



Katello

Container Management

Ian Ballou

GitHub: [ianballou](#)

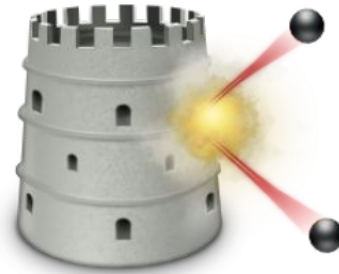
Matrix: [iballou](#)



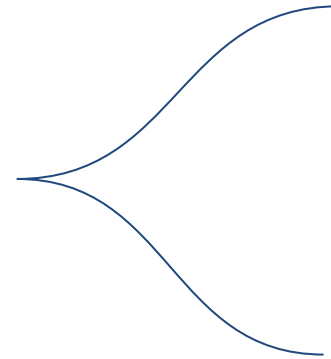
The Stack



Foreman



Katello



Candlepin



Pulp



Katello Container Registry



Registry: Katello vs Pulp functions

Katello

- Token authentication
- Catalog
- Search

Pulp

- Content distribution



Registry: Apache Setup

```
<Location "/pulpcore_registry/v2/">  
  RequestHeader unset REMOTE_USER  
  RequestHeader set REMOTE_USER "%{SSL_CLIENT_S_DN_CN}s"  
  env=SSL_CLIENT_S_DN_CN  
  ProxyPass unix:///run/pulpcore-api.sock|http://pulpcore-api/v2/  
  ProxyPassReverse unix:///run/pulpcore-api.sock|http://pulpcore-api/v2/  
</Location>
```

Pulp-Container's /v2 endpoint clashes with Katello's /v2 smart proxy endpoint



Katello Container Naming

- Container repos have prefixes/namespaces
 - Determined by CV & LCE
- Namespaces are slash-delimited by default
- Necessary to differentiate between images



```
$ podman search katello.example.com  
NAME  
katello.example.com/my_org/library/my_cv/my_prod/firefox
```



Katello Container Naming

❗ The *Registry Name Pattern* overrides the default name by which container images may be pulled from the server. (By default this name is a combination of Organization, Lifecycle Environment, Content View, Product, and Repository labels delimited by forward slashes "/".)

The name may be constructed using ERB syntax. Variables available for use are:

```
organization.name  
organization.label  
repository.name  
repository.label  
repository.docker_upstream_name  
content_view.label  
content_view.name  
content_view_version.version  
product.name  
product.label  
lifecycle_environment.name  
lifecycle_environment.label
```

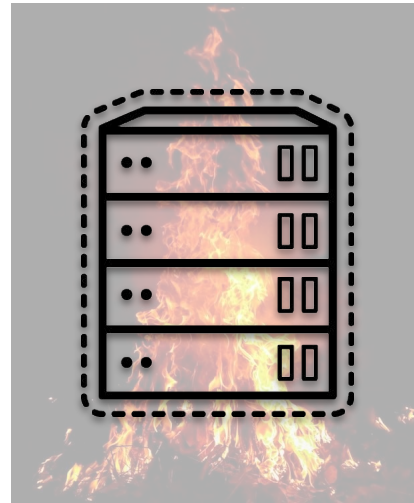
Examples:

```
<%= organization.label %>-<%= lifecycle_environment.label %>-<%= content_view.label %>  
<%= organization.label %>/<%= repository.docker_upstream_name %>
```



