Coordinating Fedora Instances

Plan for coordinating instances

There are three Fedora repositories, "development", "testing" and "production". The production repository always contains the master copy of the data. The contents of the development repository are never considered authoritative; they are only there for supporting development/testing of new features. The testing repository does not always exist, but contains a snapshot of content from the development repository and/or the production repository for purposes of performing user tests. Specialized projects, like Evia, may have their own development repository. Individual developers may maintain their own development repository.

The repositories are deployed on machines as best fits their purpose. Typically, the development repository will run on a development machine, and the production repository will run on a production machine. We may eventually purchase an additional machine for the testing repository, but for the time being, the testing repository will run on a development machine.

All features are deployed on the development repository for testing. The development repository is considered unstable, and may be brought down at any time to install/test a new feature. If the development repository does not contain relevant data to test the feature, data may be copied from the production repository, or new test objects may be created. Any new features should be tested to ensure they do not break existing features or corrupt existing data.

All data is ingested to the development repository while a collection is first being developed. This is to ensure that the production repository does not become littered with test data. Once the collection has been deemed stable, it will be moved to the production server. When the collection is stable, the Import Tool can be used to move it to the production repository.

Incremental updates to a published collection will be performed directly on the production repository. This allows users of the Image Cataloging Application to see the results of their updates immediately. It also allows us to better track changes to the collection. Any changes that will affect a large portion of the collection should be tested on the development repository before being applied to the production repository.

Process for testing individual tools

- All tools (test and production) typically point to the production versions of other tools.
- When a tool is being tested, it may temporarily be pointed at non-production versions of other tools, but this should only occur when:
  - the tool being tested relies on test data that is not available from a production tool, OR
  - the tool being tested will make modifications to data held by other tools
- In general, all tools that need to be modified are copied the development machine. This allows them to be restarted, but they may still retrieve data from production tools/services.
- Fedora and the tools shipped with it are treated as a single piece. For example, FedoraGSearch always points to the Fedora it is bundled with.

Open questions:

1. How can we keep "in progress" items from ending up on normal user's screens? For example, in a published collection of books, we don't want to display a book that only has half of the pages digitized. We will need to support multiple worflows for digitizing/cataloging objects while maintaining appropriate state information.
2. Will large ingests have an impact on the performance of the production repository?
3. What kind of load will the preservation system put on our production repository? Probably not much, because it will spend most of its time waiting for retrieval of files from HPSS and processing the files.
4. How can we best separate statistics for "machine use" of the repository (like our preservation integrity service) from regular use statistics? By IP?
5. Is it possible to do a "quick ingest" by loading data to a temporary server, and copying the resultant LowLevelStore directory to the production directory? We would need to coordinate this with targeted database updates, and it seems fairly risky, but it would be a huge time savings.
6. Does this proposal meet Michelle's testing needs?