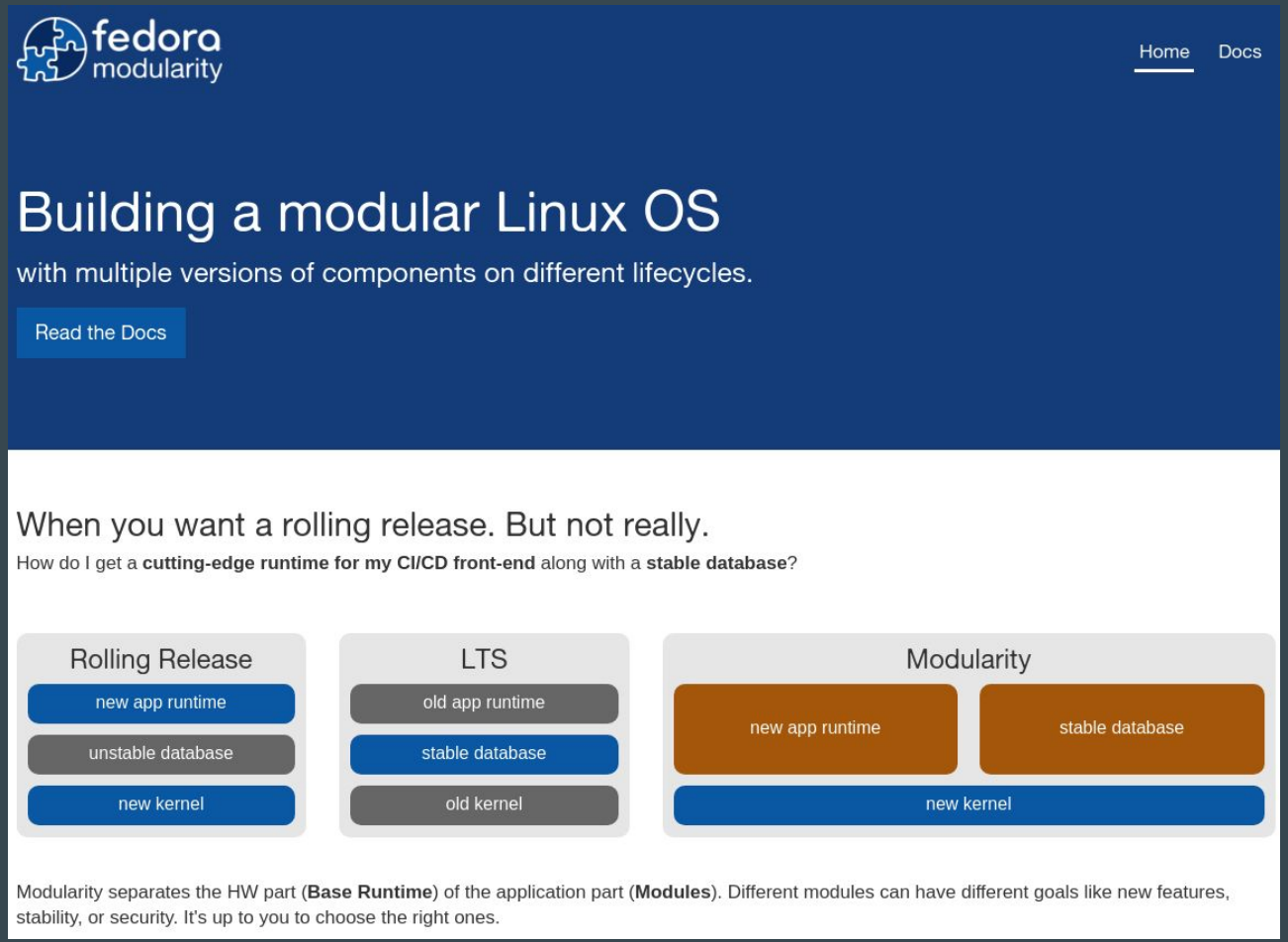
Modularity Update

Fedora Council, Mar. 22, 2017

fedora
modularity

Then

- Accomplished
  - New website & logo
  - Content Plan
  - Architecture
  - Update model



