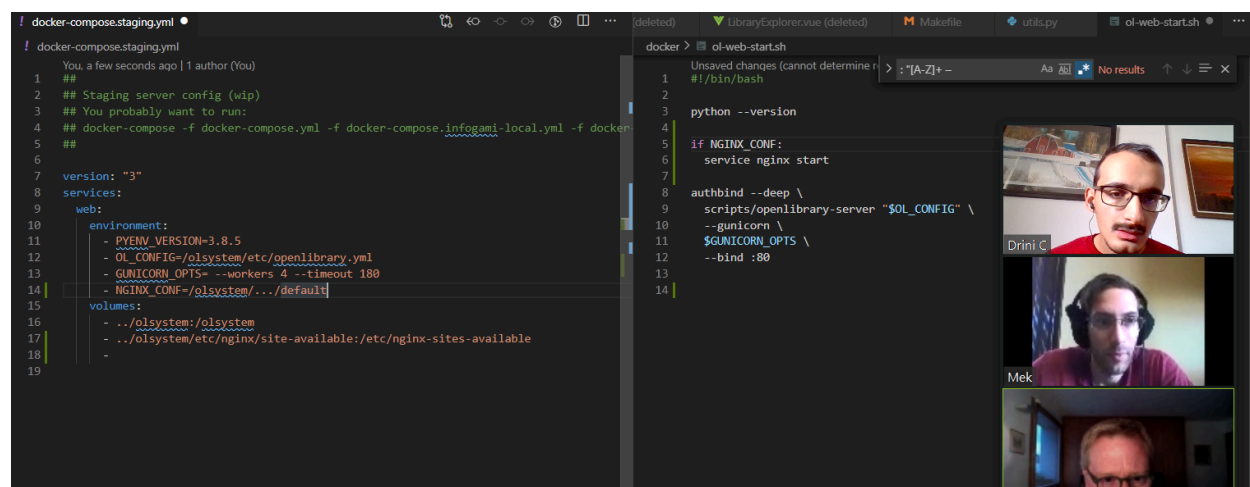
Useful screenshots:

```
drini@ol-dev0:~$ ssh -A ol-web3
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 4.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Plan your installation, and FAI installs your plan.

Last login: Thu Sep 24 22:40:15 2020 from ol-solr0.us.archive.org
drini@ol-web3:~$ cd /etc/nginx
drini@ol-web3:/etc/nginx$ ls
conf.d          mime.types      nginx.conf      sites-enabled
fastcgi_params  naxsi-ui.conf.1.4.1  proxy_params    uwsgi_params
koi-utf         naxsi.rules     scgi_params     win-utf
koi-win         naxsi_core.rules  sites-available
drini@ol-web3:/etc/nginx$ ls -al
total 64
drwxr-xr-x  4 root root 4096 Dec  3  2014 .
drwxr-xr-x 108 root root 4096 Sep 25 22:15 ..
drwxr-xr-x  2 root root 4096 Sep 17  2014 conf.d
-rw-r--r--  1 root root  911 Mar  4  2014 fastcgi_params
-rw-r--r--  1 root root 2258 Mar  4  2014 koi-utf
-rw-r--r--  1 root root 1805 Mar  4  2014 koi-win
-rw-r--r--  1 root root 2085 Mar  4  2014 mime.types
-rw-r--r--  1 root root  222 Mar  4  2014 naxsi-ui.conf.1.4.1
-rw-r--r--  1 root root  287 Mar  4  2014 naxsi.rules
-rw-r--r--  1 root root 5287 Mar  4  2014 naxsi_core.rules
lrwxrwxrwx  1 root root   30 Nov 19  2014 nginx.conf -> /olsystem/etc/nginx/nginx.conf
-rw-r--r--  1 root root  180 Mar  4  2014 proxy_params
-rw-r--r--  1 root root  465 Mar  4  2014 scgi_params
lrwxrwxrwx  1 root root   35 Dec  3  2014 sites-available -> /olsystem/etc/nginx/sites-available
drwxr-xr-x  2 root root 4096 Feb  4  2018 sites-enabled
-rw-r--r--  1 root root  532 Mar  4  2014 uwsgi_params
-rw-r--r--  1 root root 3071 Mar  4  2014 win-utf
drini@ol-web3:/etc/nginx$
```



