

Magnolia Docker cookbook

DRAFT

There may not be such thing as a ready-made, one-fits-all Magnolia container image; yet there is a common pattern.

This is what we summarize in a few steps, in this cookbook. It is not perfect admittedly, but fairly straightforward.

Magnolia Docker cookbook

- Pick an image to start [FROM](#)
 - A good choice is a [tomcat](#) image. It has the least setup hassle.
 - Alternatively [AdoptOpenJDK](#) images are gaining popularity
 - they exist in various versions and OS/distros (alpine 🚩, debian, ubuntu, ubi (redhat), windows);
 - Then provision tomcat or alternative app server
 - be it via package-managers, plain download or `COPY --from`
 - Lastly, [distroless](#) images might be an option too, in case no package manager nor shell is required; beware managing native dependencies can become challenging.
- Configure Magnolia environment, best via tomcat's `setenv.sh` & `CATALINA_OPTS`
 - set magnolia.properties as system properties e.g. `-Dmagnolia.repositories.home` , `-Dmagnolia.resources.dir`
 - set JVM options, such as memory settings, also as `CATALINA_OPTS`
- Consider Docker [volumes](#) for the repository (Jackrabbit index), light-modules, content-importer if needed
- Provision JDBC drivers if you are using an external DB
- [Include](#) your Magnolia webapp war
 - include project-specific config (publishing key-pairs, receivers config)

Upcoming plans

While Magnolia **currently does not offer/support any official Docker image**, we intend to improve Magnolia developer & DevOps experience with Docker:

- first off, provide *official* guidelines and documentation; this cookbook is a start 🍴, likely also for composing an author/public setup
 - clarify the existing, non-supported, Magnolia Docker artifacts on [GitHub](#) and Docker Hub
- consider a default image suitable for light development
- facilitate externalized configuration (whether YAML and/or properties files, shouldn't matter)
- we do not foresee publishing a large array of Magnolia OS/JDK version/vendor combinations (rather few build arguments instead), which is why a cookbook matters.

Feedback

The cookbook certainly misses few tips and tricks, pitfalls, etc. Do let us know in the comments so we can improve!

Known issues

- DX Core on alpine doesn't work currently: native libs required for AI-based functionality are incompatible. CE should work.