https://docs.pagure.org/modularity

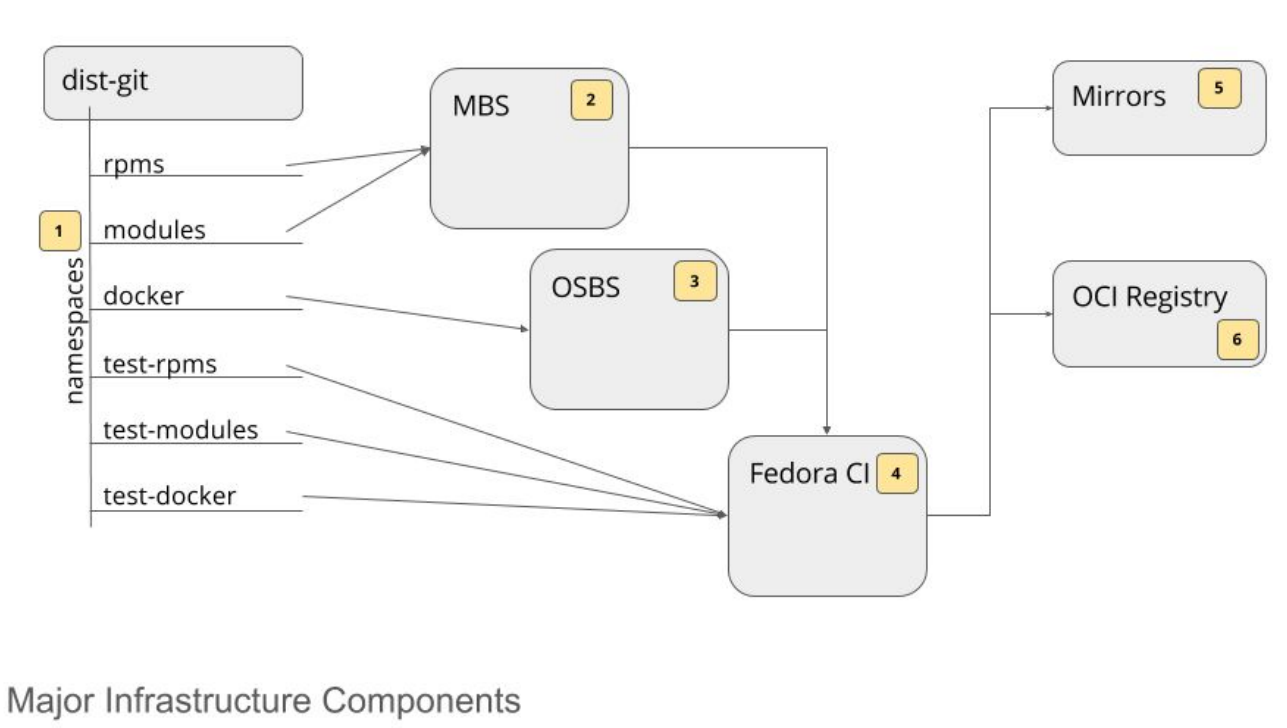
General Work Complete

- Content plan — a list of services to be provided, by use case. Still in progress (probably forever)
- Architecture: exploration via prototype container versions of
  - postfix & dovecot — dealing with root requirements, multi-container services
  - chrony — tightly host integrated service
  - dns server — multiple use cases (authoritative & caching), same container
  - memcached — same container, Stand-Alone and Orchestrated use cases
- Architecture: container deployment models for single host
- Update model, keeping to a stream

Now

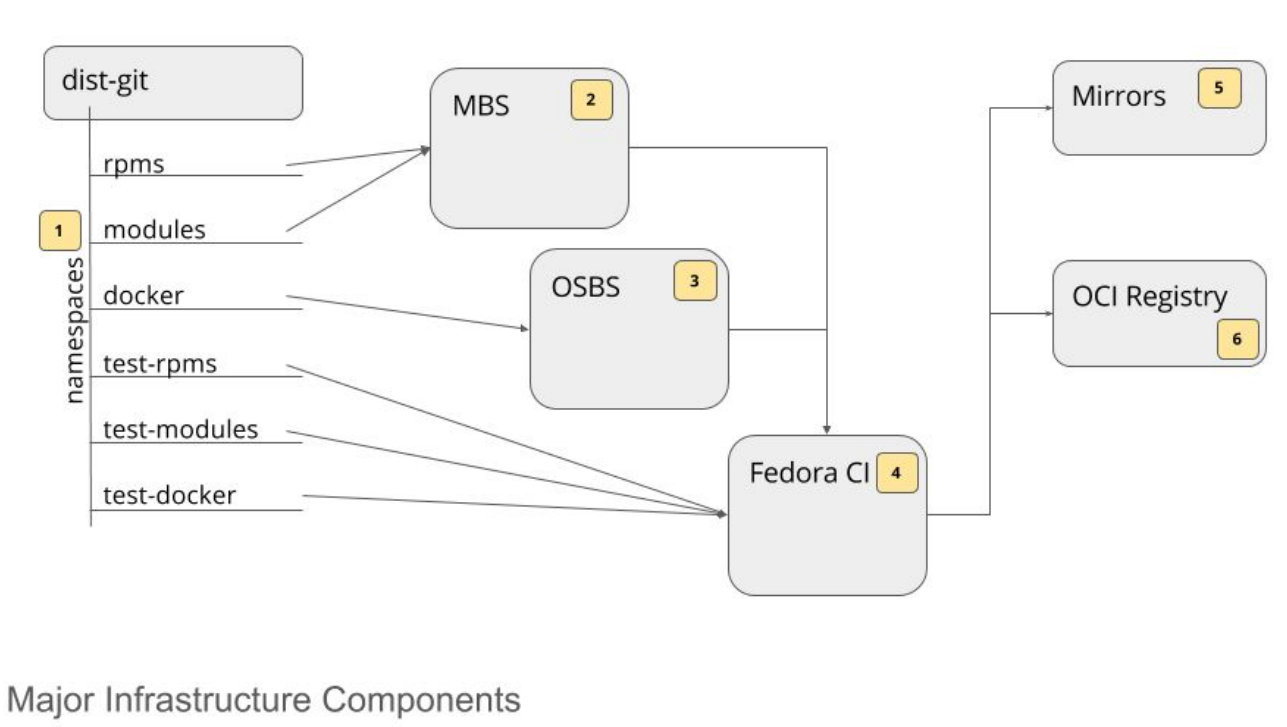
- F26 Boltron
  - Build Infrastructure
  - Base Runtime
  - Modules & Containers
  - Client Tooling
  - General Tests
  - Documentation

