

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Katello 3.1 Installation

These instructions are for installing Katello 3.1, but the latest stable is [3.18](#).

Note: After installation of Katello, be sure to trust Katello's CA certificate on your system. This is required for the encrypted NoVNC connections. You will find `katello-default-ca.crt` in the `/pub` directory of your Katello server (e.g.

<http://katello.example.com/pub/katello-default-ca.crt>).

Hardware Requirements

Katello may be installed onto a baremetal host or on a virtual guest. The minimum requirements are:

- Two Logical CPUs
- 4 GB of memory (8 GB highly recommended)
- The filesystem holding /var/lib/pulp needs to be large, but may vary depending on how many different Operating Systems you wish to synchronize:
 - Allocate 30 GB of space for each operating system. Even though an operating system may not take up this much space now, this allows space for future updates that will be synchronized later.
- The filesystem holding /var/lib/mongodb needs at least 4 GB to install, but will vary depending on how many different Operating Systems you wish to synchronize:
 - Allocate around 40% of the capacity that has been given to the /var/lib/pulp filesystem
- The root filesystem needs at least 20 GB of Disk Space

Required Ports

The following ports need to be open to external connections:

- 80 TCP - HTTP, used for provisioning purposes
- 443 TCP - HTTPS, used for web access and api communication
- 5647 TCP - qdrouterd - used for client and capsule actions
- 9090 TCP - HTTPS - used for communication with the smart proxy

Production

Katello provides a puppet based installer for deploying production installations. Production installations are supported on the following OSes:

OS	
CentOS 6	X
CentOS 7	X
RHEL 6	X
RHEL 7	X

Katello can only run on an x86_64 operating systems.

Installation may be done manually or via our recommended approach of using [forklift](#).

Required Repositories

Select your Operating System: Red Hat Enterprise Linux 6

```
yum -y --disablerepo="*" --enablerepo=rhel-6-server-rpms install yum-utils wget  
yum-config-manager --disable "*"  
yum-config-manager --enable rhel-6-server-rpms epel  
yum-config-manager --enable rhel-6-server-optional-rpms
```

```
yum -y localinstall https://fedorapeople.org/groups/katello/releases/yum/3.1/katello/el6/x86_64/katello-repos-latest.rpm  
yum -y localinstall https://yum.theforeman.org/releases/1.12/el6/x86_64/foreman-release.rpm  
yum -y localinstall https://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm  
yum -y localinstall https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm  
yum -y install foreman-release-scl
```

Installation

After setting up the appropriate repositories, install Katello:

```
yum -y install katello
```

At this point the `foreman-installer` should be available to setup the server. The installation may be customized, to see a list of options:

```
foreman-installer --scenario katello --help
```

Note

Prior to running the installer, the machine should be set up with a time service such as ntpd or chrony, since several Katello features will not function well if there is minor clock skew.

These may be set as command line options or in the answer file (/etc/foreman-installer/scenarios.d/katello-answers.yaml). Now run the options:

```
foreman-installer --scenario katello <options>
```

Forklift

Foreman provides a git repository designed to streamline setup by setting up all the proper repositories. Forklift provides the ability to deploy a virtual machine instance via Vagrant or direct deployment on an already provisioned machine. For details on how to install using forklift, please see the [README](#).

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Capsule Installation

Hardware Requirements

The Capsule server is only supported on x86_64 Operating Systems

- 2 Two Logical CPUs
- 4 GB of memory
- Disk space usage is similar to that of the main Katello server [Installation](#)

Required Ports

At a minimum, the following ports need to be open to external connections for installation:

- 80 TCP - HTTP, used for provisioning purposes
- 443 TCP - HTTPS, used for web access and api communication
- 9090 TCP - HTTPS - used for communication with the smart proxy

See the [User Guide](#) for additional information about Capsule services and required ports.

Installation

Install needed packages:

The same yum repositories need to be configured on the Capsule server as the main Katello server. See the installation guide for the [list of required repositories](#).

Once you get the repositories configured, install the foreman-installer-katello package on the capsule

```
yum install -y foreman-installer-katello
```

Generate Certificates for the Capsules

Prior to installing the Capsule, we need to generate certificates on the main [Katello](#) server:

```
capsule-certs-generate --capsule-fqdn "mycapsule.example.com"\n  --certs-tar  "~/mycapsule.example.com-certs.tar"
```

In the above example, replace 'mycapsule.example.com' with your Capsule's fully qualified domain name. This will generate a tar file containing all the needed certificates. You will need to transfer those certificates to the server that you will install your Capsule on using whatever method you prefer (e.g. SCP).

The capsule-certs-generate command will output an example installation command. For example:

```
Installing      Done      [100%] [.....]
Success!
```

To finish the installation, follow these steps:

1. Ensure that the foreman-installer-katello package is installed on the system.
2. Copy ~/mycapsule.example.com-certs.tar to the system mycapsule.example.com
3. Run the following commands on the capsule (possibly with the customized parameters, see foreman-installer --scenario capsule --help and documentation [for](#) more info on setting up additional services):

```
yum -y localinstall http://katello.example.com/pub/katello-ca-consumer-latest.noarch.rpm
subscription-manager register --org "Default_Organization"
foreman-installer --scenario capsule \
  --parent-fqdn      "katello.example.com"\
  --register-in-foreman  "true"\
  --foreman-base-url  "https://katello.example.com"\
  --trusted-hosts    "katello.example.com"\
  --trusted-hosts    "mycapsule.example.com"\
  --oauth-consumer-key "UVrAZfMaCfBiiWejoUVLYCZHT2xhzuFV"\
  --oauth-consumer-secret "ZhH8p7M577ttNU3WmUGWASag3JeXKgUX"\
  --pulp-oauth-secret "TPk42MYZ42nAE3rZvyLBh7Lxob3nEUi8"\
  --certs-tar        "~/mycapsule.example.com-certs.tar"

The full log is at /var/log/capsule-certs-generate.log
```

Install Capsule

Use the provide installation command from `capsule-certs-generate`, and tailor for your own purposes as needed. The defaults will give you a Capsule ready for Content-related services.

See the [User Guide](#) to learn about setting up provisioning related services, as well as the [Foreman manual](#)

[Foreman 2.3.3 has been released! Follow the quick start to install it.](#)

[Foreman 2.2.2 has been released! Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

1. Installation

1.1 Katello

1.2 Capsule

1.3 Clients

2. Upgrade

2.1 Katello

2.2 Capsule

2.3 Clients

3. Release Notes

4. CLI

5. User Guide

5.1 Activation Keys

5.2 Backup

5.3 Capsules

5.4 Content

5.5 Content Hosts

5.6 Content Views

5.7 Disconnected

5.8 Docker Management

5.9 Email Notifications

5.10 Errata

5.11 Glossary

5.12 GPG Keys

5.13 Host Collections

5.14 Lifecycle Environments

5.15 Provisioning

5.16 Puppet Integration

5.17 Red Hat Content

6. Troubleshooting

Client Installation

Client machines can be added in one of two ways: manually or via a provisioned host.

Manual

Install the appropriate Katello client release packages. For CentOS 6, you will also need to enable the COPR repository for subscription-manager.

Select your Operating System: Enterprise Linux 5 (CentOS, etc.) ▾

```
yum install -y https://fedorapeople.org/groups/katello/releases/yum/3.1/client/el5/x86_64/katello-client-repos-latest.rpm  
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-5.noarch.rpm
```

Now you are ready to install the katello-agent:

```
yum install katello-agent
```

Provisioned

In order to install the katello-agent package on a host you are provisioning, you will need to make the appropriate client repository available within your Katello. The first step is to either create a new product or add to an existing product, the appropriate client repository from the dropdown in the [manual](#) section above. After you create the new repositories, they will need to be synced locally. Next, you will then need to add them to the relevant content view(s) for the hosts you are wanting to provision. At this point, a new version of the content view can be published and promoted to the appropriate environments that you are wanting to provision a host into. At this point, you can go provision a host and the host will install the katello-agent package during setup.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Katello Upgrade

Katello supports upgrades from version 2.0. For users transitioning from 1.4, please see - [Transition Guide](#).

Pre-upgrade considerations

Before you upgrade, you need to run the upgrade check script that will check for any active tasks, your version of Katello, and if there are any content hosts that will be deleted (see below). This script is included in Katello 2.4.3.

To run the script:

```
foreman-rake katello:upgrade_check
```

In addition to the information on the screen, the command will print the number of Content Hosts that will be deleted upon upgrade and will generate a csv with the full list including the uuid, name, last checkin date, and a reason for their deletion.

Content Hosts

Katello 3.0 unifies Host and Content Host objects to provide a more unified experience. As a result, it is no longer allowed to have more than one Content Host with the same FQDN. This matches the requirement that is already imposed on the Foreman Host object. Upon upgrade to Katello 3.0 any Hosts that violate one of three rules will be deleted:

- More than one Content Host cannot exist with the same FQDN (all but the most recently registered will be deleted upon upgrade)
- Content Hosts must have a reported FQDN
- If a Host and a Content Host exist with the same FQDN, they must be within the same Organization.

This report is automatically run as part of the `katello:upgrade_check` script. However, if you want to get just the content host info, you can run:

```
foreman-rake katello:preupgrade_content_host_check
```

Step 1 - Backup

If Katello is running on a Virtual Machine, we recommend to take a snapshot prior to upgrading. Otherwise, take a backup of the relevant databases by following the [instructions here](#).

Step 2 - Operating System

Ensure your operating system is fully up-to-date:

```
yum -y update
```

NOTE: If kernel packages are updated here (e.g. upgrading el 6.6 to 6.7), you must reboot and ensure the new kernel and SELinux policy is loaded before upgrading Katello.

Step 3 - Repositories

Update the Foreman and Katello release packages:

- RHEL6 / CentOS 6:

```
yum update -y https://fedorapeople.org/groups/katello/releases/yum/3.1/katello/el6/x86_64/katello-repos-latest.rpm  
yum update -y https://yum.theforeman.org/releases/1.12/el6/x86_64/foreman-release.rpm
```

- RHEL7 / CentOS 7:

```
yum update -y https://fedorapeople.org/groups/katello/releases/yum/3.1/katello/el7/x86_64/katello-repos-latest.rpm  
yum update -y https://yum.theforeman.org/releases/1.12/el7/x86_64/foreman-release.rpm
```

Step 4 - Update Packages

Clean the yum cache

```
yum clean all
```



Update the required packages:

```
yum -y update
```

Step 5 - Run Installer

The installer with the `--upgrade` flag will run the right database migrations for all component services, as well as adjusting the configuration to reflect what's new in Katello 3.1

```
foreman-installer --scenario katello --upgrade
```

Congratulations! You have now successfully upgraded your Katello to 3.1 For a rundown of what was added, please see [release notes](#)!

If for any reason, the above steps failed, please review `/var/log/foreman-installer/katello.log` – if any of the “Upgrade step” tasks failed, you may try to run them manually below to aid in troubleshooting.

Foreman 2.3.3 has been released! Follow the quick start to install it.

Foreman 2.2.2 has been released! Follow the quick start to install it.

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

1. Installation

1.1 Katello

1.2 Capsule

1.3 Clients

2. Upgrade

2.1 Katello

2.2 Capsule

2.3 Clients

3. Release Notes

4. CLI

5. User Guide

5.1 Activation Keys

5.2 Backup

5.3 Capsules

5.4 Content

5.5 Content Hosts

5.6 Content Views

5.7 Disconnected

5.8 Docker Management

5.9 Email Notifications

5.10 Errata

5.11 Glossary

5.12 GPG Keys

5.13 Host Collections

5.14 Lifecycle Environments

5.15 Provisioning

5.16 Puppet Integration

5.17 Red Hat Content

6. Troubleshooting

Capsule Upgrade

Step 1 - Operating System

Ensure your operating system is fully up-to-date:

```
yum -y update
```

NOTE: If kernel packages are updated here (e.g. upgrading el 6.6 to 6.7), you must reboot and ensure the new kernel and SELinux policy is loaded before upgrading Katello.

Step 2 - Repositories

Update the Foreman and Katello release packages:

- RHEL6 / CentOS 6:

```
yum update -y https://fedorapeople.org/groups/katello/releases/yum/3.1/katello/el6/x86_64/katello-repos-latest.rpm  
yum update -y https://yum.theforeman.org/releases/1.12/el6/x86_64/foreman-release.rpm
```

- RHEL7 / CentOS 7:

```
yum update -y https://fedorapeople.org/groups/katello/releases/yum/3.1/katello/el7/x86_64/katello-repos-latest.rpm  
yum update -y https://yum.theforeman.org/releases/1.12/el7/x86_64/foreman-release.rpm
```

Step 3 - Update Packages

Clean the yum cache

```
yum clean all
```

Update packages:

```
yum update -y
```

Step 4 - Regenerate Certificates

On the Katello server, regenerate the certificates tarball for your capsule:

```
capsule-certs-generate --capsule-fqdn "mycapsule.example.com"\n--certs-tar  "~/mycapsule.example.com-certs.tar"
```

And copy them to your capsule:

```
scp ~/mycapsule.example.com-certs.tar mycapsule.example.com:
```

Step 5 - Run Installer

The installer with the `-upgrade` flag will run the right database migrations for all component services, as well as adjusting the configuration to reflect what's new in Katello 3.1

```
foreman-installer --scenario capsule --upgrade\n  --certs-tar ~/mycapsule.example.com-certs.tar\n  \\\n    --certs-update-all --regenerate --deploy
```

Congratulations! You have now successfully upgraded your Capsule to 3.1 For a rundown of what was added, please see [release notes!](#)

If for any reason, the above steps failed, please review `/var/log/foreman-installer/capsule.log` – if any of the “Upgrade step” tasks failed, you may try to run them manually below to aid in troubleshooting.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! Follow the quick start to install it.

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

1. Installation

1.1 Katello

1.2 Capsule

1.3 Clients

2. Upgrade

2.1 Katello

2.2 Capsule

2.3 Clients

3. Release Notes

4. CLI

5. User Guide

5.1 Activation Keys

5.2 Backup

5.3 Capsules

5.4 Content

5.5 Content Hosts

5.6 Content Views

5.7 Disconnected

5.8 Docker Management

5.9 Email Notifications

5.10 Errata

5.11 Glossary

5.12 GPG Keys

5.13 Host Collections

5.14 Lifecycle Environments

5.15 Provisioning

5.16 Puppet Integration

5.17 Red Hat Content

6. Troubleshooting

Client Upgrade

When upgrading clients there are 2 scenarios: manually added clients and provisioned clients.

Step 1 - Update Repositories

Manually Added Clients

Update the Katello client release packages:

Select your Operating System:

```
yum update -y https://fedorapeople.org/groups/katello/releases/yum/3.1/client/RHEL/5Server/x86_64/katello-client-repos-latest.rpm  
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-5.noarch.rpm
```

Provisioned Clients

If the katello-agent was setup during provisioning from a locally synced repository then you will need to go through some [initial setup](#) to add the 3.1 client repositories to your Katello for each version needed. After you create the new repositories, they will then need to be added to the relevant content view(s) and the older versions removed. At this point, a new version of the content view can be published and promoted to the appropriate environments. Once the new package is available the clients can be updated following the next steps.

