

Making Good Decisions:

**Taking a Hyrax Application From Development to
Production**

Who we are

Captain/Moderator: Sadie Roosa (she/her),
WGBH

Crew Members/Panelists:

Mark Bussey (he/him), DCE

Jason Corum (he/him), WGBH

Andrew Myers (he/him), WGBH

Henry Neels (he/him), WGBH

What we're talking about

- Brief overview of WGBH's use case
- The steps WGBH took to get Hyrax running in production
- DCE's experiences with running Hyrax in production
- Approaches the community can take to sharing experiences and providing more centralized documentation around this topic

WGBH's Use Case

- Metadata only, describing AV files stored externally
- Very hierarchical data model
- Batches of 100s to 1000s of records ingested at a time, rarely have items deposited one by one

DCE's use case(s)

- Internal and Client production systems
- Deployment pipeline management
- Metadata & Content, wide variety of content types and data models
- Generally flat metadata models
- IR - ETD repository with individual deposit and approval workflows
- Special collections - Batches of 100s to 1000s of records ingested at a time, sometimes individually remediating items post-ingest
- Coexist with existing IT practices and standards (OS / Database / Infrastructure)

Step 1

Captain's Log, Stardate 72160.4: We begin our mission into the undocumented territory of setting up a Hyrax instance in a production environment. We've gathered evidence that other crews have been able to succeed in similar missions, but, without being able to recover their precise mission plans and logs, we have to forge ahead on our own. It's comforting to know we'll have the support of other Samvera crews as long as we stay within communications range.

Step 1

Talking Points

- Starting Up
- What are the required resources to think about hosting?
- What are factors that might affect the decision to go with various hosting options?

Step 2

Captain's Log, Stardate 72516.5: Having successfully engineered machines to run our Rails app, Solr, and Fedora, we begin the process of generating records in Hyrax. As the number of ingested records increases, we have started receiving bug reports and error messages mentioning an unknown entity. Perhaps the situation will become clearer when the crew can determine the nature of our latest discovery: Modeshape.

Step 2

Talking Points

- What are some of the underlying systems that you didn't know about when you started?
- What methods did you use to figure out what was causing these bug reports and error messages?

Step 3

Captain's Log, Stardate 72618.2: We've received intelligence from our Samvera network helping us to identify Modeshape and ascertain its relationship with Fedora. It also seems we were operating under the false assumption that the default file-backed method in which we were running Fedora in our development environments is the optimal configuration for Hyrax. Upgrades to run Fedora with a SQL-based database persistence layer are underway.

Captain's Log, Stardate 72620.2: With upgrades nearly complete, we must now decide if we can run the new Postgres database Fedora will use on the same machine as Fedora, or if there are benefits to running it separately.

Step 3

Talking Points

- What's the difference between what you were running by default from your development environment (filesystem) and what we decided to run in production (SQL-based database)?
- What additional decisions did you have to make once you realized you needed to change the way Fedora was configured?
- What affected the decision of where to run the Fedora database?

Step 4

Captain's Log, Stardate 72820.9: The application has been running for weeks with Fedora's Postgres database in an RDS instance. The crew has begun to complain that the regular maintenance tasks are taking an unreasonably high percentage of their time. Although the database is supposed to be self-maintaining with autovacuum features, in order to keep things running smoothly, we have to send a crew member in between each batch of 1000 records to manually run a VACUUM FULL, or else the database fills up the disk space and ceases to operate.

Step 4

Talking Points

- What were the problems you had with the Postgres Database, and how did that affect the project overall?
- How/why did you decide to go with Postgres first, and how/why did you switch to MySQL when you did?
- What other issues to did you encounter during this stage (hint, hint: HikariPool defaults)?
- How is MySQL working so far?

Step 5

Captain's Log, Stardate 73001.3: The switch to MySQL as Fedora's database has proven to be a fruitful choice. The crew has been able to focus attention on other tasks, and the database seems to be behaving itself, at least in terms of its rate of growth. However, the rate of errors during batch ingests remains high. I've assigned a research team to investigate if the system has the capacity to run ingest jobs concurrently. We may need to revert to processing one at a time.

Step 5

Talking Points

- What assumptions did you make around concurrency when you developed the `hyrax-batch_ingest` gem?
- What issues have you run into in terms of running jobs concurrently?
- What, if anything, have you been able to do to make concurrency work better?

What can we share and how can we share it?

- Benchmarking
- Docker images
- Cloudformation/Terraform/etc. templates
- Diagrams
- Tutorials and walkthroughs
- Narrative sharing (something more permanent than slack or email, like a blog?)

- DCE's Bookshelf
 - [Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation](#)
 - & [The DevOps Handbook](#)

- Plug for the [infrastructure working group unconference session](#)

Thank you!

Contact:

sadie_roosa@wgbh.org

jason_corum@wgbh.org

andrew_myers@wgbh.org

henry_neels@wgbh.org

mark@curationexperts.com