Goal: ol-web1 by next Wednesday

- Ol-web[x]
 - (check) [Ferm](#) rules

- Make sure Docker is setup
- Nginx set up in Docker
- Open questions?
 - memcached?
- [Figure out minimum requirements](#) for running production ol-web[x] (this should be informed by **ol-dev1**)
 - @cclauss: Git clone olssystem, openlibrary onto the machine into /opt/openlibrary and /opt/olsystem
 - @cclauss: Apt-get install docker docker-compose
 - Please note these dependencies in some build script
 - Nginx (we're doing this because doing it is easier than removing, but not because we think it's necessary)
- Reverse engineer the deployment process to include ol-web1 in the pool: fab file
 - Postgres db config (hb_pga)
 - Haproxy add a definition of ol-web[x] (ol-www1's version of haproxy, lives in olssystem/etc/haproxy/conf.lol)
- Bugs
 - Changing covers?
 - StringIO issue may impact editing?
- Post ol-web[x]
 - Explore removing nginx dependency for ol-web[x]

Questions:

1. How do we test olssystem (esp. bss.py)? Where is it getting launched
2. Do we need two separate versions of acs4_py ?
3. [Do we need sudo everywhere?](#) And, ; *git rev-parse --short HEAD*
4. How do you have infogami set up inside openlibrary?
5. Mek: What is the best time (not Wednesdays) for a pair programming session

2020-09-24

Completed this week

[x] @mek grant @cclauss access to olssystem **Today**

[x] @cdrini to setup event for Th meeting @ 09 PT / 12 PT / 18 CET

[x] @cdrini See if we can give Mek & Drini & Chris sentry access

[x] @cclauss look at sentry errors

1. ol-web1 (current set up) Python 3
 - a. Add nginx to docker (which maps to the config in `/olsystem/etc/nginx/sites-available/default`` (or figure out how to deprecate, and be sure this is the right decision)

- b. Have ability to spin up production web node: Get the olsystem configs into/onto ol-web1 in a semi-automated way
 - i. Here is an automated example of spinning up a web node w/ production config settings:
 - <https://git.archive.org/www/openlibrary/-/blob/master/Dockerfile>
 - 1. @tracey and @mek built this custom Dockerfile which diverges from OL's dockerfile and this pulls in / git clones `olsystem` w/ all the right configurations to spin up a **production data** staging pod.
 - c. Getting gunicorn settings used on prod, embedded into a docker-compose.prod.yml
 - d. Add ol-web1 to pg_hba.conf (IPs acceptlist for postgres dbs) @mek
 - e. Add ol-web1 to sentry @cdrini
 - f. For (Thu) call...
 - g. **Ping Andy (holiday Oct 3) to reprovision ol-web1 to match**
 - i. Latest focal ubuntu -- **Done**: Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0)
 - ii. Docker installation into the ansible recipe
 - h. Add ol-web1 to the haproxy pool
 - i. Revise our deployment process to work w/ Docker
 - i. It would be nice if performing a deploy from ol-home resulted in an **async** trigger of a Docker rebuild on ol-web1 [et al] (which shouldn't be too hard to add to our olsystem fab file (for now, this seems like the right place). We'll also need a new mechanism to restart.
 - 1. docker-compose build web; docker-compose down; docker-compose up -d --no-deps webs
2. Olsystem-audit (required: because they import new python3 OL) must work as python2 or be patched (or at least issues identified)
 - a. Coverstore (covers.openlibrary.org)
 - b. Importbot supervisor?
 - c. Solr-updater code if any python2->3 required
 - d. olsystem/bin/cron
 - e. Look at ol-home logs to see if infobase has any more errors now that **its** using OL python3 (ol-home isn't in sentry and we could be introducing problems without knowing it) -- **ol-home:/var/log/openlibrary**
 - i. Infobase
 - ii. Import-bot
 - iii. Solr-updater
 - iv. upstart
 3. @cclauss owning OL PRs for Python
 4. As we're blocked... there's also OPDS/bookserver, BSS lending (python2 -> 3), ACDC
 - a. BSS https://github.com/internetarchive/acs4_py/blob/master/bss.py
 5. Open questions