Step 2: Update Packages

Clean the yum cache

```
yum clean all
```



Update packages:

```
yum update katello-agent
```

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Katello 3.1 (California Common) Release Notes

Features

Tooling

- Have the katello service script handle pulp_streamer and squid ([#15484, 7ce677fe](#))

Installer

- Make migration generation easier ([#15205, 2af541b4](#))
- Provide a Rake command to setup a local set of modules similar to how the RPM is built ([#14334, ada4a4f2](#))

CLI

- Upgrade to rubocop 0.39 ([#15029, 36439787](#))

Tests

- Run rubocop on rake files ([#15004, 7e1ce1b8](#))

foreman-debug

- Do not collect squid logs when called from sosreport ([#14608, 32f4434b](#))

API

- migrate Release version to host subscriptions controller ([#13903, 536c81e3](#))

Repositories

- Running Sync status to contain a direct link to sync tasks ([#10315, c7746d63](#))

Other

- Add the bin directory and rails script to katello ([#15035, 2d5b1b8e](#))
- Support rubocop 0.39.0 ([#14950, bd547d2d, 97f83ee6, 5e9b7a60, f940aab0, 36c185c5](#))
- Update Katello to use Task Groups ([#13566, f803b183](#))

Bug Fixes

Web UI

- Sync Status page not drawn properly if sync in progress ([#15499, 43ddadae](#))
- Duplicate key on "katello_content_facet_errata_eid_caid" under load ([#15407, fa3f1fa3](#))
- Available pool quantity is wrong after removed subscribed pool(by virt-who) ([#15394, e2875646](#))
- "Reset Puppet Environment to match selected Content View" does not work ([#15295, 525a381b](#))
- The page does not load while accessing Content -> "Gpg Keys" when having Any Context set ([#15221, 16d3a1cb](#))
- Unnecessary date/time transformation for sync plan start date ([#15180, 0baa312f](#))
- Subscription table has mismatched row colors when there are an odd number of subscriptions under one name ([#11871, 5a4af062](#))

CLI

- content view filter cannot be created ([#15395, 3b381817](#))
- 'hammer sync-plan info' doesn't return start date ([#15204, 59b43952](#))
- Unable to remove host collection from activation key via hammer ([#15162, bb736223](#))
- Some tests in hammer-cli-katello use MiniTest's assert() method improperly ([#15142, 27fe17d0](#))
- Repository list test references nonexistent option ([#15032, aa6f1757](#))
- (bats fail) hammer repository upload_content missing --organization param ([#15013, e8c714b8](#))
- hammer-cli-katello: 'capsule content synchronize' generates error with 'name' input ([#14871, a167e20e](#))
- hammer host errata apply async - always displays success ([#14860, 8fbe49ee](#))
- hammer capsule content synchronization needs --help to indicate which fields are required and/or optional ([#14656](#))
- hammer activation-key --unlimited-content-hosts flag is confusing ([#14333, fb494a72](#))

Installer

- katello install failing due to passenger error ([#15393, 56adbf64, 55154c9a](#))
- The try method is used in katello-installer, but it only exists in Rails, not Ruby ([#15327, 41df3788](#))
- There is no nil check for missing kafo params 'capsule_certs' and 'parent_fqdn' ([#15325, 0aa56393](#))
- Capsule upgrade 6.1->6.2 snap 13 incorrectly resets trusted hosts and foreman-url ([#15315, b2322b35](#))
- Make proxy.rb executable ([#15309, cee964f1](#))

- Remove unused sam-installer (#15208, 4cfdddef7)
- Set Client SSL certs via parameters (#15063, fe37561e)
- capsule-certs-generate does not honour “–parent-fqdn” parameter (#14978, 3df90771)
- Have Katello-installer check to see if there is a proxy env set before running (#14956, af52c83c)
- deploy candlepin url using fqdn (#14617, f18f34db)
- [RFE] katello-installer should include –capsule-tftp true by default (#13546)
- Katello 3.0 RC5 Upgrade fail (#15269, b5f07391)

Errata Management

- errata page broken (#15389, 42502063)
- distinct used when finding applicable errata, causing severe slowdowns on dashboard (#15253, 4667141b)

Pulp

- checksum-type does not update on already synced repository at Capsule. (#15377, 57ab21e3)
- Removing a content view, doesn't free disk space (#15369)

Activation Key

- Activation Key returning all systems under associations tab (#15347, 515a48bf)

Hosts

- Subscription status is wrong on content host details > subscriptions tab (#15334, 3c040d05)
- No permissions for power, puppetrun e.t.c in API (#15236, 5d553340)
- Duplicate entries for hypervisors after renaming content host (#15197, 78bb525a)
- Content hosts licensing status is hard to determine from dot colors (#14992, bf97d409)
- Enable feature to update all packages for a host collection to patch all hosts at once (#14940, f85fc0dc)
- Katello should be using host.comment instead of content_host.description (#14822, 155c2f55)
- Content Host status is green, but shows “Warning” (#14372, d31dff1e)

Subscriptions

- Hammer subscription list no longer reports subscriptions as uuids (#15275, edafb346)
- webUI, Subscription Status shows green instead of red if registered system without auto-attach. (#15209, c621ef28)
- removing a vdc subscription causes candlepin listener to fail as the guest subs are removed too (#15207, a89f1f0c)
- content host installed_products array needs to pass product arch and version to candlepin when specified (#14593, 20efbb2e)
- dynflow_executor memory usage continues to grow, causing performance degradation (#12650, 7f661597)
- manifest import messages no longer displayed in UI (#11125, 9789f11e)
- Importing rhsm facts doesn't set the virtual flag (#15559, 7b2e6900)

API

- API /products/:product/repositories fails with undefined method (#15186, aad44630)
- Organization show api broken (#15185, 77e1ab61)
- Adding puppet module to Content View by API does not set version, and therefore exclude it in publishing the module (#15177, 849f26b6)
- Api request fails for GET /katello/api/organizations/:organization_id/environments/:id/repositories (#15159, 195b9a69)
- PUT /api/v2/host_collections/ do not return ‘host_ids’ attribute (#15486, 15c1a2eb)
- Unable to add puppet module stdlib to my content view through the UI (#15435, 11a5a536)

API doc

- Document facet attributes for hosts (#15165, 29f42a35)

Capsule

- capsule-remove doesn't remove foreman-installer configs and old certs which causes failure of next capsule install (#14906, 2f31c5f9)
- Main sync page throws error/warning upon completion of sync if capsules non-responsive (#14873, 750181d0)
- Capsule sync progress should be shown only up to 2 decimal places (#14644, 91e6967e)

Organizations and Locations

- “Content Host Counts” column on the Errata page should only count hosts in the current org (#14933, 1bddf751)

foreman-debug

- [foreman-debug] collect /etc/qpid-dispatch/qdrouterd.conf ([#14881, b73086c2](#))

Documentation

- katello-packaging README points to wrong branch ([#14515, 1ada7271](#))

Performance

- product API call with 'enabled=true' slow, causing product page to load slowly ([#14259, 98e9acf6](#))

Upgrades

- On upgraded Viewer role user still can manage Content Views ([#15460, 118ea637](#))

Candlepin

- sub-man registration currently blocks on applicability generation ([#12749, 8305bfbf](#))

Other

- list of packages on About page missing crucial entries ([#15242, 1ade0806](#))
- REST Timeout during manifest import ([#15170, 022e3d86](#))
- Remove old broken script files ([#15034, ebf96ef2](#))
- Errant single quote in apidoc for repository create param checksum ([#14975, 0879c327](#))
- Nightly fails with undefined method 'value' for module ([#14973, e19e50ab](#))
- Docs links in the contributing markdown file are broken ([#14972, 34bfd757](#))
- Uploaded puppet modules from hammer CLI not visible in the UI ([#14949, 4730c9d2](#))
- CDN url is allowed to be "https", which only works for one hostname ([#14916, 850a07fe](#))
- foreman proxy still running on port 8000 needlessly. ([#14750, 01635aa2](#))
- update of subscription facet triggers a second call to dynflow ([#14709, 77b7b555](#))
- extracting strings is not working due to humanized names ([#14610, 42c1152d](#))
- hammer ping prompts for admin password ([#14397](#))
- Katello's late initialization causes issues when resuming dynflow tasks ([#15114, 28cb3ef5](#))

Contributors

Adam Price Adam Ruzicka Andrew Kofink Brad Buckingham Bryan Kearney Chris Duryee Chris Roberts Christine Fouant David Davis Eric D Helms Ivan Nečas John Mitsch Justin Sherrill Marek Hulan Mike McCune Ondrej Prazak Partha Aji Ranjan Kumar Sean O'Keeffe Shimon Shtein Shlomi Zadok Stephen Benjamin Tomer Brisker Tom McKay Walden Raines xprazak2 Zach Huntington-Meath

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Hammer

What is the CLI?

The Hammer CLI provides users with a command-line interface for interacting with Katello. It's our goal to make all functionality that's accessible through Katello's Web UI also available through Hammer so that users may use Hammer for their

entire Katello workflow.

Installation

The first step to install the CLI is to setup the appropriate repositories: foreman, katello and epel.

Select your Operating System: 

```
yum -y --disablerepo="*" --enablerepo=rhel-6-server-rpms install yum-utils wget  
yum-config-manager --disable "*"  
yum-config-manager --enable rhel-6-server-rpms epel  
yum-config-manager --enable rhel-6-server-optional-rpms
```

```
yum -y localinstall https://fedorapeople.org/groups/katello/releases/yum/3.1/katello/el6/x86_64/katello-repos-latest.rpm  
yum -y localinstall https://yum.theforeman.org/releases/1.12/el6/x86_64/foreman-release.rpm  
yum -y localinstall https://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm
```

After setting up the appropriate repositories, install Katello:

```
yum -y install rubygem-hammer_cli_katello
```

How do I use Hammer?

To get started with hammer, view the help:

```
hammer -u <user> -p <password> --help
```

How do I contribute to Hammer?

See the [Katello Hammer CLI project](#) if you want to get setup for contributing to the hammer code.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Source available: [github/theforeman/theforeman.org](#)

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Activation Keys

Activation Keys provide a mechanism to define properties that may be applied to Content Hosts during registration. This includes properties such as:

- Lifecycle Environment
- Content View

- Subscriptions
- Repository Enablement
- Host Collections

Definitions

- [Content Host](#)
- Host Collection - A statically defined group of Content Hosts.
- Subscription - The right to receive the associated content from Katello.

General Features

The following is a high-level summary of the Activation Key features:

- [Create an Activation Key](#)
- [Add subscriptions to an Activation Key](#)
- [Change repository enablement for an Activation Key](#)
- [Add Host Collections to an Activation Key](#)
- [Register a Content Host using an Activation Key](#)

Create an Activation Key

To create a new key,

- navigate to: Content > Activation Keys
- click **New Activation Key**

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with tabs for 'Monitor', 'Content', 'Hosts', 'Configure', and 'Infrastructure'. On the far right, it shows 'Admin User' and 'Administer'. Below the navigation, the main content area has a title 'Activation Keys' and a search bar. A modal window is open in the center, titled 'New Activation Key'. It contains fields for 'Name*' (set to 'research'), 'Content Host Limit' (checkbox checked for 'Unlimited Content Hosts'), 'Description' (text area containing 'This is the activation key used by the research department.'), 'Environment' (a diagram showing 'Library' as the active environment, with 'dev' and 'test' also listed), and 'Content View' (set to 'Default Organization View'). At the bottom of the modal are 'Cancel' and 'Save' buttons.

- *Name*: This required option is used to identify the activation key to command line tools, such as `subscription-manager`.
- *Content Host Limit*: This option will control how many Content Hosts may be registered using the key. An “unlimited” value will not place any limits on usage. Specifying a quantity will limit the number of registered content hosts. Registering with an activation key consumes one of the available limit quantity, while unregistering makes it available again. (i.e. this quantity is not a usage counter but a limit of actively registered content hosts)
- *Description*: A free form text field that can be used to store a description of the key for later reference or for pseudo-tagging that can be used to search.

- *Environment and Content View:* Although optional, at least one activation key used during registration must specify a content view. Activation keys are used in the order specified to *subscription-manager* meaning the last activation key with a content view takes precedence.

The following example would use CV_B's content view:

```
subscription-manager register --org Default_Organization --activationkey NO_CV --activationkey CV_A --activationkey CV_B
```

Or equivalently:

```
subscription-manager register --org Default_Organization --activationkey NO_CV,CV_A,CV_B
```

For registration to succeed, at least one activation key must be successfully applied. For an activation key to succeed, at least one of the listed subscriptions must be successfully attached to the registering content host.

Add Subscriptions to an Activation Key

To add subscriptions to a key:

- navigate to: Content > Activation Keys
- select the desired key from the list
- click **Subscriptions**
- click **Add**
- select the Subscriptions you would like to add
- click **Add Selected**

The **Auto-Attach** setting controls how the group of subscriptions are processed during registration.

When 'Auto-Attach' is enabled but no subscriptions are added to the activation key, subscriptions will be automatically added to cover the installed products. This is equivalent to passing the '--auto-attach' flag to the subscription-manager command:

```
subscription-manager register --org=Default_Organization --auto-attach
```

When 'Auto-Attach' is enabled and subscriptions are listed for the activation key, two things will happen. First all subscriptions for custom products will be attached to the registering content host. Second, the group of Red Hat subscriptions will be attached as needed to cover the content host's installed Red Hat products. This is most commonly used when there is a group of similar subscriptions (eg. several Red Hat Enterprise Linux from different contracts, or guest subscriptions from different hypervisors) and which one used is not important. Katello's subscription tooling, [Candlepin](#), will automatically choose the minimal proper subscriptions from the group.

Finally, when 'Auto-Attach' is disabled, all subscriptions on the activation key will attach to the registering content host, regardless of whether needed to cover an installed product or not. For example, adding an OpenStack Platform subscription would then allow that product to be installed after registration.

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, Administer, and Admin User. Below the navigation is a search bar and a message indicating 1 total result. On the left, a sidebar lists activation keys, with 'research' selected and highlighted in blue. The main content area is titled 'Activation Key: research'. It has tabs for Details, Subscriptions (which is selected), Product Content, Host Collections, and Associations. Under 'Activation Key Type:', the 'Auto-Attach' setting is set to 'Yes' with a checked checkbox. A tooltip explains that auto-attach registers systems to all associated custom products and Red Hat subscriptions. Below this, there are 'List/Remove' and 'Add' buttons. A table titled 'Subscriptions' shows one entry: 'Red Hat Enterprise Linux Server, Premium (8 sockets) (Up to 4 guests)' with a quantity of 1, attached to 0 out of 25, type Physical, starting on 12/31/14, and expiring on 12/31/15. The account is listed as 5481114. Buttons for 'Remove Selected' and 'Copy Activation Key' are at the top right of the table.

Change Repository Enablement for an Activation Key

To change repository enablement settings using a key:

- navigate to: Content > Activation Keys
- select the desired key from the list
- click **Product Content**
- click the edit icon for the repository content set that you would like to modify
- select the desired value (e.g. 'Override to Yes', 'Override to No', 'Defaults to Yes', 'Defaults to No')
- click **Save**

View current settings:

FOREMAN

Default_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Activation Keys

Search... Showing 1 of 1 (1 Total) 0 Selected | Deselect All

Name
research >

Activation Key: research

[Remove](#) [Close](#)

Details Subscriptions **Product Content** Host Collections

Below are the repository content sets currently available for this activation key through its subscriptions. For Red Hat subscriptions, additional content can be made available through the [Red Hat Repositories page](#). Changing default settings for content hosts that register with this activation key requires subscription-manager version 1.10 or newer to be installed on that host.

RHEL 6

Content

Enabled by Default Defaults to Yes

Change current settings:

FOREMAN

Default_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Activation Keys

Search... Showing 1 of 1 (1 Total) 0 Selected | Deselect All

Name
research >

Activation Key: research

[Remove](#) [Close](#)

Details Subscriptions **Product Content** Host Collections

Below are the repository content sets currently available for this activation key through its subscriptions. For Red Hat subscriptions, additional content can be made available through the [Red Hat Repositories page](#). Changing default settings for content hosts that register with this activation key requires subscription-manager version 1.10 or newer to be installed on that host.

RHEL 6

Content

Enabled by Default

Add Host Collections to an Activation Key

To add Host Collections to a key:

- navigate to: Content > Activation Keys
- select the desired key from the list
- click **Host Collections**
- click **Add**
- select the Host Collections you would like to add
- click **Add Selected**

Activation Keys

Search... Showing 1 of 1 (1 Total)

0 Selected | Deselect All + New Activation Key

Name

research >

Activation Key: research

Details Subscriptions Product Content Host Collections

Host Collection Management

List/Remove Add

Filter

2 Selected | Deselect All Add Selected

Name	Description	Capacity
<input checked="" type="checkbox"/> development		0 / Unlimited
<input type="checkbox"/> finance		0 / Unlimited
<input checked="" type="checkbox"/> research		0 / Unlimited

Register a Content Host using an Activation Key

The simplest form of registering a content host with an activation key is this:

```
subscription-manager register --org=Default_Organization --activationkey=$KEY_NAME
```

[Click here for more information](#)

Note that modifying an activation key does not change anything on content hosts previously registered with the key.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Backup

In the following sections, these assumptions are being made with respect to making the backup:

- `/tmp/backup` will be used as the target for backups
- All commands are executed as `root`

Option One: Offline repositories backup

By default, the whole Katello instance will be turned off completely for the entire backup.

```
# katello-backup /tmp/backup
```

Option Two: Online repositories backup

Backing up the repositories can take an extensive amount of time. You can perform a backup while online. In order for this procedure to succeed, you must not change or update the repositories database until the backup procedure is complete. Thus, you must avoid publishing, adding, or deleting content views, promoting content view versions, adding, changing, or deleting sync-plans, and adding, deleting, or syncing repositories during this time. To perform an online-backup of the repositories, run:

```
# katello-backup --online-backup /tmp/backup
```

Option Three: Skip repositories backup

There may be situations in which you want to see a system without its repository information. You can skip backing up the Pulp database with the following option:

```
# katello-backup --skip-pulp /tmp/backup
```

Please note you would not be able to restore a Katello instance from a directory where the Pulp database was skipped.

