HZB Helmholtz Zentrum Berlin

# A Docker Image for ICAT and Deployment at HZB

Rolf Krahl

ICAT F2F Meeting, 04 May 2023, Berlin

# ICAT Docker Image: Considerations

- Separation of Docker image and configuration. The image should be generic.
- The ICAT image is based on a matching GlassFish / Payara image.
- One image for all ICAT components: `authn.anon`, `authn.db`, `authn.ldap`, `authn.simple`, `authn.oidc`, `icat.server`, `icat.lucene`, `icat.oaipmh`, `ids.storage_file`, `ids.server`, `topcat`. Runtime configuration controls, which components will be in use.
- Multiple versions of the image, one for each `icat.server` minor version (4.4, . . . , 4.11, 5.0). The other components "matching in time".

# ICAT Docker Image: Build

- The underlying GlassFish / Payara image is built with the distribution unzipped into `/opt`. `GLASSFISH_HOME` is set to the corresponding directory.
- The `domain1` from the distribution is deleted. A startup script will automatically create it when the container starts.
- The release distribution for each ICAT component is unzipped into `$GLASSFISH_HOME/apps`.

# ICAT Docker Image: Runtime Configuration

- The runtime configuration for ICAT is expected to be added to the container as a volume in `$GLASSFISH_HOME/etc/icat`
- It should contain one text file `APPS` that simply lists all components to be deployed and one subdirectory for each component.
- The container startup script will copy all files from the component directory in the config to the corresponding application directory and then run `./setup install` in the application directory.
- Optionally, a `filter.sh` shell script may be placed in the config directory to filter the files on the fly while they are copied to the application directory.

```
$GLASSFISH_HOME/etc/icat/
├── APPS
├── authn.anon/
├── authn.oidc/
├── authn.simple/
├── filter.sh
├── icat.lucene/
├── icat.oaipmh/
├── icat.server/
├── ids.server/
└── ids.storage_file/
```

# Deployment at HZB: Environment

- Two dedicated server, RHEL 7.
- Operation in active/passive mode, e.g. the full stack is running on the active server, while the passive one is idle.
- Switch the server manually between active and passive role. No high availability (yet).
- Have an extra IP address for the ICAT service that is moved between active and passive server in order to move the service transparently for the user.
- Use `systemd` unit files to control starting and stopping the service (and a bunch of related stuff).

# Deployment at HZB: Orchestration

- Very simple orchestration using one docker-compose file.
- Make heavy use of volumes to add runtime configuration to the container, to keep the databases in the host file system (and thus having them included in the backup) and to provide access to storage.
- Have all ICAT components in one single container.
- Container that make up the service:
  - `icat`: the Payara application server running ICAT,
  - `mysql`: the database backend (actually MariaDB),
  - `datahub`: ESRF Datahub (e.g. Logbook),
  - `icatplus`: ESRF ICAT+, needed for Datahub,
  - `mongodb`: Mongo DB, needed for ESRF ICAT+,
  - `apache`: Apache HTTP Server acting as reverse proxy for all services and running a few more web service scripts,
  - `smtp`: a postfix mail server to allow containers to send out mail,
  - `keycloak`: Keycloak SSO (testing, not really in production).
- Only the `apache` is reachable from outside.

# Deployment at HZB: Client Image

- Have an extra dedicated `client` image for variuos maintenance tasks.
- Timers run the `client` for regular tasks, such as import of proposals from the useroffice.
- Run the `client` interactively for admin duties.

# Outlook

- Consider to move to Kubernetes in the future.
- Main goal: make a step towards high availability.
- Would require to run the database backend as a cluster.
- Option to split ICAT components into separate containers and also parallelize some of them.
- Still need a lot to learn on Kubernetes, though.

# References

- Using Docker with ICAT. Presentation at ICAT F2F 2016 in Copenhagen.
- rkrahl/icat. Docker image on docker hub.
- icatproject-contrib/icat-config. Git repository with configuration files from each ICAT component. Suitable as a starting point to build your own runtime configuration directory.
  (Do not push your own configuration to GitHub!)