2020-09-21

- Break last week 🌴
- StringIO undo <https://github.com/internetarchive/openlibrary/issues/3806>
- ☒ Dbs moved over
- Aasif PRs
- Testing something around infobase
- Chris access to vms
- Sentry in limbo; Tracey is looking into it

Action items:

[] Chris: Get access to archive.org vms

- List of vms:

<https://github.com/internetarchive/openlibrary/wiki/Production-Service-Architecture>

[x] Mek: Give Chris crash-course on deploying to dev and staging

[x] Mek: Send ssh command to Chris

[] Chris: StringIO undo <https://github.com/internetarchive/openlibrary/issues/3806>

[x] Drini: Ping Brenton about Sentry

[] Drini: setup nginx on ol-dev1 and via docker (web)

[] Drini: move IP addresses to `extra_hosts` in `docker-compose`

SSL needs to be setup for gunicorn on dev1 (or nginx, see `olsystem/etc/nginx/sites-available`)

2020-09-09

Onboarding notes

- @judec + @jye: TTScribe
- @judec + @brenton: Adobe Digital Editions
- @cr + @merlijn: ArchiveCD
- @mek + @charles: [OPDS](https://bookserver.archive.org) - json standard for syndication books (<http://bookserver.archive.org>) needs python3 porting

Next Steps for Staging:

- SSL needs to be setup for gunicorn on dev1 (or nginx, see `olsystem/etc/nginx/sites-available`)
- setup nginx on ol-dev1 or via docker (web)
- move IP addresses to `extra_hosts` in `docker-compose`

Want a new a ol-web server to be with new code - Andy, @abezella

OPDS - JSON book server; needs py3 porting. Could take like a week, needs to get done and would be really useful; https://en.wikipedia.org/wiki/Open_Publication_Distribution_System
TTScribe - software that powers book scanners
Adobe Digital Editions - How people can access books offline
Archive CD
(edited)

EVERYTHING ABOVE THIS POINT IN REVERSE CHRONOLOGICAL ORDER

2020-07-13

- Chris doesn't have access to olsystem
- Test does infogami master work with OL py2? -- Done in Travis CI
- Production path to infogami

Action items:

- [x] Chris: Travis "box" that runs infogami-master with py2 (needs to be allow_failures)
 - <https://github.com/internetarchive/openlibrary/pull/3579>
- [x] Drini: Take a look at <https://github.com/internetarchive/openlibrary/pull/3411>
- [x] Chris + Drini: Env variable to override which `git submodule`
- [x] Drini: update cal event with zoom link + repeat
- [x] Drini: Review 3579 and move to dev and then prod

2020-07-16

- [x] Chris: [Tracking issue](#) updated with the current status and TODOs

2020-07-19

Travis CI is looking more comprehensive and much greener!!
See: <https://travis-ci.org/github/internetarchive/infogami/builds/709760105>
This is based on the PR <https://github.com/internetarchive/infogami/pull/87> which incorporates all of #91 plus much more. #87 Enables all Infogami tests that pass on Py2 also pass on Py3!! (There are lots of tests that fail on Py2 and they also fail on Py3.) Even after fixing Infogami to pass the valid tests on Py3, the Open Library master tests still pass on that Infogami. Nice!