Final check-up

After a successful backup, the backup directory should have the following files:

```
# ls /tmp/backup  
config_files.tar.gz  
mongo_data.tar.gz  
pgsql_data.tar.gz
```

Additionally, if you ran the backup without skipping the Pulp database, you will see the additional file:

```
pulp_data.tar
```

Katello instance should be up and running. Next chapter is dedicated to restoring a backup.

Restore

All the following commands are executed under `root` system account.

Please note only backups that include the Pulp database can be restored. To verify that your backup directory is usable, make sure it has the following files:

```
# ls /tmp/backup  
config_files.tar.gz  
mongo_data.tar.gz  
pgsql_data.tar.gz  
pulp_data.tar
```

Once verified, simply run:

```
# katello-restore /tmp/backup
```

This command will require verification in order to proceed, as the method will destruct all databases before restoring them. Once the procedure is finished, all processes will be online, and all databases and system configuration will be reverted to the state and the time of the backup.

Check log files for errors, such as `/var/log/foreman/production.log` and `/var/log/messages`.

[Foreman 2.3.3 has been released! Follow the quick start to install it.](#)

[Foreman 2.2.2 has been released! Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Capsules

What are Capsules?

The Capsule server is a Katello component that provides federated services to discover, provision, control, and configure hosts. Each Katello server includes a Default Capsule, and you may deploy additional Capsules to remote data centers. A Capsule

server provides the following features:

- Content features, including:
 - Repository synchronization
 - Content delivery
 - Host action delivery (package installation updates, etc)
 - Subscription management proxy (RHSM)
- Foreman Smart Proxy features, including:
 - DHCP, including ISC DHCP servers
 - DNS, including Bind and MS DNS servers
 - Realm, including FreeIPA
 - Any UNIX-based TFTP server
 - Puppet Master servers
 - Puppet CA to manage certificate signing and cleaning
 - Baseboard Management Controller (BMC) for power management
 - Provisioning template proxy

The Katello Capsule server is a means to scale out the Katello installation. Organizations can create various capsules in different geographical locations. These are centrally managed through the Katello server. When a Katello user promotes content to a particular environment, the Katello server will push the content to each of the Capsule servers subscribed to that environment. Hosts pull content and configuration from the Katello Capsule servers in their location and not from the central server.

In a fully configured capsule, communication is completely isolated between hosts and the Katello server.

How do Capsules differ from Smart Proxies?

A Katello Capsule is a Foreman Smart Proxy with the addition of content-related services.

Deployment

In the simplest use case, a user may only want to use the Default Capsule. Larger deployments would have a single Katello server with multiple Capsules attached, with these remote Capsules deployed to various datacenters. Capsules can also be used to scale the number of hosts attached to a single Katello server.

Installation

See [Capsule Installation](#)

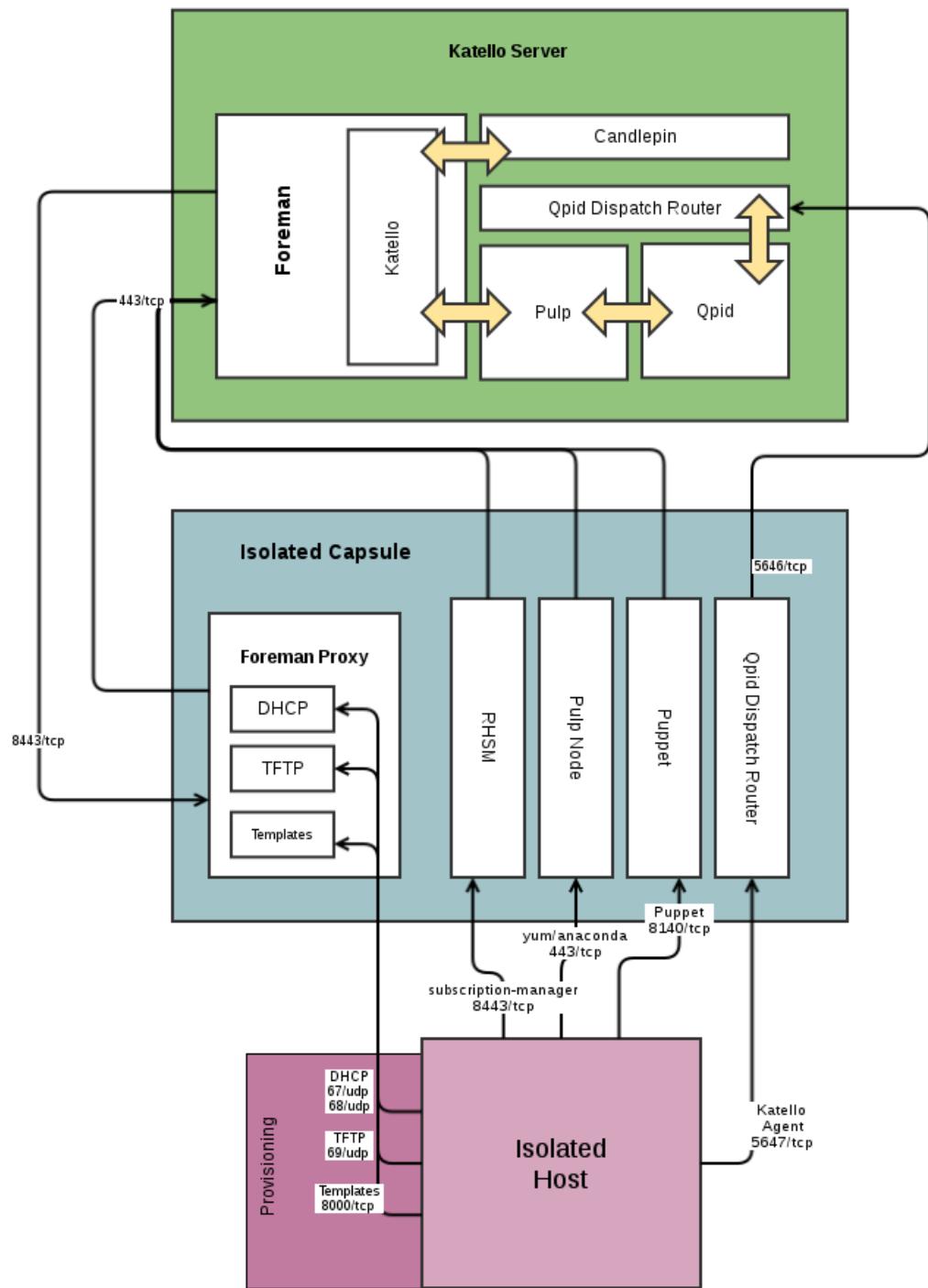
Removal

To stop all services and remove all Katello and Foreman related packages, run the following command as root on the capsule:

```
capsule-remove
```

Capsule Isolation

The goal of Capsule Isolation is to provide a single endpoint for all of a client's communication, so that in remote network segments, you need only open Firewall ports to the Capsule itself. The following section details the communication clients need to have with a Capsule. The installation options mentioned are the default starting with Katello 2.2.



Content and Configuration Services

There are five primary areas that require client communication:

1 - Content Delivery

That is, yum. Katello Capsules by default have the Pulp feature, which mirrors content for the selected Lifecycle Environments.

Install Option:

- `--pulp=true`

Required Connectivity:

- Clients need to be able to communicate with the Capsule on port 443/tcp.

2 - Katello Agent

The Katello agent is a goferd plugin which allows you to schedule remote actions on hosts such as package installation, updates, etc. A capsule must be running the Qpid Dispatch Router service for this feature to work.

Install Option:

- `--qpid-router=true`

Required Connectivity:

- Clients need to be able to communicate with the Capsule on port 5647/tcp

3 - Puppet & Puppet CA

By default, the Puppet CA feature on the Capsule is an independent CA which will manage the certificates for all the clients registered against the Capsule. Simply select the Puppetmaster and Puppet CA to be the Capsule when creating a host.

Install Option:

- `--puppet=true --puppetca=true .`

Required Connectivity:

- Clients need to communicate with the Capsule on port 8140/tcp.

4 - Subscription Management

Content Hosts utilize [Subscription Manager](#) for registration to Katello and enabling/disabling specific repositories.

Install Option:

- `--reverse-proxy=true`

Required Connectivity:

- Clients need to talk to the Capsule on port 8443/tcp.

5 - Provisioning Services

When provisioning a host using DHCP/PXE, you will need, at a minimum, the TFTP feature enabled on the capsule, and a DHCP server available. While not required, the Capsule can provide the DHCP service. In order for the installer to obtain its kickstart template from the Capsule, you should enable the templates feature.

If a TFTP proxy has the Templates feature as well, Foreman will automatically make the communication isolated. Your clients need to talk to the Capsule on port 67/udp and 68/udp for DHCP, 69/udp for TFTP, and 8000/tcp for Templates.

Consult the installer's `--help` for the full range of provisioning options.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

3.1 ▾

Katello 3.1 Documentation

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Content

Katello can currently host two different types of content, RPMs and puppet modules. RPMs and Puppet Modules can be synced from an external resource or can be uploaded directly.

The advantages to using Katello to mirror your local content are:

- Reduce bandwidth usage and increase download speed by having client machines pull updates from Katello

- Provision hosts using local Repositories covered HERE TODO
- Customize content locally, covered HERE TODO

Definitions

- Repository - Collection of content (either RPM or puppet)
- Product - Collection of Repositories, Content Hosts subscribe to a product
- Library - A special pre-created Lifecycle Environment where Repositories are created and content is synced or uploaded to. A Content Host can subscribe to library and receive content as soon as the content is synced or uploaded.

Creating a Product

From the web UI, navigate to:

Content > Products > New Product (top right)

Name	Value
Name*	Epel
Label*	Epel
GPG Key	(dropdown menu)
Sync Plan	(dropdown menu with '+ New Sync Plan' link)
Description	(text area)

Creating a Repository

From the web UI, navigate to:

Content > Products > Select desired product > Create Repository (right hand side)

Note the following options:

- Publish via HTTP: allows access to the Repository without any restriction. Unless you desire to restrict access to your content in this Repository, we recommended to leave this checked.
- URL: If you are syncing from an external Repository (yum or puppet), this would be filled in. This can be changed, added, or removed later. For example if you are wanting to create a mirror of EPEL, you would set this to 'https://dl.fedoraproject.org/pub/epel/6/x86_64/'.

The screenshot shows the Foreman web interface with the title 'New Repository'. The URL in the browser is 192.168.122.75:3000/katello/products#/products/127/repositories/new. The main navigation bar includes 'MyCompany', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', 'Administer', and a user icon for 'Admin User'. Below the navigation is a search bar and a message 'Showing 1 of 1 (1 Total)'. A toolbar at the top right includes '0 Selected | Deselect All', 'Bulk Actions', 'Repo Discovery', and '+ New Product'. On the left, a sidebar titled 'Name' lists 'Epel >'. The main content area is titled 'Product Epel' and shows tabs for 'Details', 'Repositories', and 'Tasks'. Under 'Details', there's a form for adding a new repository with fields: Name* (Epel x86_64), Label* (Epel_x86_64), Type* (yum), URL (http://download-i2.fedoraproject.org/pub/epel/6/x86_64/), Publish via HTTP (checked), and GPG Key (empty dropdown). Buttons for 'Cancel' and 'Save' are at the bottom.

Syncing a Repository

From the web UI, navigate to:

Content > Products > Select desired product > Select the Repository > Sync Now

The screenshot shows the Foreman web interface for managing repositories. The URL is 192.168.122.75:3000/katello/products#/products/127/repositories/80. The top navigation bar includes 'MyCompany', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', 'Administer', and a user icon for 'Admin User'. The main title is 'Products' with a sub-section 'Product Epel'. On the left, a sidebar lists 'Name' and 'Epel >'. The main content area has tabs for 'Details', 'Repositories', and 'Tasks'. Under 'Details', there's a 'Basic Information' section with fields like Name (Epel x86_64), Label (Epel_x86_64), Type (yum), URL (http://download-i2.fedoraproject.org/pub/epel/6/x86_64/), Publish via HTTPS (true), Publish via HTTP (unchecked), Published At (http://centos.installer/pulp/repos/MyCompany/Library/custom/Epel/Epel_x86_64), and GPG Key (unchecked). To the right, there's a 'Content Counts' section with 'Content Type' sub-sections for Packages (0), Errata (0), and Package Groups (0). Buttons for 'Sync Now' and 'Remove Repository' are also present. A note at the bottom says 'All dates and times below are relative to this server'.

The progress will be displayed:

The screenshot shows the Foreman interface for viewing task details. The URL is 192.168.122.75:3000/katello/products#/products/127/tasks/fbd1fd1d-8f38-4a4e-aae0-ae4a74df6846. The top navigation bar and sidebar are identical to the previous screenshot. The main title is 'Product Epel'. The 'Repositories' tab is selected. The 'Task Details' section shows an action type of 'Synchronize', user 'admin', started at '4/15/14 12:52 PM', finished at '4/15/14 12:52 PM', parameters 'repository 'Epel x86_64' ; product 'Epel' ; organization 'MyCompany'', state 'running', and result 'pending'. Below this, a 'Details' section shows 'New packages: 37/10694 (6.68 MB/9.78 GB)'.

Syncing multiple repositories

To easily sync multiple repositories at once and track their progress, navigate to:

Content > Sync Status

From here you can expand the desired products, and select multiple repositories to sync.

The screenshot shows a web browser window titled "Sync Status" from the Foreman interface. The URL is "abed.usersys.redhat.com:3000/katello/sync_management#!=". The top navigation bar includes links for "MyCompany", "Monitor", "Content", "Hosts", "Configure", and "Infrastructure". On the right, there are user and administrator dropdown menus. The main content area is titled "Sync Status" and contains a table with the following data:

PRODUCT	START TIME	DURATION	SIZE (PACKAGES)	RESULT
▼ Epel			0 Bytes	
<input checked="" type="checkbox"/> Epel i386			0 Bytes (0)	
<input checked="" type="checkbox"/> Epel x86_64			0 Bytes (0)	

A "Synchronize Now" button is located at the bottom right of the table.

Uploading RPM Content

Uploading RPM content directly is not currently supported. You will need to build a custom yum Repository. TODO Provide instructions on creating a custom yum repo

Uploading Puppet Content

To upload puppet modules, first create a Repository with type puppet (similarly to creating a yum Repository above):

The screenshot shows the Foreman web interface with a blue header bar. The title bar says "New Repository". The main navigation menu includes "MyCompany", "Monitor", "Content", "Hosts", "Configure", and "Infrastructure". On the right, there's an "Administer" section with a user icon and "Admin User". Below the header, a search bar and a message "Showing 2 of 2 (2 Total)" are visible. A toolbar at the top right includes "0 Selected | Deselect All", "Bulk Actions", "Repo Discovery", and "+ New Product".

The main content area is titled "Product Configuration" under "Product". It has tabs for "Details", "Repositories", and "Tasks", with "Repositories" selected. A sub-section titled "Add New Repository" contains fields for "Name*" (Puppet), "Label*" (Puppet), "Type*" (puppet), "URL" (empty), "Publish via HTTP" (checked), and "GPG Key" (empty). Buttons for "Cancel" and "Save" are at the bottom.

When creating this Repository the URL field can be left blank.

Puppet modules can be uploaded via the Web UI, navigate to:

Content > Repositories > Products > Select desired Product > Select desired Puppet Repository > Select file on the right

This screenshot shows the same Foreman interface as the previous one, but for an existing repository named "Puppet". The title bar says "Repository: Puppet". The main content area shows the "Product Configuration" for the "Puppet" repository. The "Repositories" tab is selected. The "Basic Information" section lists the repository details: Name (Puppet), Label (Puppet), Type (puppet), URL (empty), Publish via HTTPS (true), Publish via HTTP (checked), Published At (http://localhost/pulp/repos/MyCompany/Library/custom/Configurati/Puppet), and GPG Key (empty). To the right, there are sections for "Content Counts" (Puppet Modules: 0) and "Content Type". A "Sync Now" button and a "Remove Repository" button are also present. A "Upload Puppet Module" form with "Choose File" and "Upload" buttons is shown. At the bottom, a "Sync Status" section indicates "All dates and times below are relative to this server".