Problem: Container Tagging

Your
Container



Latest tag
moved to who
knows what

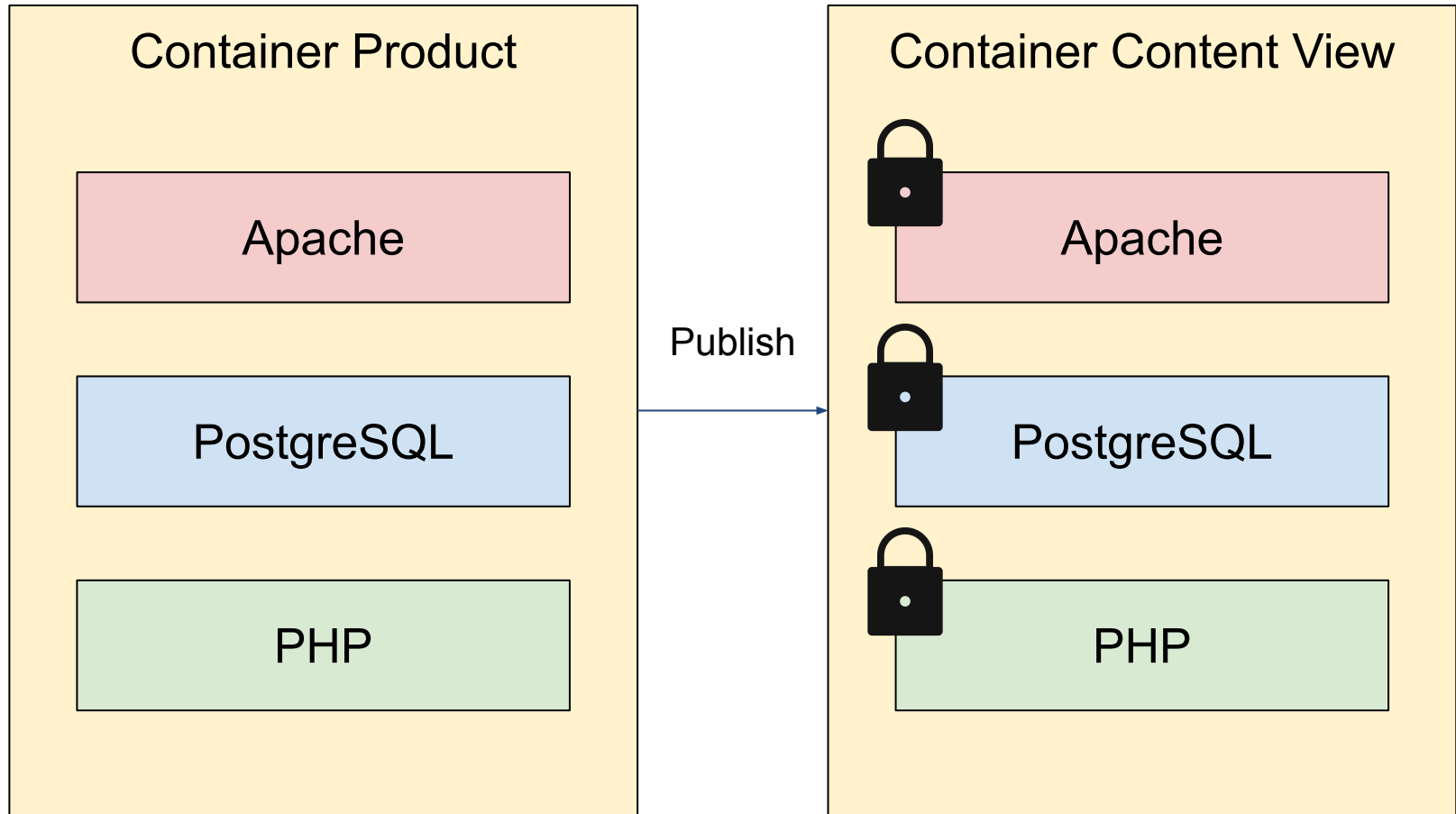


Solution: Content Views

- Snapshot container repositories
- Serve snapshot and newest synced containers at once
- Separate different snapshot versions by lifecycle environment
 - Can power a pipeline



Solution: Content Views



Solution: Content Views

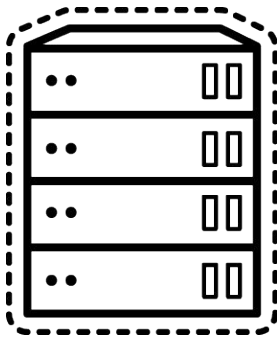
```
$ podman search apache  
NAME  
example.com/my_org/my_prod/apache  
example.com/my_org/mv_cv/apache
```

