Layered Image Build System

a.k.a. OSBS

Tomáš Tomeček <ttomececk@redhat.com>
OSBS, what’s that?

- OpenShift Build System
- We are using OpenShift to schedule builds
- osbs (client)
  - talks to OpenShift’s API
- dock (soon to be Atomic Reactor)
  - builds the image
Architecture

fedpkg

osbs-client

OSBS

build container
dock
Architecture

fedpkg

osbs-client

auth

httpd

OSBS

build container

dock
Architecture

koji

builder

osbs-client

auth

httpd

OSBS

build container

dock

fedpkg

auth
Architecture

koji
  builder
  osbs-client
  auth
  fedpkg

httpd
  auth
  tar via http

OSBS
  pulp registry
  docker image
  build container
dock
  tar via nfs

image-export
Layered vs. Base

● layered image
  ○ FROM fedora
  ○ RUN yum install -y ...

● base image
  ○ FROM scratch
  ○ ADD fs.tar.gz
Workflow

1. Ask for dist-git repo
2. Put your Dockerfile to the repo
3. `git commit && git push`
4. `fedpkg container-build`
5. `docker pull`
fedpkg container-build --help

--build-with {koji,osbs}
Build container with specified builder type.
[default: osbs]

--target TARGET
Override the default target

--repo-url [REPO_URL [REPO_URL ...]]
URL of yum repo file
Signed vs. Unsigned Content

1. Koji targets provide unsigned packages
2. Signed packages (composes, distill)
   - Getting signed packages is hard
   - We can ship images with signed content ONLY
Features of Build System

- Downloads base image for you from preconfigured registry
- Puts base image ID to dockerfile
- Fetches dist-git artifacts
- Injects `LABEL` s inside dockerfile
- Final image is squashed
Features of Build System (2)

- Pushes final image to registry
- Stores dockerfile inside image
- Magic with yum repositories
- Inspects final image (signed content)
- Provides thorough build logs
- Imports image to koji
Resources

https://github.com/DBuildService/dock
https://github.com/DBuildService/osbs
https://github.com/openshift/origin
https://pagure.org/rpkg/908028b17f84c3d0c853837f56f62d55ffcc8f99