Subscribing a System to a Product for yum content

To read about registering systems and subscribing them to the Product click TODO.

Scheduling Repository Synchronization

Sync plans give you the ability to schedule Repository synchronization on a hourly, daily or weekly basis. Sync Plans are applied to Products and thus all Repositories within a Product will be synchronized according to the products plan.

Creating a Sync Plan

If you would like to schedule certain repositories to sync on a hourly, daily or weekly basis, Sync Plans give you this capability.

To create a Sync Plan, navigate to:

Content > Sync Plans > click "New Sync Plan" on the upper right

The *Start Date* and *Start Time* fields are used as the day of the week/month and time of the day to run the re-occurring syncs.

For example a sync plan that starts on Sunday 2014-04-06 at 2:30 will occur every Sunday at 2:30 every week if it has a weekly interval. If on a daily interval it would sync every day at 2:30.

The screenshot shows the Katello interface with a blue header bar. The title bar says 'New Sync Plan'. The main content area is titled 'Sync Plans' and shows a table with one row. The row has a checkbox labeled 'Name' and the value 'Weekly Sync'. To the right of the table is a 'New Sync Plan' dialog box. The dialog box contains the following fields:

- Name*: Weekly Sync
- Description: (empty)
- Interval: weekly
- Start Date*: 2014-04-06
- Start Time (-0400 GMT)*: 02 : 30 (with up and down arrows for hours and minutes)

At the bottom of the dialog box are 'Cancel' and 'Save' buttons. The 'Save' button is highlighted with a blue border.

Assigning a Sync Plan to a Product

Navigate to:

Content > Sync Plans > Select your Sync Plan > Products > Add

Then select the Products you want to add and click "Add Selected" in the upper right.

The screenshot shows the Foreman interface for managing sync plans. The top navigation bar includes links for Sync Plans, MyCompany, Monitor, Content, Hosts, Configure, Infrastructure, Administer, and a user icon for Admin User.

The main title is "Sync Plans". A search bar shows "Showing 1 of 0 (0 Total)". There are buttons for "0 Selected | Deselect All" and "New Sync Plan".

A sidebar on the left lists "Name" and "Weekly Sync >". The main content area is titled "Sync Plan Weekly Sync". It has tabs for "Details" (selected) and "Products".

Under "Product Management", there are buttons for "List/Remove" and "Add". A filter input field is present. Below is a table:

Name	Description	Sync Status	Repositories
Configuration		Never synced	1
Epel		Never synced	2

GPG Keys

To learn about securing your packages and Repositories with GPG Keys click here. TODO

Foreman 2.3.3 has been released! Follow the quick start to install it.
Foreman 2.2.2 has been released! Follow the quick start to install it.

[View our Privacy Policy](#)

This web site is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Content Hosts

What is a Host?

A Host is a Foreman concept that represents a server/host/system/computer (whatever you want to call it). In addition to holding facts about the system, it:

- Stores which operating system the system should be running
- Stores which puppet classes should be assigned
- Stores which parameters apply to which puppet classes
- Allows you to re-provision the machine

What are Content Hosts?

Content Hosts are the part of a host that manages Content and Subscription related tasks. As time goes on more and more of this functionality will be moved to the Host object. A Host's Content Host:

- Stores which Products are assigned (i.e. which Repositories will the system pull content from)
- Initiates package install/upgrade/removal
- Determines which errata apply to a system
- Initiates errata installation

How is a Content Host registered?

Subscription Manager is the client for Katello that handles registration.

Installing Subscription Manager

Depending on your Operating System, for:

- RHEL, subscription-manager is installed by default
- Fedora, subscription-manager is available from the Everything repo for its release: `yum install subscription-manager`
- CentOS 7, subscription-manager is available in the 'os' repo for its release: `yum install subscription-manager`
- CentOS 5/6, enable the upstream subscription-manager repo and then install subscription-manager (be sure to change '6' to '5' if you're on EL5, as the version from 6 will not work):

```
wget -O /etc/yum.repos.d/subscription-manager.repo http://copr.fedoraproject.org/coprs/dgoodwin/subscription-manager/repo/epel-6/dgoodwin-subscription-manager-epel-6.repo  
yum install subscription-manager -y
```

Registering with Subscription Manager

First install the bootstrap rpm from your Katello server:

```
rpm -Uvh http://$KATELLO_HOSTNAME/pub/katello-ca-consumer-latest.noarch.rpm
```

Then register:

```
subscription-manager register --org=Default_Organization --environment=Library
```

Subscription manager will prompt for your username and password. You can also specify `--username $USER --password $PASS` on the command line.

Registering to a Content View

To register to Content View "MyView" in a "Devel" Lifecycle Environment:

```
subscription-manager register --org=Default_Organization --environment=Devel/MyView
```

Registering without using a username and password

Activation Keys allow you to register and consume content without using a username and password. To create an Activation Key see the [Activation Key Guide](#)

Once you have created an activation key, register with:

```
subscription-manager register --org=Default_Organization --activationkey=$KEY_NAME
```

Actions with registered Content Hosts

To see the list of your Content Hosts, navigate to Hosts > Content Hosts

Content Hosts						
Search... <input type="text"/>		Showing 5 of 5 (5 Total)		0 Selected Deselect All <input type="button"/> Bulk Actions <input type="button"/> Register Content Host		
Name	Subscription Status	OS	Environment	Content View	Registered	Last Checkin
client.devel	●	Red Hat Enterprise Linux Server 6.4		Default Organization View	7/3/14 10:02 AM	7/3/14 10:02 AM
dhcp129-73.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		Default Organization View	6/19/14 10:30 AM	7/10/14 12:56 PM
dhcp129-81.rdu.redhat.com	●	Red Hat Enterprise Linux Server 7.0		Default Organization View	6/18/14 3:16 PM	6/23/14 7:15 PM
testks.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		MyDistro	6/19/14 4:45 PM	6/20/14 4:47 PM
testks.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		ErrataTest	6/19/14 4:26 PM	6/19/14 4:30 PM

Changing the Lifecycle Environment and Content View of a Content Host:

Navigate to the Content Host Details page, Host > Content Hosts > Click the name of the desired Content Host

Content Hosts						
Search... <input type="text"/>		Showing 5 of 5 (5 Total)		0 Selected Deselect All <input type="button"/> Bulk Actions <input type="button"/> Register Content Host		
Name	Subscription Status	OS	Environment	Content View	Registered	Last Checkin
client.devel	●	Red Hat Enterprise Linux Server 6.4		Default Organization View	7/3/14 10:02 AM	7/3/14 10:02 AM
dhcp129-73.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		Default Organization View	6/19/14 10:30 AM	7/10/14 12:56 PM
dhcp129-81.rdu.redhat.com	●	Red Hat Enterprise Linux Server 7.0		Default Organization View	6/18/14 3:16 PM	6/23/14 7:15 PM
testks.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		MyDistro	6/19/14 4:45 PM	6/20/14 4:47 PM
testks.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		ErrataTest	6/19/14 4:26 PM	6/19/14 4:30 PM

Content Host dhcp129-73.rdu.redhat.com						
Details		Provisioning Details	Subscriptions	Host Collections	Tasks	Packages
Basic Information Name: dhcp129-73.rdu.redhat.com <input checked="" type="checkbox"/> UUID: 2df6452c-852d-4a58-b6ac-728362f913c1 <input checked="" type="checkbox"/> Description: Initial Registration Params <input checked="" type="checkbox"/> Type: Virtual Guest <input checked="" type="checkbox"/>		Content Host Content Release Version: <input checked="" type="checkbox"/> Content View: Default Organization View <input checked="" type="checkbox"/> Environment: Library Dev				
Subscriptions Subscription Status: invalid <input checked="" type="checkbox"/> Details: Red Hat Enterprise Linux Server - Not supported by a valid subscription. <input checked="" type="checkbox"/> Auto-Attach: Yes <input checked="" type="checkbox"/> Service Level: <input checked="" type="checkbox"/> Activation Keys: None <input checked="" type="checkbox"/>		Content Host Status Registered: 6/19/14 10:30 AM <input checked="" type="checkbox"/> Checkin: 7/10/14 12:56 PM <input checked="" type="checkbox"/>				
Content Host Properties OS: Red Hat Enterprise Linux Server <input checked="" type="checkbox"/> Release: 2.6.32-431.el6.x86_64 <input checked="" type="checkbox"/> Arch: x86_64 <input checked="" type="checkbox"/>		Networking Hostname: dhcp129-73.rdu.redhat.com <input checked="" type="checkbox"/> IPv4 Address: 10.13.129.73 <input checked="" type="checkbox"/> IPv6 Address: ::1 <input checked="" type="checkbox"/> Interfaces: + eth0 <input checked="" type="checkbox"/> + lo <input checked="" type="checkbox"/>				

Look in the upper right corner for the "Content Host Content":

Content Host Content

Release Version



Content View

Default Organization View



Environment



Then select the new Lifecycle Environment you desire, select the new Content View you desire, and click save.

Assigning a Content Host to a Product

In order for a Content Host to receive package updates and access Repositories hosted on Katello, it needs to be subscribed to a product.

Navigate to Hosts > Content Hosts > Select Content Host > Subscriptions > Click the "Add" tab

The screenshot shows the Foreman interface with the 'Content Hosts' page open. The host 'Content Host dhcp129-73.rdu.redhat.com' is selected. The 'Subscriptions' tab is active, displaying a message 'Successfully removed 1 subscriptions.' and a table of attached products:

Quantity	Attached	Starts	Expires	Support Level	Contract	Account
1	0 out of Unlimited	6/18/14	6/10/44	cds		
1	1 out of Unlimited	6/4/14	5/27/44	testProduct		

Check the checkbox under the Products you want to add and select "Add Selected" in the upper right.

To see existing attached Products, click the "List/Remove" tab. To remove a Product, select the checkbox under the desired Product in this list and click "Remove Selected".

Package Management

To perform package actions on a single Content Host, navigate to: Hosts > Content Hosts > Select Content Host > Packages

Content Hosts

Showing 5 of 5 (5 Total)

0 Selected | Deselect All | Bulk Actions | Register Content Host | Unregister Content Host | Close

Name
client.devel
dhcp129-73.rdu.redhat.com
dhcp129-81.rdu.redhat.com
testks.rdu.redhat.com
testks.rdu.redhat.com

Content Host dhcp129-73.rdu.redhat.com

Details Provisioning Details Subscriptions Host Collections Tasks Packages Errata Product Content

Package Actions

Package Install Package/Group Name Perform

Installed Packages

Filter Showing 416 Packages Update All

Installed Package	Remove
acl-2.2.49-6.el6.x86_64	X
aic94xx-firmware-30-2.el6.noarch	X
apr-1.3.9-5.el6_2.x86_64	X
apr-util-1.3.9-3.el6_0.1.x86_64	X
apr-util-ldap-1.3.9-3.el6_0.1.x86_64	X
atmel-firmware-1.3-7.el6.noarch	X
attr-2.4.44-7.el6.x86_64	X
audit-2.2-4.el6_5.x86_64	X

From here you can:

- See a list of installed packages
- Perform a yum install/update/remove of a Package or Package Group
- Update all packages (equivalent of running 'yum update')

View and Install Applicable Errata

If your synced Repositories contain Errata, you can use Katello's Errata management to track and install Errata.

Navigate to: Hosts > Content Hosts > Select Content Host > Errata

Content Hosts

Showing 1 of 1 (1 Total)

0 Selected | Bulk Actions | Register Content Host | Unregister Content Host | Close

Name
client.devel.foo6

Content Host client.devel.foo6

Details Provisioning Details Subscriptions Host Collections Tasks Packages Errata Product Content

Applicable Errata

Show from: Current Environment (dev/incremental/test)

Showing 231 of 231 (231 Total) 0 Selected | Apply Selected

Type	Id	Title	Issued
Bug Fix Advisory	RHBA-2014:1909	lvm2 bug fix update	11/25/14
Bug Fix Advisory	RHBA-2014:1734	yum-rhn-plugin bug fix update	10/29/14
Product Enhancement Advisory	RHEA-2014:1733	tzdata enhancement update	10/28/14
Bug Fix Advisory	RHBA-2014:1666	gcc enhancement update	10/19/14
Security Advisory - Moderate	RHSA-2014:1671	Moderate: rsyslog5 and rsyslog security update	10/19/14
Security Advisory - Moderate	RHSA-2014:1655	Moderate: libxml2 security update	10/15/14

To apply errata, search for the errata you want and select the checkbox beside each errata. Then click "Apply Selected" at the top right.

The "Show From" filters what applicable errata to show:

- Current Environment - Shows only Applicable Errata available in the Host's Content View & Lifecycle Environment.
- Previous Environment - Shows Applicable Errata that are available from the Host's Content View but in the previous Lifecycle Environment. Promoting the Content View Version from that previous Lifecycle Environment to the current Lifecycle Environment for this Host would cause all Applicable Errata shown to then be available.
- Library Synced Content - Shows Applicable Errata which have been synced to the Library. This shows you what is applicable even when the Errata have not been published into a Content View. All applicable Errata are shown regardless of availability to the Content Host.

Change Host Collection Asssignments

To change Host Collection assignments for a Content Host, navigate to: Hosts > Content Hosts > Select Content Host > Host Collections

Content Hosts

Showing 5 of 5 (5 Total)

0 Selected | Deselect All | Bulk Actions | Register Content Host

Unregister Content Host | Close

Name	Description	Capacity
Database Servers		1 / Unlimited

Bulk Actions

Katello provides the ability to perform actions on many Content Hosts at once such as:

- Package installation/upgrade/removal
- Listing and applying applicable errata
- Assigning Host Collections
- Changing Lifecycle Environment and Content View assignments

In order to use the bulk actions, perform whatever search you desire and select which Content Hosts you want to modify. If you want to select all Content Hosts from a search result, click the 'checkbox' above the table:

Content Hosts

<input type="checkbox"/>	Name	Subscription Status	OS	Environment	Content View	Registered	Last Checkin
<input type="checkbox"/>	client-devel	●	Red Hat Enterprise Linux Server 6.4	Library	Default Organization View	7/3/14 10:02 AM	7/3/14 10:02 AM
<input type="checkbox"/>	dhcp129-73.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5	Library	Default Organization View	6/19/14 10:30 AM	7/11/14 12:56 PM
<input type="checkbox"/>	dhcp129-81.rdu.redhat.com	●	Red Hat Enterprise Linux Server 7.0	Library	Default Organization View	6/18/14 3:16 PM	6/23/14 7:15 PM
<input type="checkbox"/>	test1-devel.example.com	●		Dev	StableOS	7/11/14 4:23 PM	Never checked in
<input type="checkbox"/>	test10-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test11-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test12-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test13-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test14-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test15-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test16-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test17-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input type="checkbox"/>	test18-devel.example.com	●		Dev	StableOS	7/11/14 4:33 PM	Never checked in
<input type="checkbox"/>	test19-devel.example.com	●		Dev	StableOS	7/11/14 4:33 PM	Never checked in

This will select all Content Hosts on that page (only the ones that are visible). To select all that correspond to that search query, notice a bar has now appeared:

<input type="checkbox"/>	Name	Subscription Status	OS	Environment	Content View	Registered	Last Checkin
All 20 items on this page are selected; Select all 35.							
<input checked="" type="checkbox"/>	client-devel	●	Red Hat Enterprise Linux Server 6.4	Library	Default Organization View	7/3/14 10:02 AM	7/3/14 10:02 AM
<input checked="" type="checkbox"/>	dhcp129-73.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5	Library	Default Organization View	6/19/14 10:30 AM	7/11/14 12:56 PM
<input checked="" type="checkbox"/>	dhcp129-81.rdu.redhat.com	●	Red Hat Enterprise Linux Server 7.0	Library	Default Organization View	6/18/14 3:16 PM	6/23/14 7:15 PM
<input checked="" type="checkbox"/>	test1-devel.example.com	●		Dev	StableOS	7/11/14 4:23 PM	Never checked in
<input checked="" type="checkbox"/>	test10-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test11-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test12-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test13-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test14-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test15-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test16-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test17-devel.example.com	●		Dev	StableOS	7/11/14 4:32 PM	Never checked in
<input checked="" type="checkbox"/>	test18-devel.example.com	●		Dev	StableOS	7/11/14 4:33 PM	Never checked in
<input checked="" type="checkbox"/>	test19-devel.example.com	●		Dev	StableOS	7/11/14 4:33 PM	Never checked in

Next select the 'Bulk Actions' button in the top right.