Build Infrastructure



Major Infrastructure Components

1. dist-git will be enhanced to include a number of new namespaces to store different kinds of information.
   a. modules: module-md files defining the contents of a module, any other supporting material
   b. docker: dockerfiles defining the export artifact and any supporting materials
   c. test-modules: tests specifically for modules
   d. test-docker: test specifically for containers
2. MBS: The Module Build Service is essentially a wrapper around Koji that knows when and how to rebuild RPMs based on events in the infrastructure. For example, if one RPM is changed it tracks and rebuilds any RPMs that rely on it.
3. OSBS: OpenShift Build Service: The Fedora implementation of OSBS. Has awareness of modules and how to build them in to containers as requested. Also tracks when and if containers need to be rebuilt.



Major Infrastructure Components

4. Fedora CI: Activated by Taskotron, executed using behave or other frameworks on various CI tools. Remains to be seen what the easiest choices will be.
5. RPMs and module-md data will be distributed as a collapsed directory tree with some number of modules identified in the flat tree. We will be experimenting with this model for F26, may not be the long-term best answer. No content is released on this channel unless it has passed the CI infrastructure.
6. OCI Registry: In theory, this is a universal repository but currently just supports Docker containers. No content is released on this channel unless it has passed the CI infrastructure. https://registry.fedoraproject.org/

Build Infrastructure

Build Infrastructure: Not available for F26

- The F26 CI implementation will really just be automated testing with results in ResultsDB. Better UI, direct integration with developer workflow (“real” CI), etc. is planned for F27.
- Any path for updates to the content. We will not be modifying any of the update infrastructure to support rebuilds, CVEs, etc. For example, Bodhi will not be impacted during F26. Planned to be addressed in F27.
- No changes to bug reporting paths. Bugzilla will not be modified to support a user navigating from a Boltron component to the appropriate components in the Fedora Infrastructure. Planned to be addressed in F27.
- The new branching model, i.e. branching components based on upstream release cadence vs “distro release” (e.g. f27 branch) will not be available for F26. We will be able to simulate the use case using F26 and rawhide branches but it will not be a complete test. The branching is planned for F27.

Base Runtime

- Base Runtime Module
  - Docker Base Image & Install Profile
  - QEMU Image & Install Profile
  - RPM Repo
- Automated Tests
- Static Build Root

Modules & Containers

- A number of modules exercising various use cases most delivered as both Stand-Alone Containers* & RPM Repos
- Each module will have a dist-git.modules repo
- Each container will have a dist-git.docker repo
- Automated tests for each module and container will be created. These may be very basic for F26

* terminology may change

F26 Modules & Containers

Committed

- Container Runtime Module
- DNF Full Version
- Shared User Space Module
- MariaDB Server
- MongoDB Server
- Python Runtime
- NodeJS Runtime

Targeted

- SSSD
- System logging
- Debugging Tools
- NTP Server
- FTP Server
- DHCP Client
- Authoritative DNS server
- Caching DNS server
- PostgreSQL Server
- Apache Web Server (prod)
- Apache Web Server (dev)
- nginx Web Server
- TCP/HTTP Load Balancer
- PHP Runtime

Proposed

- NTP Client
- Mail Transfer Agent
- NFS Server
- Samba Domain Controller
- Mail Delivery Agent
- DHCP Server
- Memory Object Cache
- HTTP Reverse Proxy Cache
- Ruby Runtime
- Perl Runtime
- C Runtime

Client Tooling

- Experimental version of DNF & plugins to expose and interact with modules
- Experimental version of DNF & plugins to expose and interact with modules as containers
- Organize a Fedora Test Day to gather feedback on tooling

General Tests

- Modulemd Linter & Tester: An automated test that evaluates the quality of a module. Should automate the testing of the Fedora Module Guidelines as much as possible.
- Dockerfile Linter & Tester: An automated test that evaluates the quality of a container. Should automate the testing of the Fedora Container Guidelines as much as possible.
- An initial implementation of libAbigail testing to ensure a module meets the API it claims. Stretch goal to indicate change in API between versions of a module.

Documentation & Miscellaneous

- A quick start guide for users of Boltron
- A compatibility guide indicating what services will be available from the “platform”
- How to become a module creator
- Module creation guidelines (a la packaging guidelines)
  - Establish processes for review
- SELinux support for modularity proposal and changes

Questions?

fedora
modularity