2019-02-08 - System Operations and Management SIG
Agenda and Notes

Date
08 Feb 2019

Attendees

- Ingolf Kuss
- Todd Wallwork
- Robert
- Florian Ruckelshausen
- Greg Delisle
- Brandon Tharp
- Stephen Pampell
- Patty Wanninger
- Dale Arntson
- Jason Root
- Chris Creswell

Goals

Discussion items

<table>
<thead>
<tr>
<th>Time</th>
<th>Item</th>
<th>Who</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Welcome</td>
<td>Ingolf</td>
<td>Determine note taker: Ingolf</td>
</tr>
</tbody>
</table>
| 15   | Review goals for 2019 Folio meeting | All | Review SysOps lines in this spreadsheet: https://docs.google.com/spreadsheets/d/1JELsHTfu_xYA1yTsyRCorZzEv0dBNjsg1Lo32rm_3s/edit (to be presented to the Product Council)  

  Meeting Notes:  
  Place and time of the meeting has not yet been decided. PC is still working out what is possible and what is in scope. As yet, nothing to plan for us. Group acknowledges SysOps f2f-goals as laid out in this document: https://docs.google.com/spreadsheets/d/1JELsHTfu_xYA1yTsyRCorZzEv0dBNjsg1Lo32rm_3s/edit. |
**Sharing Q4-2018 release experiences**

**Discussion Summary**
- Proposal to move dependency resolution out of Okapi into a different tool
- We need a separate build tool, a tool to build the backend outside Okapi. We need to strictly separate the build and run stages.
- Need to impose these as guiding principles: [https://12factor.net/](https://12factor.net/)
- We need a versioning of database schemas in order to be able to upgrade modules.
- We need a disaster recovery perspective. A way to rollback changes. We need to be able to start up the system from a (cloud application) backup.
- Maintaining a principle of decoupling of the modules
- Data integrity needs to be ensured from release to release.

**Meeting Notes:**

Stephen: Installation experiences. How to effectively stand up a Folio instance? Dependency resolution is a nightmare.

Chris Creswell ran into blocking problems with a filter issue (could solve), see #sys-ops channel of Feb 7.

- Proposal to move dependency resolution out of Okapi into a different tool

Biggest grief: Installation with a "simulate" account for Okapi. Okapi gets a list of frontend modules and will spit you out what backend modules it needs. With "simulate=true" you have to pull everything that is in the IndexData repository. Want to get away from that, want to be able to use one's own repository.

What do we need to build the backend? There need to be a tool to build the backend outside Okapi. We need a separate build tool

See the discussion in the #tech-council channel (Stephen, Jakub et.al. on Feb 6)

OTS made an assessment of FOLIO architecture and submitted to the PC on Jan 11. We need to push forward these principles for cloud native apps for Folio: [https://12factor.net/](https://12factor.net/), especially principle 5: separate build and run stages. Need to strictly separate the build and run stages. At the moment, Okapi is used for the run and the build stages.

Problematic: The dependencies have not been worked out yet, until you run the system. It's a dependency resolution issue.

Dependency resolution was a fixed issue in the Q3/Q4 releases, but as the system gets bigger, most libraries don't want to employ the whole platform-complete system, but rather pick a set of modules which suits their institution's needs. Exchanging a single module from an older to a newer version in the running system (while maintaining referential integrity) is another use case to be considered. It's an issue of upgrading modules and the versioning of database schemas. We would like to upgrade this one module, but don't want to upgrade Okapi.

- Need a versioning of database schemas

Right now, there is no way to go from release to release other than setting up a complete new system.

In future, single VM installation will not be an option anymore, as RAM requirements will be expected to go to 32GB, which will be huge virtual machines.

Question is: What stripes modules perform what functionality? The number of Stripes modules is smaller than the number of backend modules, maybe this is why one first has to install the Stripes modules and these then pull (a large number of) backend modules that they need. (Ingolf: what is the criticism/suggestion here? To be able to deploy the backend modules first?)

- Rolling back of changes – need a disaster recovery perspective. Can I start up the system from a (cloud application) backup? (a cloud application backup is not what one traditionally thinks of as a backup)
- Maintaining a principle of decoupling of the modules
- How to ensure data integrity from release to release?

Stephen: One possibility is to generate module descriptors from the github repository of Folio. But the current state of dependency resolution is antithetic to that goal.

Dale: Test cross-module interoperability

**Deployment strategies:**

Chicago (Dale): Kubernetes bundles

Cornell (Robert): Kubernetes

**Action items**