From here you can select the tab corresponding to any action you wish to perform.

Foreman 2.3.3 has been released! [Follow](#) the quick start to install it.

Foreman 2.2.2 has been released! [Follow](#) the quick start to install it.

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

1. Installation

[1.1 Katello](#)[1.2 Capsule](#)[1.3 Clients](#)

2. Upgrade

[2.1 Katello](#)[2.2 Capsule](#)[2.3 Clients](#)

3. Release Notes

4. CLI

5. User Guide

[5.1 Activation Keys](#)[5.2 Backup](#)[5.3 Capsules](#)[5.4 Content](#)[5.5 Content Hosts](#)[5.6 Content Views](#)[5.7 Disconnected](#)[5.8 Docker Management](#)[5.9 Email Notifications](#)[5.10 Errata](#)[5.11 Glossary](#)[5.12 GPG Keys](#)[5.13 Host Collections](#)[5.14 Lifecycle Environments](#)[5.15 Provisioning](#)[5.16 Puppet Integration](#)[5.17 Red Hat Content](#)

6. Troubleshooting

Content Views

What can a Content View be used for?

- To stage content through environments (Dev, Test, Production).
- To filter the contents of a repository (include a package or exclude certain errata, for example).
- To have multiple snapshots of the same repository and/or puppet modules.

Definitions

- Content View - snapshot of one or more repositories and/or puppet modules.
- Composite Content View - a Content View that contains a collection of other Content Views.
- Filter - provides finer grained control over content in a Content View. Can be used to include or exclude specific packages, package groups, or errata.
- Publishing - Content Views are ‘published’ in order to lock their contents in place. The content of the Content View is cloned and all filters applied. Publishing creates a new version of the Content View.
- Promoting - Content Views can be cloned to different [Lifecycle Environments](#) (Dev, Test, Production).

General Workflow

First [create a product and repository](#) in the library environment and populate the repository with content (by syncing it or uploading content). A [Content Host](#) can now register directly to library and be attached to the content therein. Updates will be available as soon as new content is synced or uploaded.

To utilize Content Views for filtering and snapshotting:

1. Create a Content View
2. Add the desired repository and/or puppet modules to the Content View
3. Optionally create one or more Filters to fine tune the content of the Content View.
4. Publish the Content View
5. Attach the Content Host to the Content View
6. Optionally promote the Content View to another environment

At this point the Content Host will no longer be getting content directly from Library, but from the Content View. Updates to library will not affect this Content Host.

Note that all of the actions below can also done with [hammer, the CLI tool](#), and examples are given at the end of each section.

Creating a Content View

To create a Content View using the web UI, navigate to:

Content > Content Views

Click the *Create New View* button on the top right of the screen.

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Admin User, Administer, and various system sections like Monitor, Content, Hosts, Configure, and Infrastructure. Below the navigation is a search bar and a message indicating 3 total results. On the left, there's a sidebar with a 'Content Views' section containing checkboxes for Name, COMPOSITE, Puppet Modules, and Yum Repos. The main area is titled 'New Content View' and contains a 'View Details' form. The 'Name*' field is filled with 'New Content View', the 'Label*' field with 'New_Content_View', and the 'Description' field with the placeholder 'This is my new content view, not sure what it will contain yet.' There's also a 'Composite View?' checkbox which is unchecked. At the bottom of the dialog are 'Cancel' and 'Save' buttons.

From the CLI:

```
hammer content-view create \
--organization="Default Organization" \
--name="New Content View" \
--description="This is my new content view."
```

Creating a Composite Content View

To create a Composite Content View using the web UI follow the above steps for [Creating a Content View](#) but check the “Composite View?” checkbox.

From the CLI:

```
hammer content-view create \
--organization="Default Organization" \
--name="New Composite Content View" \
--description="This is my new composite content view." \
--composite
```

Adding Repositories

Adding a repository to a Content View means whenever a Content View is published, all of the content contained within the repository at that time is included in the Content View. If the [repository is synced](#) after publishing the Content View, the Content View will contain the state of the repository prior to syncing. A new version of the Content View must be published in order for the new version to get the contents of the newly synced repository.

To add a repository using the web UI, navigate to:

Content > Content Views > Select the desired Content View > Content (within sub navigation) > Repositories

Name	Product	Last Sync	Sync State	Content
Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server	Red Hat Enterprise Linux Server	N/A	N/A	0 Packages 0 Errata
Red Hat Enterprise Linux 6 Server - Supplementary RPMs i386 6Server	Red Hat Enterprise Linux Server	N/A	N/A	0 Packages 0 Errata

From the CLI, adding a repository:

```
hammer content-view add-repository \
--organization="Default Organization" \
--name="New Content View" \
--repository="CentOS 6.5"
```

Adding a Puppet Module

Adding a puppet module to a Content View means that whenever the Content View is published the puppet module is locked to the version selected. If the “Use Latest” version is selected then the puppet module will be “frozen” at the latest version available when the Content View is published. A new version of the Content View must be published in order for the new version to get any updated puppet module.

To add a puppet module using the web UI, navigate to:

Content > Content Views > Select the desired Content View > Puppet Modules (within sub navigation)

FOREMAN

content views Monitor Content Hosts Configure Infrastructure

Admin User Administer

Content Views

Search... Showing 4 of 4 (4 Total) 0 Selected | Deselect All + Create New View

Name
 COMPOSITE
 New Content View >
 Puppet Modules
 Yum Repos

New Content View

Versions Content **Puppet Modules** History Details Tasks

Currently Selected Puppet Modules

Filter Showing 0 of 0 (0 Total) 0 Selected | Deselect All + Add New Module

You currently don't have any Puppet Modules included in this Content View, you can add puppet modules using the button on the right.

FOREMAN

content views Monitor Content Hosts Configure Infrastructure

Admin User Administer

Content Views

Search... Showing 4 of 4 (4 Total) 0 Selected | Deselect All + Create New View

Name
 COMPOSITE
 New Content View >
 Puppet Modules
 Yum Repos

New Content View

Versions Content **Puppet Modules** History Details Tasks

Module List

Select A New Puppet Module To Add

Filter Showing 4 of 4 (4 Total)

Name	Actions
apache	Select a Version
postgresql	Select a Version
registry	Select a Version
stdlib	Select a Version

The screenshot shows the Foreman interface for creating a new Content View. The left sidebar has 'New Content View' selected. The main area shows a table of available Puppet modules:

Author	Version	Summary	Repositories	Actions
puppetlabs	Use Latest (currently 1.0.1)	Puppet module for Apache	Puppet	<button>Select Version</button>
puppetlabs	1.0.1	Puppet module for Apache	Puppet	<button>Select Version</button>

The screenshot shows the Foreman interface for creating a new Content View. The 'Content' tab is selected. The main area shows the currently selected Puppet module:

Name	Author	Version	Actions
apache	puppetlabs	Latest (Currently 1.0.1)	<button>Select new version</button> <button>Remove Module</button>

From the CLI, first find the UUID of your puppet module from the list:

```
hammer puppet-module list \
--organization="Default Organization" \
--repository "Puppet Modules"
```

Then add the puppet module:

```
hammer content-view puppet-module add \
--organization="Default Organization" \
--content-view="New Content View" \
--uuid=91cc9bb7-dbb3-4798-b50a-45173b763cbb
```

Adding Content Views to a Composite Content View

Adding a version of a Content View to a Composite Content View means whenever the Composite Content View is published, all of the content contained within the specific version of that Content View is contained in the Composite Content View. If the Content Views contained within the Composite Content View are updated (i.e. a new version is published) or if their content is

updated after publishing the Composite Content View, the Composite Content View will only contain the versions of the Content View(s) prior to syncing. A new version of the Composite Content View must be published in order for it to get the updated Content Views.

To add a Content View to a Composite Content View using the web UI, navigate to:

Content > Content Views > Select the desired Content View > Content (within sub navigation) > Repositories

The screenshot shows the Foreman web interface with the following details:

- Header:** FOREMAN, Admin User, Administer.
- Left Sidebar:** Content Views, showing a list of content views: Name (COMPOSITE), New Content View, Puppet Modules, Yum Repos. The "COMPOSITE" view is selected.
- Main Content Area:** Title: Composite Content View COMPOSITE. Sub-tabs: Versions (selected), Content Views, History, Details, Tasks.
- Buttons:** Publish New Version, + Create New View, List/Remove, Add.
- Message:** In order to add a content view to a composite view you must first publish an initial version of the content view.
- Table:** Filter... 1 Selected | Deselect All | + Add Content Views. The table lists content views:

Name	Version	Environment	Description	Content
Puppet Modules	1	Library	gdsfhdfh	0 Repositories 0 Puppet Modules
Yum Repos		prod, test, Library		1 Repositories 0 Puppet Modules

Find the Content View ID of the specific version of the Content View to add:

```
hammer content-view version list \
--organization="Default Organization" \
--content-view="New Content View"
```

From the CLI, add a Content View to a composite Content View:

```
hammer content-view update \
--organization="Default Organization" \
--content-view="New Composite Content View" \
--component-ids=2
```

Creating a filter

If only using Content Views as snapshots, Filters are unnecessary. If the desire is to filter what content make it into the view, such as blacklisting a package by name or version, or blacklisting errata by date or type, Filters can help accomplish these tasks.

To create a new Content View Filter using the web UI, navigate to:

Content > Content Views > Select the desired Content View > Content (within sub navigation) > Filters > New Filter

FOREMAN

content views Monitor Content Hosts Configure Infrastructure

Admin User Administer

Content Views

Search... Showing 4 of 4 (4 Total) 0 Selected | Deselect All + Create New View

Name
 COMPOSITE
 New Content View >
 Puppet Modules
 Yum Repos

New Content View

Versions Content ▾ Puppet Modules History Details Tasks

Filters

Search... Showing 0 of 0 (0 Total) 0 Selected | Deselect All + New Filter Remove Filters

You currently don't have any Filters included in this Content View, you can add a new Filter by using the button on the right.

FOREMAN

content views Monitor Content Hosts Configure Infrastructure

Admin User Administer

Content Views

Search... Showing 4 of 4 (4 Total) 0 Selected | Deselect All + Create New View

Name
 COMPOSITE
 New Content View >
 Puppet Modules
 Yum Repos

New Content View

Versions Content ▾ Puppet Modules History Details Tasks

< Filters List

Add New Filter

Name* New Filter

Content Type* Package

Type* Exclude

Description Here is my new filter!

Cancel Save

The screenshot shows the Foreman web interface under the 'Content Views' section. On the left, there's a sidebar with options like 'Name', 'COMPOSITE', 'New Content View >', 'Puppet Modules', and 'Yum Repos'. The 'New Content View >' option is highlighted with a blue background. The main content area is titled 'New Content View' and shows a sub-section for 'Filters'. A table lists a single filter rule:

Package Name	Detail
something-else	Equal To 10.0.0
httpd	All Versions

From the CLI, adding a Content View Filter:

```
hammer content-view filter create \
--organization="Default Organization" \
--content-view="New Content View" \
--name="New Filter" \
--inclusion=false \
--type=rpm
```

From the CLI, adding a Content View Filter rule:

```
hammer content-view filter rule create \
--organization="Default Organization" \
--content-view="New Content View" \
--content-view-filter="New Filter" \
--name="something-else" \
--max-version="10.0.0" \
--min-version="10.0.0"
```

Selecting which Repositories to Filter

By default a Filter applies to all repositories (present and future) in the Content View. It's possible to select which repositories within the Content View apply to the filter. This is useful, for example, if the desire is to exclude errata from only certain repositories in a view.

To select which repositories to Filter in the web UI, navigate to:

Content > Content Views > Select the desired Content View > Content (within sub navigation) > Filters > Select the desired Filter > Affected repositories (within sub navigation)

New Content View

Versions Content Puppet Modules History Details Tasks

Filters / New Filter

New Filter (Exclude Packages)

Packages Affected repositories

This filter applies to all repositories in the content view (current and future).
This filter applies only to a subset of repositories in the content view.

Affected?	Name	Product	Type	Sync Status	Content
<input checked="" type="checkbox"/>	Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server	Red Hat Enterprise Linux Server	yum	N/A	0 Packages 0 Errata
<input type="checkbox"/>	Red Hat Enterprise Linux 6 Server - Supplementary RPMs i386 6Server	Red Hat Enterprise Linux Server	yum	N/A	0 Packages 0 Errata

All Products Filter Update Repositories

From the CLI, adding a Content View Filter:

```
hammer content-view filter update \
--organization="Default Organization" \
--name="New Filter" \
--repository-ids=2,3,7
```

Publishing a Content View

Publishing a Content View produces a new version of the content view that is subsequently promoted to the Library lifecycle environment. This newly published version of the content view is now available to any content host registered to Library.

To publish a Content View, in the web UI, navigate to:

Content > Content Views > Select the desired Content View > Publish New Version

New Content View

Versions Content Puppet Modules History Details Tasks

Publish New Version

A new version of New Content View has been promoted to the Library environment. It can be promoted to other environments from the Versions tab of this Content View.

Version Details

Version 1

Comment initial version.

Cancel Save

The screenshot shows the Foreman interface for managing content views. On the left, there's a sidebar with options like 'Name', 'COMPOSITE', 'New Content View >', 'Puppet Modules', and 'Yum Repos'. The main area shows a search bar and a message 'Showing 4 of 4 (4 Total)'. A modal window titled 'New Content View' is open, showing a table with one row for 'Version 1'. The table columns are 'Version', 'Status', 'Environments', 'Content', 'Author', and 'Actions'. The 'Status' column has a green bar with the text 'Publishing and promoting to 1 environment.'. The 'Content' column shows '0 Packages' and '0 Errata'. In the 'Actions' column, there's a button labeled 'Promote'.

From the CLI:

```
hammer content-view publish \
--organization="Default Organization" \
--name="New Content View"
```

Registering a Content Host

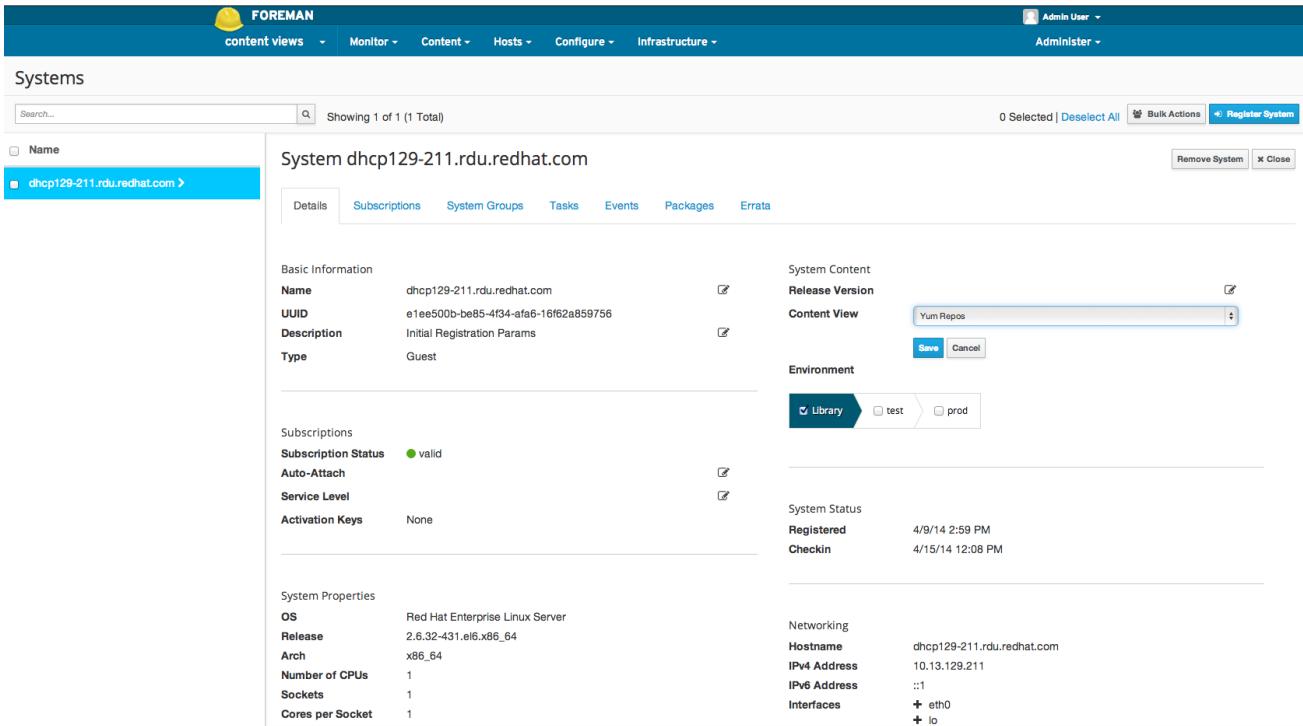
To register a Content Host that is not currently registered to the Content View, simply use subscription manager on the client Content Host and run:

```
subscription-manager register --org=ACME_Corporation --environment=Library/my_rhel_view
```

This would register the Content Host to the Library environment and the my_rhel_view Content View.