* results will depend on the lifecycle environment
container naming scheme



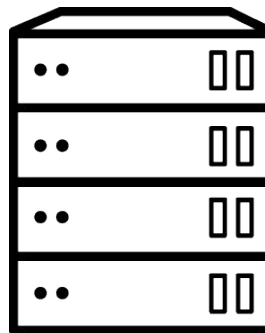
Solution: Content Views

Development



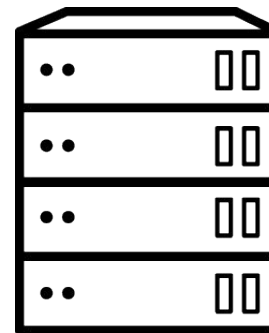
```
host/org/cv/  
dev/apache
```

Testing



```
host/org/cv/  
test/apache
```

Production



```
host/org/cv/  
prod/apache
```



Alternative?

```
$ podman pull apache@sha256:7bba87...
```

Issue: not reproducible

- Container content discoverable by tag only
- Another environment may never see the same digests if upstream tags move



Introducing Container Push

```
$ podman push 66640e9653de \  
example.com/org/prod/arianna3:latest
```

```
Getting image source signatures
```

```
Copying blob 29e363cff31e skipped: already exists
```

```
Copying config 66640e9653 done |
```

```
Writing manifest to image destination
```

Why?

Distribute your own containers without a second registry!



How-to

- Namespace pushes by organization and product

```
$ podman push 66640e9653de \  
  `hostname`/default_organization/product/busybox:latest
```



How-to

- IDs may be used if names are ambiguous
 - Org & product labels are case sensitive in Katello, but container repository names are downcased

```
$ podman push 66640e9653de \  
  `hostname`/id/1/130/busybox:latest
```



Technical Lesson: Container Push & Content Views

- What are some core features of content views?
 - Snapshotting repositories
 - Distributing content at customizable paths



Technical Lesson: Container Push & Content Views

- How do you snapshot repositories?
 - Save the current repository version
- How do you distribute content at a custom path?
 - Create a new Pulp Distribution

Issue:

Note

Content is pushed to a push repository type. A push repository does not support mirroring of the remote content via the Pulp API.

Note

Rollback to the previous repository versions is not possible with a push repository. Its latest version will always be served.



Technical Lesson: Container Push & Content Views

- To work with the limitations: copy Pulp push repo
- CopyAllUnits action either:
 - Copies current latest_version_href, or
 - Copies content to a new repo if filtering

The solution then is to force copying for all container push repositories



OCI Flatpak Support

- Katello can act an OCI flatpak remote for registered clients.
- Katello can sync flatpak repositories and distribute them.
- Clients can point to katello server as a flatpak remote and install repositories hosted on it:

```
flatpak --user remote-add  
--authenticator-name=org.flatpak.Authenticator  
.0ci katello oci+https://katello.example.com/  
flatpak remote-ls  
flatpak install firefox
```



```
$ hammer flatpak-remote create --name=$name --url=$url
$ hammer flatpak-remote list
$ hammer flatpak-remote info
$ hammer flatpak-remote scan --id=$ID
$ hammer flatpak-remote repository list \
    --flatpak-remote-id=$ID
$ hammer flatpak-remote remote-repository mirror \
    --id=$remote_repo_id --product-id=$product
$ hammer repository sync --id=$repo_id
```



OCI Flatpak Support Status

- Hammer-only for Katello 4.16
- Taxonomy / content view / lifecycle environment scoping calculated by IP address for registered hosts
 - Unknown hosts will see global latest Flatpaks on system
- No taxonomy / content view / lifecycle environment scoping on smart proxies yet



Bootable Containers

“The [bootc documentation](#) summarizes bootable containers as "transactional, in-place operating system updates using OCI/Docker container images". In other words, updates to the operating system (OS) are shipped by using container images. That implies that the Linux kernel, the bootloader, drivers, etc. are all part of the container image which renders the container image "bootable".”

