

Concepts for Developer Support

Developer needs

- Tools - coding, building, testing, deploying, debugging
- Code - the sources of the application or service being built
- Run time - to deploy/test/debug/teardown
- Workflow - a process to develop, test, and publish the developed software

Development in the container context

Development in the container world is typically cli based using a **build container** that contains the tooling support for a specific artifact. The ndslabs /developer-shell is an example, it contains the tooling to build the ndslabs system and to deploy a small-scale test environment.

Issues

- There are no known *comprehensive* integrated development environments that are cloud-native. There are cloud-native editing tools, and some early IDE offerings primarily based around web developers and nothing that is container oriented.
- Tooling and workflows vary greatly among the various languages, runtimes, and communities. There are no definitive toolsets even within development groups working on a single project.
- Developer tools and processes are highly opinionated - there are many specific custom toolsets and processes for achieving the same goal
- Crossing the GUI boundary is an issue,

Meanings

- It is impossible to provide tooling for every use, although we can provide some basic development bootstrapping tools such as code editing
- Standalone IDE's are not web-ready, they must be used in a "machine" context, although that context could be in a container.
- The cloud-native GUI (html) does not simply match with native IDE local GUI models (X11, windows)

Design

- Tools/Stacks are published to catalogs
- Catalogs are searchable, sharable, and any number can exist
- Each user has one or more toolboxes - populated from catalogs
 - A toolbox is just another catalog
- Tools/Stacks Can be Configured - apply specific parameters
- Configurations can be saved and become attached to the tool in the catalog
- A Configuration can be launched - becomes one of the user's running tools
- A running environment can be saved to a catalog - saves the tools/stacks and configs
- Toolboxes, stacks, configs can be export/import to file
- User starts with a default catalog (shells/editors/etc) and basic home environ
 - Edit personal/group files
- User can add catalogs from her groups to get preconfigured stacks

Alternatives/Options/Ideas

- No Dominant tool (Developer with TestDrive/Workbench or the other way)
- Are there permanent, session, and ad-hoc instances? How to manage?