If the Content Host is already registered, from the UI:

Hosts > Content Hosts > Select the desired Content Host



Systems

Showing 1 of 1 (1 Total)

0 Selected | Deselect All | Bulk Actions | Register System | Remove System | Close

Name: dhcp129-211.rdu.redhat.com

Basic Information

Name	dhcp129-211.rdu.redhat.com
UUID	e1ee500b-be85-4f34-afaf-16f62a859756
Description	Initial Registration Params
Type	Guest

System Content

Release Version: Yum Repos

Content View: Library (selected)

Environment: Library (selected)

System Status

Registered	4/9/14 2:59 PM
Checkin	4/15/14 12:08 PM

System Properties

OS	Red Hat Enterprise Linux Server
Release	2.6.32-431.el6.x86_64
Arch	x86_64
Number of CPUs	1
Sockets	1
Cores per Socket	1

Networking

Hostname	dhcp129-211.rdu.redhat.com
IPv4 Address	10.13.129.211
IPv6 Address	::1
Interfaces	+ eth0 + lo

From the CLI:

```
hammer content-host update \
--organization="Default Organization" \
--name="dhcp129-211.rdu.redhat.com" \
--content-view="New Content View" \
--lifecycle-environment="Library"
```

Promoting a Content View

Initially a Content View is published to Library as version 1. If there are Content Hosts in other environments that would like to consume this Content View, a version of the content view will need to be promoted to those environments. For example, given the Content View "New Content View", version 1 of which has been promoted to the Dev environment. Any Content Hosts in Dev attached to the Content View would remain at version 1 until a version 2 is both published and promoted to the Dev environment.

To promote a Content View in the Web UI, navigate to:

Content > Content Views > Select the desired Content View > Versions (within sub navigation) > Click promote for desired version

FOREMAN

content views Monitor Content Hosts Configure Infrastructure

Admin User Administer

Content Views

Search... Showing 4 of 4 (4 Total) 0 Selected | Deselect All + Create New View

Name
 COMPOSITE
 New Content View >
 Puppet Modules
 Yum Repos

New Content View

Versions Content Puppet Modules History Details Tasks

Filter

Version	Status	Environments	Content	Author	Actions
Version 1	Published. (4/15/14 3:57 PM)	• Library	487 Packages 112 Errata (▲ 44 ▲ 13 🔍 55)		<input type="button" value="Promote"/>

FOREMAN

content views Monitor Content Hosts Configure Infrastructure

Admin User Administer

Content Views

Search... Showing 4 of 4 (4 Total) 0 Selected | Deselect All + Create New View

Name
 COMPOSITE
 New Content View >
 Puppet Modules
 Yum Repos

New Content View

Versions Content Puppet Modules History Details Tasks

[« Back to Version List](#)

Promote Version 1

Choose one or more lifecycle environments from the existing promotion paths available in 'walden'.

Library test prod

New Content View

Version	Status	Environments	Content	Author	Actions
Version 1	Promoting to 1 environment.	• Library • test	0 Packages 0 Errata (▲ 0 ⚡ 0 🔍 0)		Promote

To promote a Content View in the CLI:

```
hammer content-view version promote \
--organization="Default Organization" \
--content-view="New Content View" \
--to-lifecycle-environment="Test" \
--version 1
```

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

1. Installation

1.1 Katello

1.2 Capsule

1.3 Clients

2. Upgrade

2.1 Katello

2.2 Capsule

2.3 Clients

3. Release Notes

4. CLI

5. User Guide

5.1 Activation Keys

5.2 Backup

5.3 Capsules

5.4 Content

5.5 Content Hosts

5.6 Content Views

5.7 Disconnected

5.8 Docker Management

5.9 Email Notifications

5.10 Errata

5.11 Glossary

5.12 GPG Keys

5.13 Host Collections

5.14 Lifecycle Environments

5.15 Provisioning

5.16 Puppet Integration

5.17 Red Hat Content

6. Troubleshooting

Katello Inter-Server Sync

NOTE: This feature is intended to replace the 'katello-disconnected' script.

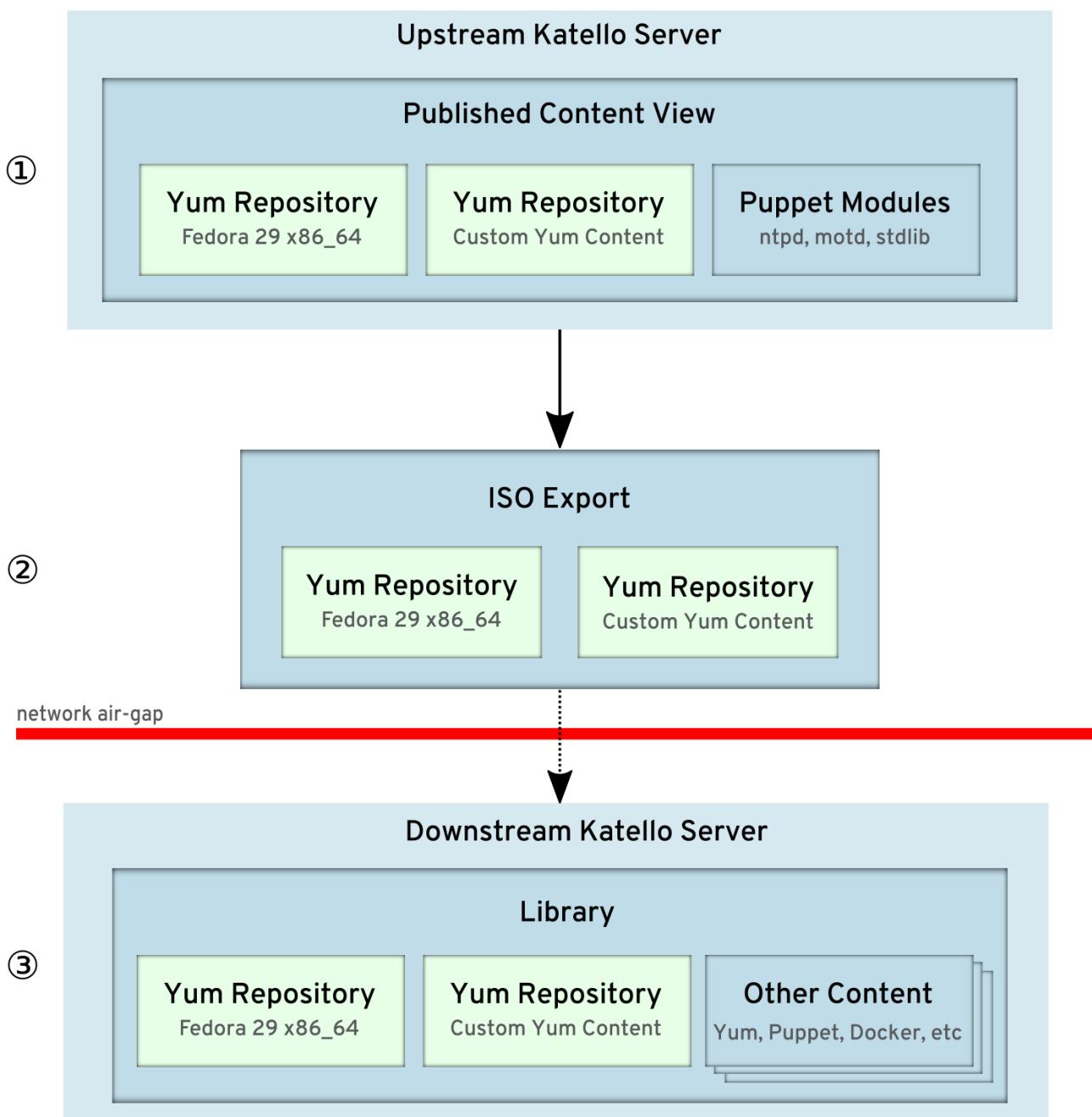
Intro

If you are working in an air-gapped network environment where some of your Katello servers do not have Internet connectivity, you may be interested in using the Katello Inter-Server Sync (ISS) feature. This allows you to export repos, including repos in content views, on your “upstream” Katello server, and then import said repos into your “downstream” server that does not have connectivity. Individual repos can be exported, or all of the repos in a content view.

List of currently supported repo content types:

- yum

Future releases will enable support for additional content types.



The diagram above shows an example scenario where a user wants to export all Yum content in a content view and then import to another Katello server. The ISO file is burned to media and then walked across the air-gap in the network.

Detailed Operation

Exporting

The ISS feature allows users to move Yum content from one Katello server to another, in a way that is compatible with air-gapped networks. Typically users will set up an upstream server that is connected to the Internet, and then create a content view that contains Yum content that they would like to present to the downstream server (step 1).

Content is exported via either `hammer repository export` or `hammer content-view version export` (step 2). It is exported to the location set in “`pulp_export_destination`” in the Settings page, under the Katello tab. This must be set by the user prior to the

first export or else an error will be raised. Please be aware that the location needs to be readable and writable by the `foreman` user.

You can select to either export as a plain set of directories, or as a set of ISO files. The “`iso_size_mb`” parameter sets how large you would like each ISO file to be. It defaults to 4380 MB, which is the size of a single-side, single-layer DVD.

Importing

Importing (step 3) can be done in one of two ways. The first way is to make the export available via HTTP to the importing Satellite instance. Simply put the export in `/var/www/pub/export`. After that, edit your CDN location from the manifest import page to point to “`http://localhost/export/path/to/export`” and the Red Hat Repos page will then work as expected, using your exported data.

The second way is to perform a repository sync via hammer, specifying the source location. Please see the `hammer repository sync` command for more information. This method is the only way to import custom content, and is the only way to import incremental content.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Docker Management

Katello can be used to manage and deploy [Docker](#) content. Katello can retrieve Docker content from a variety of sources such as Docker hub, private Docker registries, the Red Hat CDN, and so forth. Docker content can then be published and promoted via [Content Views](#) and then pulled or provisioned to a server running Docker.

What is Docker?

Docker is a tool used to manage Linux containers. To read more about Docker, [check out the official Docker site](#). Docker repositories, which contain images and tags, can be retrieved, stored, managed, and deployed from Katello.

How to sync a Docker repository

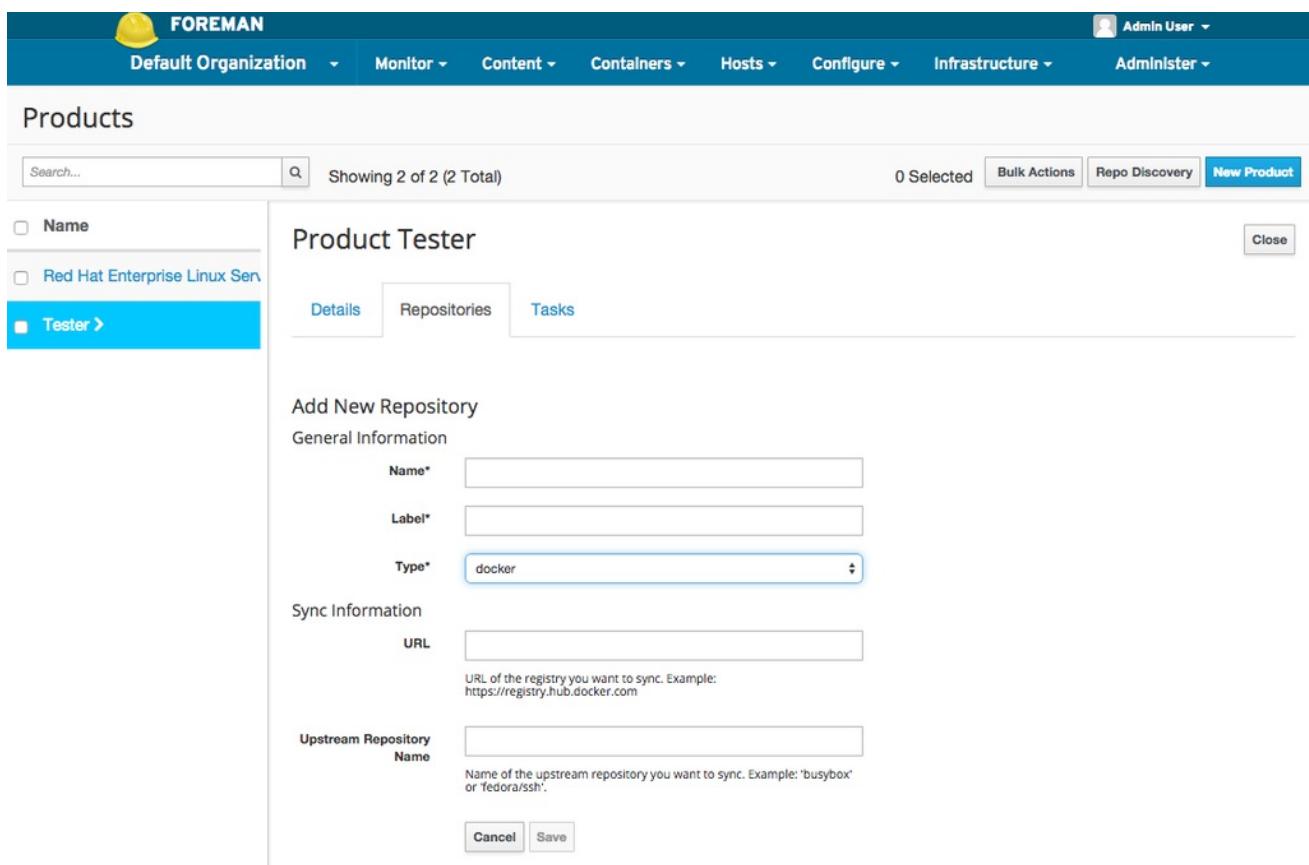
The easiest way to get Docker content into Katello is to sync it in. You can either sync Docker content from the Red Hat CDN (if you have subscriptions for the content) or from a registry such as Docker Hub.

Red Hat Docker Images

Content can be synced into Katello using a Red Hat manifest in much the same way as yum content. See [our guide on how to manage Red Hat content for more information](#).

Docker Hub/Docker Registry

To sync content from a Docker registry such as Docker Hub (which is the official Docker-run registry), simply start by creating a new Repository.



The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, and Administer. On the left, a sidebar lists 'Name', 'Red Hat Enterprise Linux Ser.', and 'Tester >' (which is highlighted). The main area is titled 'Product Tester' and contains a sub-titled 'Add New Repository'. It has three tabs: 'Details' (selected), 'Repositories', and 'Tasks'. Under 'General Information', there are fields for 'Name*' (empty), 'Label*' (empty), and 'Type*' (set to 'docker'). Under 'Sync Information', there's a 'URL' field containing 'https://registry.hub.docker.com' with a note below it: 'URL of the registry you want to sync. Example: https://registry.hub.docker.com'. Under 'Upstream Repository Name', there's a field containing 'busybox' with a note: 'Name of the upstream repository you want to sync. Example: "busybox" or "fedora/ssh"'. At the bottom are 'Cancel' and 'Save' buttons.

On the new Repository screen, select “Docker” as the content type. Once you do that, you’ll be given two options: upstream name and URL. The URL will be the registry URL; for Docker Hub, this would be <https://registry.hub.docker.com> .

For the upstream name, you want to use the fully qualified upstream name which also includes any namespace such as the username. This can be just “busybox” if the Repository is an official Docker Hub Repository or it can be something like “fedora/ssh” where “fedora” is the username/namespace.

Then click save and then sync the Repository as you normally would. Katello will fetch all the images and tags contained within that Repository.

