

Design - Multitenancy

Background

The [Hydra-in-a-box project](#) is a grant funded project with several goals, one of which being to implement a hosted service using Hydra software. Being that Fedora is a critical component of the Hydra stack, there is a clear interest in using Fedora to support the storage needs of the hosted service. The following discussion considers how some of the requirements of the Hydra-in-a-box service may impact Fedora.

There are many assumptions included in the comments below. The purpose of this page is to begin discussion of the included topics.

Goals

- The Hydra-in-a-box hosted service would like to deploy a single architecture which will be used to support a large number of institutions using the Hydra-in-a-box software
- As one of the components in the Hydra stack, it would be very helpful if Fedora could be deployed such that:
 - Fedora can be scaled up and down to handle varying levels of request load
 - Fedora can handle the content of multiple distinct accounts, where the users of each account interact with Fedora without needing to be aware that other accounts exist
 - Fedora allows for accounts to be added and removed as needed
 - Fedora allows the binary content stored for each account to be stored in a distinct location

Implementation Notes

The goals listed above define two distinct but overlapping concerns: Scaling and Multi-tenancy

Scaling

- In order to scale up effectively, will need to be able to add compute capacity and distribute load with load balancing. This suggests that clustering of the Fedora instances will be required.
- In order to be able to add and remove compute capacity efficiently, the storage of assets must be in a persistent store outside of the compute resources.
- A shared persistent store is preferred, so that once a file is written, it is available to all other instances in the cluster without having to be written again at each node.
- The obvious shared persistent store in the AWS environment (where the Hydra-in-a-box hosted service will be deployed) is S3.
- Storage of files through Fedora going to S3 suggests the need for a [ModeShape binary store implementation for S3](#).
- The object metadata of Fedora will be written to a relational database for ease of querying and performance and because Modeshape only supports using a relational database as the object storage location in a cluster

Multi-tenancy

- In order to handle multiple accounts, there is a need to have distinct object graphs and associated binary storage for each account
- Two potential options for managing accounts are (1) ModeShape workspaces or (2) a distinct root node created for each account (with appropriate access controls)
- Fedora will need to be aware of the fact that there are multiple accounts so that the API can expose the option to specify an identifier which would distinguish between the accounts/tenants
- Need to be able to configure distinct binary storage locations for each account (S3 accounts or buckets)
- Need to be able to add and remove accounts on-the-fly (without requiring a restart to pick up new configuration) - this may impact how account division is implemented