2020-07-20

Worked with Drini to get Firefox and Docker working well together.

<https://github.com/internetarchive/openlibrary/pull/3598> will allow us to launch the servers in Docker on Python 3 and get through the boot-up process and display the home screen without raising any Python exceptions. Clicking a book will raise an exception. Added a few small Infogami PRs <https://github.com/internetarchive/infogami/pulls/cclauss> to pass more pytests.

2020-07-23

Action items:

- [x] Mek: Get ~~dev1~~ `dev1`.openlibrary.org hooked up
 - <http://staging.openlibrary.org:8080/status>
- [x] Drini: Get docker on ~~dev1~~ `dev1` staging up-to-date
- [] Drini: add dummy data into it
- [] Mek: Create a testing guide spreadsheet
- [x] Drini: Show debugger flow
- [x] Chris: Test/update web.py (py3 only)
 - <https://github.com/internetarchive/openlibrary/pull/3619> fixes **def with** exception.

2020-08-06

Would it help debug to have better repr() strings for things, books, works, editions, templates?

Discuss 127.0.0.1 vs. localhost PR #3664

How do I create `site` for a doctest?

https://github.com/internetarchive/openlibrary/blob/1a6e364c581b0d4933189178618c36837a2e9136/openlibrary/plugins/upstream/tests/test_merge_authors.py#L244-L251

How do we get Python 3-based OL onto staging?

[x] Chris: Give use the branches you want for infogami/ol python3 on both

[x] Drini: Update the state of ol-dev1

[x] Drini: Investigate ol-dev1 sluggishness - seems to have gone away?

Action items: <https://github.com/internetarchive/openlibrary/issues/3333>

- [x] Chris: Spend more time reviewing Requests [PR #3658](#)
 - Let's identify the majority which can be
- [x] Drini: Where is debug timeout?
- [x] Drini: Better solution for vendor/infogami dirty [PR #3642](#)
- [] Mek: Dig into **get_sorted_editions()** [Issue #3633](#) → [PR infogami#107](#)
- [] Mek: Push for rebase of [infogami#78](#) and [infogami#91](#)
 - Git checkout -b new-branch
 - Git pull fork-link branch-name
 - Git rebase -i master
 - Git push fork-link HEAD:their-branch-name -f
 - OR make a new branch, and push that onto your fork after the rebase

- [x] Reviews please on [infogami#104](#) ← Python 3 integration branch

Things todo:

- [x] Fix the broken editions table [PR infogami#107](#)
- [] scripts/ directory
- [] Scripts in olsystem (Mek/Drini)

Questions:

- Is coverstore running the latest code?

Test-list

- sorted-editions
- Testing imports (basic level)
 - MARC
 - Ask @charles for help on MARC tests Python 3
 - JSON
 - IA import
 - ISBN endpoint?
- Borrowing
- Adding/removing covers
- Infogami editing + saving
- Test cover service (on Python 3) -- separate deploy
- Adding to list
- Adding to Reading log
- Account registration/login
- SolrUpdater

2020-08-13

This call:

- [x] Get_sorted_editions [Issue #3633](#)
 - Seems like old-style; `class A:`, is no longer supported in py3, and we might not be able to override `__class__` 🤔
 - Want some good Thing tests before we start trying to tweak this
 - Can we merge [PR infogami#107](#) as a temporary fix so we can test more things, until we find the proper solution to this?
- [x] Review #333
 - Punt for now: scripts/ directory, scripts in olsystem (Mek/Drini); all scripts should be py2/3 compatible and called by e.g. ol-home with py2
- [x] Acs4py?
 - We want to remove it, but in the past, when we tried to remove it, it caused something to break
 - We might want to sync with Jude at some point later