How to Upload Docker Images

In versions of Katello prior to 3.0, Docker images could be uploaded directly via either the UI or CLI. However, Katello 3.0 only supports the Docker Registry v2 format, which is significantly different than the Docker Registry v1 format. The `docker save` command outputs a Docker image in v1 format, which cannot be uploaded directly to a v2 repository.

As a workaround, you can create a local Docker registry like so:

```
docker run -p 5030:5000 --name registry registry:2
```

Note the `:2` above, which specifies a v2 registry. Push your changes to your newly created local registry then follow the instructions in the section above to sync this registry to Katello. This will ensure that your Docker content stays in Docker's v2 registry format.

How to Publish and Promote Docker Content

Docker content can be published and promoted via Content Views much like yum or puppet content.

The screenshot shows the Foreman interface with the 'Content Views' page selected. At the top, there is a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, and Administer. On the right, there is a user profile icon and a dropdown menu for 'Admin User'. Below the navigation bar, the page title is 'Content Views'. There is a search bar and a button for 'Create New View'. A table lists content views: 'Test' (selected, highlighted in blue) and 'wat'. The 'Docker Content' tab is active. Below the tabs, there is a 'Repository Selection' section with buttons for 'List/Remove' and 'Add'. A table lists repositories: business, busybox, redis, tester, and wat. Each row includes a checkbox, the repository name, product (Tester), last sync time, sync state (Success), and content count (10 Docker Images, 5 Docker Tags for busybox; 179 Docker Images, 19 Docker Tags for redis; 12 Docker Images, 1 Docker Tag for wat).

Name	Product	Last Sync	Sync State	Content
business	Tester	N/A	N/A	10 Docker Images 5 Docker Tags
busybox	Tester	2/2/15 1:18 PM	Success	10 Docker Images 5 Docker Tags
redis	Tester	1/19/15 5:10 PM	Success	179 Docker Images 19 Docker Tags
tester	Tester	2/17/15 3:40 PM	Success	12 Docker Images 1 Docker Tags
wat	Tester			

After creating a Content View, visit the Docker Content tab. Here you can select any Docker repositories you want to add to your Content View. After you've added Docker Repositories to your view, you may proceed as normal. Visit [the Content View user guide for more information](#).

How to View and Pull Docker Content

To view Docker content contained with Katello, visit the Docker Tags page. This can be accessed under the Content menu at the top of any page.

Docker Tags



Showing 25 of 25 (25 Total)

Name	Product Name	Repository Name
2.8.14	Tester	redis
2.8.15	Tester	redis
2.8.16	Tester	redis
2.8.17	Tester	redis
2.8.18	Tester	redis
2.8.19	Tester	redis
2.8.6	Tester	redis
2.8.7	Tester	redis
2.8.8	Tester	redis
2.8.9	Tester	redis
buildroot-2013.08.1	Tester	busybox
buildroot-2014.02	Tester	busybox
latest	Tester	tester
latest	Tester	redis
latest	Tester	busybox
ubuntu-12.04	Tester	busybox
ubuntu-14.04	Tester	busybox



On the Docker Tags page, you can see a list of Docker Tags grouped by Repository in Katello. This shows you Tags grouped across Content Views and Lifecycle Environments. Suppose I wanted to pull the latest Tag from my redis repository, I would click the latest row for my redis repository.


FOREMAN

[Default Organization](#) ▾ [Monitor](#) ▾ [Content](#) ▾ [Containers](#) ▾ [Hosts](#) ▾ [Configure](#) ▾ [Infrastructure](#) ▾ [Administer](#) ▾

Admin User

Docker Tags

🔍 Showing 25 of 25 (25 Total)

Name	redis:latest			✖ Close
redis:2.8.14				
redis:2.8.15				
redis:2.8.16				
redis:2.8.17				
redis:2.8.18				
redis:2.8.19				
redis:2.8.6				
redis:2.8.7				
redis:2.8.8				
redis:2.8.9				
busybox:buildroot-2013.08.1				
busybox:buildroot-2014.02				
tester:latest				
redis:latest ➔				
busybox:latest				
busybox:ubuntu-12.04				
busybox:ubuntu-14.04				

I can see here that my redis Repository has been added to a published Content View called redisv. If I want to use the tag from that Content View, I would just copy the Published At URL and then on my docker server I would run:

```
$ docker pull localhost:5000/default_organization-library-redisv-Tester-redis:latest
Pulling repository localhost:5000/default_organization-library-redisv-Tester-redis...
```

How to Provision Docker Content

See how to provision content in the documentation in the [foreman-docker documentation](#). Provisioning content from Katello works in much the same way.

First, proceed to the new Container page by accessing it from the Containers menu at the top. Then, select the Local Content tab on the second step. This will allow you to select a Docker image from a published Katello repository which is in an environment/content view/capsule. Then just proceed in the wizard as per the Foreman Docker instructions. When you are finished, you should have a new container running from an image in Katello.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Email Notifications

Types of Email Notifications

In addition to the Email Notifications that Foreman provides:

- Puppet run summary (Daily/Weekly/Monthly)
- Puppet errors

Katello provides a few addition reports:

- Katello Host Advisory (Daily/Weekly/Monthly) - A report of all of the Errata applicable to all readable Content Hosts
- Katello Promote Errata - A report generated at Content View promotion time showing what Errata applicable to the Content Hosts within that Content View.
- Katello Sync Errata - A report generated after each Repository sync listing new Errata synced and how many Content Hosts are applicable.

Configuring the Foreman/Katello to send emails:

The configuration on how the Foreman/Katello service sends email is located in /etc/foreman/email.yaml:

```
production:
  delivery_method: :smtp
  smtp_settings:
    address: smtp.example.com
    port: 25
    domain: example.com
    authentication: :none
```

For more information see: [Email Configuration](#)

Opting in to the emails

By default a user will receive no email notifications. Each notification must be opted into.

To opt in for your own user, at the very top right of the web interface, hover over your Username, click "My Account" and then click the "Mail Preferences" tab.

To opt in for other users, navigate to "Administer" > "Users" > Click the desired User > click the "Mail Preferences" tab.

Select which emails and frequency you would like the user to have and click "Submit".

The screenshot shows the Foreman web interface with the 'Edit User' page open. The 'Mail Preferences' tab is active. The page includes sections for 'General' and 'Notifications'. Under 'Notifications', there are five categories: 'Katello host advisory' (Weekly), 'Katello promote errata' (Subscribe), 'Katello sync errata' (Subscribe), 'Puppet error state' (No emails), and 'Puppet summary' (Weekly). Each category has a description and a dropdown menu for selecting the frequency of emails. At the bottom of the form are 'Cancel' and 'Submit' buttons.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

1. Installation

[1.1 Katello](#)[1.2 Capsule](#)[1.3 Clients](#)

2. Upgrade

[2.1 Katello](#)[2.2 Capsule](#)[2.3 Clients](#)

3. Release Notes

4. CLI

5. User Guide

[5.1 Activation Keys](#)[5.2 Backup](#)[5.3 Capsules](#)[5.4 Content](#)[5.5 Content Hosts](#)[5.6 Content Views](#)[5.7 Disconnected](#)[5.8 Docker Management](#)[5.9 Email Notifications](#)[5.10 Errata](#)[5.11 Glossary](#)[5.12 GPG Keys](#)[5.13 Host Collections](#)[5.14 Lifecycle Environments](#)[5.15 Provisioning](#)[5.16 Puppet Integration](#)[5.17 Red Hat Content](#)

6. Troubleshooting

Errata

Errata are updates between major releases. An Erratum is metadata about a group of packages that explains the importance of the package updates. Errata may be released individually on an as-needed basis or aggregated as a minor release. There are three main types of errata:

- Enhancement: the new packages contain one or more added features

- Bugfix: the new packages contain one or more bug fixes
- Security: the new packages fix one or more security vulnerabilities

With regard to Content Hosts, Errata is divided into two distinct classifications depending on whether or not the Errata is present in the Content Host's Lifecycle Environment and Content View:

- Applicable: the errata applies to one or more Content Hosts
- Installable: the errata applies to one or more Content Hosts and is present in the Content Host's Lifecycle Environment and Content View

Definitions

- [Content Host](#)
- [Content View](#)
- [Lifecycle Environment](#)

General Features

The following is a high-level summary of the Errata features:

- [View List of Errata](#)
- [View Errata Details](#)
- [View Affected Content Hosts](#)
- [View Repositories Containing Errata](#)
- [Applying Errata](#)

View List of Errata

To view the list of Errata in the Organization:

- navigate to: Content > Errata

Errata				
Search... <input type="text"/>		Type	Affected Content Hosts	Updated
<input type="checkbox"/> RHSAs	Errata ID	Title	Type	Affected Content Hosts
<input type="checkbox"/> RHSA-2014:1984		Important: bind security update	▲ Security Advisory - Important	0
<input type="checkbox"/> RHSA-2014:1983		Important: xorg-x11-server security update	▲ Security Advisory - Important	0
<input type="checkbox"/> RHBA-2014:1970		dovecot bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHSA-2014:1974		Important: rpm security update	▲ Security Advisory - Important	0
<input type="checkbox"/> RHBA-2014:1964		selinux-policy bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHBA-2014:1965		curl bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHBA-2014:1967		sssd bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHBA-2014:1961		system-config-firewall bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHSA-2014:1919		Critical: firefox security update	▲ Security Advisory - Critical	0
<input checked="" type="checkbox"/> RHSA-2014:1948		Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update	▲ Security Advisory - Important	0
<input type="checkbox"/> RHEA-2014:1918		oprofile enhancement update	▣ Product Enhancement Advisory	0
<input type="checkbox"/> RHBA-2014:1909		lvm2 bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHSA-2014:1911		Moderate: ruby security update	▲ Security Advisory - Moderate	0
<input type="checkbox"/> RHBA-2014:1886		mdadm bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHBA-2014:1884		cyrus-sasl bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHBA-2014:1883		webkitgtk and gimp bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHBA-2014:1875		device-mapper-multipath bug fix update	※ Bug Fix Advisory	1
<input type="checkbox"/> RHBA-2014:1867		nss-softokn bug fix update	※ Bug Fix Advisory	0
<input type="checkbox"/> RHEA-2014:1869		new packages: kmod-oracleasm	▣ Product Enhancement Advisory	0

View Errata Details

To view the details of an Errata:

- navigate to: Content > Errata
- Click on an Errata ID

FOREMAN

Default Organization ▾ Monitor ▾ Content ▾ Containers ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Errata

Search... Showing 40 of 2809 (2809 Total) 0 Selected + Apply Errata

Errata ID
 RHSA-2014:1984
 RHSA-2014:1983
 RHBA-2014:1970
 RHSA-2014:1974
 RHBA-2014:1964
 RHBA-2014:1965
 RHBA-2014:1967
 RHBA-2014:1961
 RHSA-2014:1919
 RHSA-2014:1948
 RHEA-2014:1918
 RHBA-2014:1909
 RHSA-2014:1911
 RHBA-2014:1886
 RHBA-2014:1884
 RHBA-2014:1883
 RHBA-2014:1875
 RHBA-2014:1867
 RHEA-2014:1869
 RHSA-2014:1870

Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update

Details Content Hosts Repositories

Advisory RHSA-2014:1948
CVEs
Type Security Advisory
Severity Important
Issued 12/1/14
Last Updated On 12/1/14
Reboot Suggested? No

Topic
 Updated nss, nss-util, and nss-softokn packages that contain a patch to mitigate the CVE-2014-3566 issue, fix a number of bugs, and add various enhancements are now available for Red Hat Enterprise Linux 5, 6, and 7.

Red Hat Product Security has rated this update as having Important security impact.

Description
 Network Security Services (NSS) is a set of libraries designed to support the cross-platform development of security-enabled client and server applications. Netscape Portable Runtime (NSPR) provides platform independence for non-GUI operating system facilities.

This update adds support for the TLS Fallback Signaling Cipher Suite Value (TLS_FALLBACK_SCSV), which can be used to prevent protocol downgrade attacks against applications which re-connect using a lower SSL/TLS protocol version when the initial connection indicating the highest supported protocol version fails.

This can prevent a forceful downgrade of the communication to SSL 3.0. The SSL 3.0 protocol was found to be vulnerable to the padding oracle attack when using block cipher suites in cipher block chaining (CBC) mode. This issue is identified as CVE-2014-3566, and also known under the alias POODLE. This SSL 3.0 protocol flaw will not be addressed in a future update; it is recommended that users configure their applications to require at least TLS protocol version 1.0 for secure communication.

View Affected Content Hosts

To view the Affected Content Hosts of an Errata:

- navigate to: Content > Errata
- Click on an Errata ID
- Click on the Content Hosts Tab

Note the following option:

- Checking the box limits the display of Content Hosts to those which already have the Errata available in their Lifecycle Environment and Content View.

FOREMAN

Default Organization ▾ Monitor ▾ Content ▾ Containers ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Errata

Search... Showing 40 of 2809 (2809 Total) 0 Selected + Apply Errata

Errata ID
 RHSA-2014:1984
 RHSA-2014:1983
 RHBA-2014:1970
 RHSA-2014:1974
 RHBA-2014:1964
 RHBA-2014:1965
 RHBA-2014:1967
 RHBA-2014:1961
 RHSA-2014:1919
 RHSA-2014:1948
 RHEA-2014:1918
 RHBA-2014:1909
 RHSA-2014:1911
 RHBA-2014:1886
 RHBA-2014:1884
 RHBA-2014:1883
 RHBA-2014:1875
 RHBA-2014:1867
 RHEA-2014:1869
 RHSA-2014:1870

Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update

Details Content Hosts Repositories

Apply To Content Hosts
 Only show content hosts where Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update is currently available in the host's Lifecycle Environment.
 Filter by Environment
 Search... Showing 1 of 1 (1 Total) 0 Selected + Apply to Hosts

Name	OS	Environment	Content View
walden-rhel7.rdu.redhat.com	Red Hat Enterprise Linux Server 7.0	Library	CV

View Repositories Containing Errata

To view the Repositories Containing an Errata:

- navigate to: Content > Errata
- Click on an Errata ID
- Click on the Repositories Tab

Note that you can filter by Lifecycle Environment and Content View.

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, Admin User, and Administer. Below the navigation bar, the page title is "Errata". A search bar and a "Showing 40 of 2809 (2809 Total)" message are visible. On the left, a sidebar lists various errata IDs with checkboxes. One specific erratum, "RHSA-2014:1948", is highlighted with a blue background and a right-pointing arrow icon. The main content area has a heading "Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update". It features three tabs: Details, Content Hosts, and **Repositories**. The "Repositories" tab is selected. Below it, a sub-section titled "Repositories Containing Errata RHSA-2014:1948" shows a table with one row. The table has columns for Name, Product, Content View, and Last Sync. The single entry is "RHEL6" under "Name" and "Red Hat" under "Product". The "Content View" and "Last Sync" columns are empty or show "Success 20 days ago".

Applying Errata

How Errata is applied to a Content Host(s) depends on whether the Errata is installable.

- If the Errata is already installable then the Errata is applied to the Content Host(s).
- If the Errata is not installable then an Incremental Update is generated. An Incremental Update creates a point release of the Content View with the Errata included. The Errata can also be applied to the Content Host(s) as part of this process.

There are two ways to apply Errata:

- A single Errata can be applied to one or more Content Hosts
- Several Errata can be applied to one or more Content Hosts via a bulk operation

Applying a Single Errata

To apply a single Errata:

- Navigate to: Content > Errata
- Click on an Errata ID
- Click on the Content Host tab
- Select the desired Content Hosts
- Click "Apply to Hosts"
- Confirm the action

FOREMAN

Default Organization ▾ Monitor ▾ Content ▾ Containers ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Errata

All Repositories Applicable Installable
 Search... Showing 19 of 19 (19 Total) 1 Selected

Errata ID
<input checked="" type="checkbox"/> RHSA-2014:1948 >
<input type="checkbox"/> RHBA-2014:1875
<input type="checkbox"/> RHEA-2014:1866
<input type="checkbox"/> RHEA-2014:1733
<input type="checkbox"/> RHBA-2014:1637
<input type="checkbox"/> RHBA-2014:1515
<input type="checkbox"/> RHBA-2014:1426
<input type="checkbox"/> RHBA-2014:1378
<input type="checkbox"/> RHBA-2014:1376
<input type="checkbox"/> RHEA-2014:1472
<input type="checkbox"/> RHEA-2014:1532
<input type="checkbox"/> RHEA-2014:1530
<input type="checkbox"/> RHBA-2014:1384
<input type="checkbox"/> RHBA-2014:1555
<input type="checkbox"/> RHBA-2014:1459
<input type="checkbox"/> RHBA-2014:1590
<input type="checkbox"/> RHEA-2014:1500
<input type="checkbox"/> RHBA-2014:1362

Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update

Details

Apply To Content Hosts

Only show content hosts where Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update is currently available in the host's Lifecycle Environment.