- <https://docs.fedoraproject.org/en-US/bootc/getting-started/>



Bootable Containers

“[Bootc](#) container images differ technically from application containers in two important ways:

- bootc images use [OSTree](#) inside the container
- A bootc image has a kernel and enough other packages to boot a physical or virtual machine.”

- <https://www.redhat.com/en/blog/image-mode-red-hat-enterprise-linux-quick-start-guide>



Bootable Containers



FOREMAN



Booted container images

Search



1 - 2 of 2



Image name ↑

Image digests

Hosts



quay.io/centos-bootc/centos-bootc:stream10

3

4

Image digest

Hosts

sha256:54256a998f0c62e16f3927c82b570f90bd8449a52e03daabd5fd16d6419fd572 1

sha256:54256a998f0c62e16f3927c82b570f90bd8449a52e03daabd5fd16d6419fd573 2

sha256:54256a998f0c62e16f3927c82b570f90bd8449a52e03daabd5fd16d6419fd574 1



quay.io/centos-bootc/centos-bootc:stream9

1

1



Bootable Containers

Hosts

Schedule a job ▾

Register

Create



▾ ▾



1 - 20 of 146 ▾ < >

<input type="checkbox"/>		Name ↑	Installable updates	Last se... ↑	Content view enviro...
<input type="checkbox"/>		bootc-machine.manicotto.example.com	0 0 0 0	1 day ago	Library/Empty_View
<input type="checkbox"/>		centos9-proxy-devel.manicotto.example.com	–	–	–
<input type="checkbox"/>		centos9-stream.00b904ec.manicotto.example.com	0 0 0 0	13 days ago	Library
<input type="checkbox"/>		centos9-stream.0432006e.manicotto.example.com	0 0 0 0	13 days ago	Library
<input type="checkbox"/>		centos9-stream.0b1a4a08.manicotto.example.com	0 0 0 0	13 days ago	Library
<input type="checkbox"/>		centos9-stream.0ce9dlc7.manicotto.example.com	0 0 0 0	13 days ago	Library
<input type="checkbox"/>		centos9-stream.0d307dcc.manicotto.example.com	0 0 0 0	13 days ago	Library
<input type="checkbox"/>		centos9-stream.0ee63b79.manicotto.example.com	0 0 0 0	13 days ago	Library
<input type="checkbox"/>		centos9-stream.11c8dc30.manicotto.example.com	0 0 0 0	13 days ago	Library
<input type="checkbox"/>		centos9-stream.146aaab0.manicotto.example.com	0 0 0 0	13 days ago	Library



Bootable Containers

Image mode details



[Modify via remote execution](#)

Running image	quay.io/centos-bootc/centos-bootc:stream10
Running image digest	sha256:54256a998f0c62e16f3927c82b570f90bd8449a52e03daabd5fd16d6419fd572
Staged image	—
Staged image digest	—
Available image	—
Available image digest	—
Rollback image	quay.io/centos-bootc/centos-bootc:stream10
Rollback image digest	sha256:9ed49e9b189f5dae5a01ea9abdcef0884616300b565d32061aea619f2e916be3



Bootable Containers

Jobs > Run job

Run job

[Use legacy form](#)

- 1 Category and template
 - 2 Target hosts and inputs
 - 3 Advanced fields
 - 4 Schedule
- Type of execution
- 5 Review details

Target hosts and inputs

Search query name ^ (bootc-machine.manicotto.example...

Search query name ^ (bootc-machi...

[Clear all filters](#)

Apply to **1** hosts

action *

options

target

Next

Back

Run on selected hosts

Skip to review

Cancel



Demo time!





FOREMAN

Thanks!

<https://theforeman.org/>

<https://github.com/Katello/katello>

https://github.com/Katello/smart_proxy_container_gateway