- [x] Charles' work; positive changes, but the tests are still failing
- [x] Infogami commit sha in openlibrary
 - Switch that back to the production
- [x] Who are the testers for August (pre-prod data)?
 - Us, Nolan, maybe Jeff
- [x] Goal for end of August:
 - staging.openlibrary.org py3 + docker connected to production data
 - Infogami master deployed to production on py2
 - We will need something hooked to prod data + infogami master + py2
 - Asheesh might help give us some pointers on best way to test
 - For first version we will likely leave infobase on ol-home running with py2

Action items:

[x] Mek: review + merge <https://github.com/internetarchive/openlibrary/pull/3671>

[>] Drini: review <https://github.com/internetarchive/openlibrary/pull/3682> (lower priority)

[x] Chris: Continue reviewing Aasif's PRs

[>] Chris: Push for rebase of [infogami#78](#) and [infogami#91](#)

- Git checkout -b new-branch
- Git pull fork-link branch-name
- Git rebase -i master
- Git push fork-link HEAD:their-branch-name -f
- OR make a new branch, and push that onto your fork after the rebase

[x] Drini: Switch that back to the production - 7595

[x] Drini: Try to place a staging env for infogami master + py2 somewhere (dev probably)

2020-08-20

This call:

- [x] __class__ failing
- [] Go over Aasif's PRs that Chris has approved; let's not leave him hanging! Let's land at least one to show we're making progress.
- [x] Open PRs on Infogami
- [] Open PRs on Open Library
- Remove Acs4
 - @mek Ping jude + brenton
 - grep ol-home for acs4
 - Remove it;

Class nonsense on infogami:

- Morphing a class from one class to another is breaking us

Shadow (shallow wrapper class) that gets instantiated/blown up into a full class

Create real class with asdict property that can create dict

Options:

- LazyObject
- Stick close to what we have
- All sub-things are currently failing
- `list(edition.works)[0]` -> this forces it to load all the works

Test harness:

- `thing = Site.get('/books/OL1M', lazy=True)`
- `thing instanceof Edition == false`
- `thing.title`
- `thing instanceof Edition == True`

We can try setting `lazy=False` -> perf implications

Next steps `__class__`:

- **Ping Asheesh**
- Create test so we can quickly see when this is failing
- Find minimal example that reproduces the bug
 - Maybe the fact that we have 2 levels of inheritance?
 - Maybe one of the classmethods? static methods? Property methods?
- Maybe ping on python IRC to find a reproducible example?

Action Items:

[] Drini: review <https://github.com/internetarchive/openlibrary/pull/3682> (lower priority)

[] Chris: Push for rebase of [infogami#78](#) and [infogami#91](#)

- `Git checkout -b new-branch`
- `Git pull fork-link branch-name`
- `Git rebase -i master`
- `Git push fork-link HEAD:their-branch-name -f`
- OR make a new branch, and push that onto your fork after the rebase

[] Mek: Ping Jude + Brenton about removing acs4

2020-08-21

Landing [infogami#112](#) was MASSIVE!!

[openlibrary#3584](#) is no longer a Draft/WIP and it fixes **all remaining Pytests** on Py3 and does not open any new ones. **Please review as our highest priority for Py3.**

Essential Python 3 PRs to review are:

1. [openlibrary#3584](#)
2. [infogami#108](#)
3. [openlibrary#3708](#)

Python 3 **issues that need help:**

1. [openlibrary#3687](#) (JavaScript syntax error)

Once [openlibrary#3584](#) lands, please update **staging** with that (w/ or w/o [infogami#108](#)).

2020-08-22 Saturday evening

[x] Cannot add a list to a range [internetarchive/infogami#116](#)

[x] Python 3: Make document a cached_property [#3728](#)

[x] Fix-pytests-for-openlibrary/records-again [#3708](#)

[] Python 3: Add Cover Image hangs [#3687](#)

2020-08-24 Monday evening

[x] Python3: Sort Publish date unknown [#3731](#) or [#3734](#)

[] Fix Python 3: Add Cover Image hangs [#3733](#)

[] Update `staging` and encourage people to test it and open issues

[] What is your recommended way to find more problems with the port?

[] What is the best way to test the API, etc.?

2020-08-25 Tuesday afternoon

- [x] Python3: Sort Publish date unknown [#3731](#)
- [] Fix Python 3: Add Cover Image hangs [#3733](#)
- [x] Py3: Fix dictionary changed size during iteration again [internetarchive/infogami#117](#)
- [x] Add 16 pytests to make test-py [#3735](#)

2020-08-27 Thursday afternoon

This call:

[] How to find new problems! What is the best way to test the imports + the APIs, etc.?

[] What is your recommended way to find more problems with the port?

Start migrating staging.openlibrary.org config to prod

[] Xauth for IA login

[] Amazon keys

[] Sentry

docker-compose up web -d

Order:

To migrate: web[34], www1, covers1, home,

1. Staging (dev1)
2. web heads (web[34])
3. Covers1

Not particularly planned:

4. Home

No Python: www1(?), dbs, memcache

- Scripts used by olsystem: issue [#1449](#) → pull request [#3753](#)
- Import docs:
<https://github.com/internetarchive/openlibrary/wiki/Developer's-Guide-to-Data-Importing>

Action items:

[] @cdrini See if we can give Chris sentry access

[] @mek Check if www1 is using Python/OL deploy of it

- o [] @mek Fix Python 3: Add Cover Image hangs [#3733](#) <- last item on UI
- o Put on dev + staging @mek to merge

[x] @cdrini Make docker web service *only* run gunicorn; place infobase/etc in another service

[x] @mek Start moving staging.openlibrary.org config to prod

- After infobase pulled docker:web

Instances cheatsheet 09 Sept 2020

Instance	Hypervisor	OS	Primary_node	Status	Memory
ol-web3.us.archive.org	kvm	fai+trusty	ia600009.us.archive.org	running	12.0G

ol-acs-db2.us.archive.org	kvm	fai+xenial	ia600009.us.archive.org	running	4.0G
ol-acsl.us.archive.org	kvm	fai+xenial	ia601100.us.archive.org	running	32.0G
ol-db2.us.archive.org	kvm	fai+trusty	ia802105.us.archive.org	running	16.0G
ol-mem3.us.archive.org	kvm	fai+xenial	ia803303.us.archive.org	running	10.0G
ol-web4.us.archive.org	kvm	fai+trusty	ia803303.us.archive.org	running	12.0G
ol-dbl.us.archive.org	kvm	fai+trusty	ia803303.us.archive.org	running	16.0G
ol-solr2.us.archive.org	kvm	fai+trusty	ia803303.us.archive.org	running	16.0G
ol-covers1.us.archive.org	kvm	fai+trusty	ia803303.us.archive.org	running	4.0G
ol-www1.us.archive.org	kvm	fai+trusty	ia803303.us.archive.org	running	4.0G
ol-acs-dbl.us.archive.org	kvm	fai+xenial	ia803303.us.archive.org	running	8.0G
ol-home.us.archive.org	kvm	fai+trusty	ia803303.us.archive.org	running	8.0G
ol-mem4.us.archive.org	kvm	fai+xenial	ia803306.us.archive.org	running	10.0G
ol-mem5.us.archive.org	kvm	fai+xenial	ia803306.us.archive.org	running	10.0G
ol-solr0.us.archive.org	kvm	fai+xenial	ia803306.us.archive.org	running	16.0G
ol-web1.us.archive.org	kvm	fai+xenial	ia803306.us.archive.org	running	8.0G
ol-backup0.us.archive.org	kvm	fai+trusty	ia804002.us.archive.org	running	4.0G
ol-dev0.us.archive.org	kvm	fai+trusty	ia805200.us.archive.org	running	4.0G
ol-dev1.us.archive.org	kvm	fai+xenial	ia805203.us.archive.org	running	4.0G