Filter by Environment
 Search... Showing 1 of 1 (1 Total) 1 Selected

Name	OS	Environment	Content View
<input checked="" type="checkbox"/> walden-rhel7.rdu.redhat.com	Red Hat Enterprise Linux Server 7.0	dev	RHEL

FOREMAN

Default Organization ▾ Monitor ▾ Content ▾ Containers ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Errata

All Repositories Applicable Installable
 Search... Showing 19 of 19 (19 Total) 1 Selected

Errata ID
<input checked="" type="checkbox"/> RHSA-2014:1948 >
<input type="checkbox"/> RHBA-2014:1875
<input type="checkbox"/> RHEA-2014:1866
<input type="checkbox"/> RHEA-2014:1733
<input type="checkbox"/> RHBA-2014:1637
<input type="checkbox"/> RHBA-2014:1515
<input type="checkbox"/> RHBA-2014:1426
<input type="checkbox"/> RHBA-2014:1376
<input type="checkbox"/> RHEA-2014:1472
<input type="checkbox"/> RHEA-2014:1532
<input type="checkbox"/> RHBA-2014:1384
<input type="checkbox"/> RHBA-2014:1555
<input type="checkbox"/> RHBA-2014:1459
<input type="checkbox"/> RHBA-2014:1590
<input type="checkbox"/> RHEA-2014:1500
<input type="checkbox"/> RHBA-2014:1362

Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update

Details

Errata List > Select Content Hosts > Confirm

Apply RHSA-2014:1948

These Errata are not Installable via your published Content View versions running on the selected hosts. The new Content View Versions (specified below) will be created which will make this Errata Installable in the host's Environment. This new version will replace the current version in your host's Lifecycle Environment. To install these errata immediately on hosts after publishing check the box below.

Content View	Version	Environment	Host Count
RHEL	2.1	dev	1

Apply Errata to Content Hosts immediately after publishing.

Applying Several Errata

To apply several Errata:

- Navigate to: Content > Errata
- Select the desired Errata
- Click "Apply Errata"
- Select the intended Content Hosts
- Click "Next"
- Confirm the action

Errata

All Repositories Applicable Installable
 Showing 19 of 19 (19 Total) 1 Selected

<input type="checkbox"/> Errata ID	Title	Type	Content Host Counts	Updated
<input checked="" type="checkbox"/> RHSA-2014:1948 ➤	Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update	 Security Advisory - Important	1 Applicable, 0 Installable	12/1/14
<input type="checkbox"/> RHBA-2014:1875	device-mapper-multipath bug fix update	 Bug Fix Advisory	1 Applicable, 1 Installable	11/18/14
<input type="checkbox"/> RHEA-2014:1866	tzdata enhancement update	 Product Enhancement Advisory	1 Applicable, 1 Installable	11/16/14
<input type="checkbox"/> RHEA-2014:1733	tzdata enhancement update	 Product Enhancement Advisory	1 Applicable, 1 Installable	10/28/14
<input type="checkbox"/> RHBA-2014:1637	at bug fix update	 Bug Fix Advisory	1 Applicable, 1 Installable	10/14/14
<input type="checkbox"/> RHBA-2014:1515	audit bug fix and enhancement update	 Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14
<input type="checkbox"/> RHBA-2014:1426	openldap bug fix and enhancement update	 Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14
<input type="checkbox"/> RHBA-2014:1378	nss bugfix and enhancement update	 Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14
<input type="checkbox"/> RHBA-2014:1376	xcb-util, xorg-x11-drivers, and mesa bug fix and enhancement update	 Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14
<input type="checkbox"/> RHEA-2014:1472	elfutils bug fix and enhancement update	 Product Enhancement Advisory	1 Applicable, 1 Installable	10/13/14
<input type="checkbox"/> RHEA-2014:1532	libnl3 enhancement update	 Product Enhancement Advisory	1 Applicable, 1 Installable	10/13/14
<input type="checkbox"/> RHEA-2014:1530	new packages: json-c	 Product Enhancement Advisory	1 Applicable, 1 Installable	10/13/14
<input type="checkbox"/> RHBA-2014:1384	subscription-manager bug fix and	 Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14

All Repositories Applicable Installable
 Showing 19 of 19 (19 Total) 1 Selected

Errata ID

- RHSA-2014:1948
- RHBA-2014:1875
- RHEA-2014:1866
- RHEA-2014:1733
- RHBA-2014:1637
- RHBA-2014:1515
- RHBA-2014:1426
- RHBA-2014:1378
- RHBA-2014:1376
- RHEA-2014:1472
- RHEA-2014:1532
- RHEA-2014:1530
- RHBA-2014:1384
- RHBA-2014:1555
- RHBA-2014:1459
- RHBA-2014:1590
- RHEA-2014:1500
- RHBA-2014:1362

Apply Errata

Errata List ➤ Select Content Hosts

Only show content hosts where is currently available in the host's Lifecycle Environment.

Filter by Environment

Search... Showing 1 of 1 (1 Total) 1 Selected

<input type="checkbox"/> Name	OS	Environment	Content View
<input checked="" type="checkbox"/> walden-rhel7.rdu.redhat.com	Red Hat Enterprise Linux Server 7.0	dev	RHEL

Errata

All Repositories Applicable Installable
Search... Showing 19 of 19 (19 Total) 1 Selected

Content View	Version	Environment	Host Count
RHEL	2.1	dev	1

Apply Errata to Content Hosts immediately after publishing.

Foreman 2.3.3 has been released! Follow the quick start to install it.

Foreman 2.2.2 has been released! Follow the quick start to install it.

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Glossary

The following terms are used throughout this document, and are important for the users understanding of Katello.

Activation Key

A registration token which can be used in a kickstart file to control actions at registration. These are similar to Activation Keys in Spacewalk, but they provide a subset of features because after registration, Puppet takes control of package and

configuration management.

Application Lifecycle Environment

Steps in a promotion path through the Software (Development) Life Cycle (SDLC). Content (packages, puppet modules) can be moved through lifecycle environments via content view publishing/promotion. Traditionally these environments are things like Development -> Test -> Production. Channel cloning was used to implement this concept for this in Spacewalk.

Attach

Associating a Subscription to a Host which provides access to RPM content.

Capsule

An additional “server” that can be used in a Katello deployment to facilitate content federation and distribution in addition to other localized services (Puppet master, DHCP, DNS, TFTP, and more).

Change Set

Set of packages and puppet modules which are promoted between Application Lifecycle Environments. Katello records the progress of changesets as they promoted. Katello also provides audit capabilities to review how environments have changed over time.

Compute Profile

Default attributes for new virtual machines on a compute resource.

Compute Resource

A virtual fabric, or cloud infrastructure, where hosts can be deployed by Katello. Examples include RHEV-M, OpenStack, EC2, and VMWare.

Content

Software packages (RPMS), Package Groups, Errata, and Puppet modules. These are synced into the Library and then promoted into Lifecycle Environments via Content Views in order to be used/consumed by Hosts.

Content Delivery Network (CDN)

The mechanism to deliver Red Hat content in a geographically co-located fashion. For example, content which is synced by a Katello in Europe will pull content from a source in Europe.

Content View

A definition of content that combines products, packages, errata and Puppet modules, with capabilities for intelligent filtering and snapshotting. Content Views are a refinement of the combination of channels and cloning from Spacewalk.

External Node Classifier

A Puppet construct that provides additional data for a Puppet master to be used for configuring Hosts. Foreman acts as an External Node Classifier to Puppet Masters in a Satellite deployment.

Factor

A program that provides information (facts) about the system on which it is run (eg: total memory, operating system version, architecture, etc.) Factor facts can be used in Puppet modules in order to enable specific configurations based on Host data.

Hammer

The command line tool for Katello. Hammer can be used as a standard cli (and used in scripts) and can also be used as a shell in the same way that spacecmd, virsh and others work.

Host

A system, either physical or virtual, which is managed by Katello.

Host Group

A template for how a Host should be built. This includes the content view (which defines the available RPMS and Puppet modules), and the Puppet classes to apply (which determines the ultimate software and configuration).

Location

A collection of default settings which represent a physical place. These can be nested so that a user can set up defaults, for example, for Europe, which are refined by Tel Aviv, which are refined by DataCenter East, and then finally by Rack 22.

Library

The Library is the single origin of all content which can be used. If you are an Information Technology Infrastructure Library (ITIL) shop, it is your definitive media library.

Manifest

The means of transferring subscriptions from a Subscription Provider (such as the Red Hat Customer portal) to Katello. This is similar in function to certificates used with Spacewalk.

Organization

A tenant in Katello. Organizations, or orgs, are isolated collections of hosts, content and other functionality within a Katello deployment.

Permission

The ability to perform an action.

Product

A collection of content repositories.

Promote

The act of moving content from one Application Lifecycle Environment to another.

Provisioning Template

User defined templates for Kickstarts, snippets and other provisioning actions. These provide similar functionality to Kickstart Profiles and Snippets in Katello.

Puppet Agent

An agent that runs on a Host that applies configuration changes to that Host.

Puppet Class

A Puppet Class is re-usable named block of puppet manifest, similar to a class in an object-oriented programming language. Puppet classes must be included/instantiated in order to use their functionality. Puppet Classes can be parameterized - they can take parameters when they are included/instantiated and those parameters may be used by the underlying manifest to affect the ultimate configuration.

Puppet Manifest

A Manifest is a simple set of Puppet instructions. Manifests typically have the .pp extension. A manifest is much like a procedure in programming terms.

Puppet Master

A Capsule component that provides Puppet manifests to Hosts for execution by the Puppet Agent.

Puppet Module

A Puppet Module is a set of Puppet manifests/classes, template files, tests and other components packaged together in a specific directory format. Puppet Modules are typically associated with specific software (eg: NTP, Apache, etc) and contain various classes used to assist in the installation and configuration of that software. Puppet Labs maintains a repository of official and user-contributed modules called the Puppet Forge.

Pulp Node

A Capsule component that mirrors content. This is similar to the Spacewalk Proxy in Spacewalk. The main difference is that content can be pre-staged on the Pulp Node before it is used by a Host.

Repository

A collection of content (yum repository, puppet repository).

Role

A collection of permissions that are applied to a set of resources (such as Hosts).

Smart Proxy

A Capsule component that can integrate with external services, such as DNS or DHCP.

Smart Variable

A configuration value that controls how a Puppet Class behaves. This can be set on a Host, a Host Group, an Organization, or a Location.

Standard Operating Environment (SOE)

A controlled version of the operating system on which applications are deployed.

Subscription

The right to receive content and service from Red Hat. This is purchased by customers.

Syncing

Mirroring content from external resources into an organization's Library.

Sync Plans

Scheduled execution of syncing content.

Usergroup

A collection of roles which can be assigned to a collection of users. This is similar to the Role in Spacewalk.

User

A human who works in Katello. Authentication and authorization can be done via built in logic, or using external LDAP or kerberos resources.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Managing Content Hosts using GPG Keys

GPG Keys provide a way to verify the integrity of packages found within a Repository and/or Product. Once one or more GPG Keys have been created, they can be associated with a Product or Repository during creation or by editing an existing Product or Repository.

For more information on GPG Keys see [The GNU Privacy Guard](#).

General Features

The following is a high-level summary of GPG Key features:

- [Create a GPG Key](#)
- [View Associated Products](#)
- [View Associated Repositories](#)
- [Associate GPG Key with Product](#)

Create a GPG Key

To create a new GPG Key:

- navigate to: Content > GPG Keys
- click **New GPG Key**

Note that you may either upload your GPG Key or simply paste in the contents.

The screenshot shows the Foreman web interface with the URL http://wraines-sandbox.rdu.redhat.com/gpg_keys/new. The page title is "Gpg Keys". A search bar shows "Showing 1 of 1 (1 Total)". On the right, there is a blue button labeled "+ New GPG Key". The main content area is titled "New GPG Key" and contains a form. The "Name*" field has "Another New GPG Key" entered. Below it, there are two options: "Upload GPG Key" (with a "Choose File" button) and "Paste GPG Key Contents". The "Paste GPG Key Contents" option is selected, and its value is a long string of GPG key data starting with "-----BEGIN PGP PUBLIC KEY BLOCK-----". At the bottom of the dialog are "Cancel" and "Save" buttons, with "Save" being highlighted in blue.

View Associated Products

To view all Products that have been assigned a GPG Key:

- navigate to: Content > GPG Keys
- select the desired GPG Key from the list
- click **Products**

The screenshot shows the Foreman web interface for managing GPG keys. The URL is wraines-sandbox.rdu.redhat.com/gpg_keys/3/products. The top navigation bar includes links for Default Organization, Monitor, Content, Hosts, Configure, Infrastructure, Admin User (logged in as Admin User), and Administrator.

The main page title is "Gpg Keys". A search bar and a "Showing 1 of 1 (1 Total)" message are at the top. On the right, there are buttons for "New GPG Key" and "Remove GPG Key".

A sidebar on the left lists "Name" and "My New GPG Key". The main content area is titled "GPG Key My New GPG Key". It has tabs for "Details", "Products" (which is selected), and "Repositories".

The "Products" section shows a table with two rows:

Name	Repositories
Super Awesome Product	2
Meh Product	0

View Associated Repositories

To view all Repositories that have been assigned a GPG Key:

- navigate to: Content > GPG Keys
- select the desired GPG Key from the list
- click **Repositories**

The screenshot shows the Foreman web interface. At the top, there's a header with the Foreman logo, navigation links for 'Default Organization', 'Monitor', 'Content', 'Hosts', 'Configure', and 'Infrastructure', and user authentication dropdowns for 'Admin User' and 'Administrator'. Below the header, the main content area has a title 'Gpg Keys' and a search bar. A message indicates 'Showing 1 of 1 (1 Total)'. On the left, a sidebar lists 'Name' and 'My New GPG Key'. The main panel is titled 'GPG Key My New GPG Key' and contains tabs for 'Details', 'Products', and 'Repositories'. The 'Repositories' tab is active, showing a table with two rows:

Name	Product	Type
Another Not So Awesome Repository	Super Awesome Product	puppet
Super Awesome Repository	Super Awesome Product	yum

Buttons for 'Remove GPG Key' and 'Close' are located at the top right of the repository list.

Associate GPG Key with Product

To add a GPG to a Product:

Note that adding a GPG Key to a Product adds the GPG Key to all current and future repositories unless a repository already has a GPG Key assigned. This can also be overridden by assigning a GPG Key to an individual repository afterward as well.

- navigate to: Content > Products
- select the desired Product from the list
- click **Details**
- click the edit button on the GPG Key field
- select the desired GPG Key

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Hosts, Configure, and Infrastructure. On the right, it shows 'Admin User' and 'Administrator'. Below the navigation, a search bar and a message indicating 'Showing 2 of 2 (2 Total)' are present. A toolbar at the top right includes 'Bulk Actions', 'Repo Discovery', and 'New Product' buttons.

The main content area is titled 'Product Meh Product'. It has tabs for 'Details', 'Repositories', and 'Tasks', with 'Details' being the active tab. Under 'Basic Information', there are fields for Name ('Meh Product'), Label ('Meh_Product'), and GPG Key ('My New GPG Key'). There are 'Save' and 'Cancel' buttons. To the right, under 'Sync Status', it shows 'Synced manually, no interval set.' with a checkbox. Other sync-related fields include Last Sync (Ago (Local Time)), Next Sync (Synced manually, no interval set.), and Sync State (not_synced). Buttons for 'New Sync Plan' and 'Sync Now' are also present.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Managing Content Hosts using Host Collections

Host Collections provide a mechanism to statically group multiple Content Hosts. This enables administrators to group Content Hosts based on the needs of their organization. For example, Content Hosts could be grouped by function, department or business unit.

Once a Host Collection is created, it can be used to perform various actions on the Content Hosts contained within it. This

includes actions such as the following:

- Package installation, removal and update
- Errata installation
- Changing of assigned Lifecycle Environment or Content View

Definitions

- [Content Host](#)

General Features

The following is a high-level summary of the Host Collection features:

- [Create a Host Collection](#)
- [Add Content Hosts to a Host Collection](#)
- [Copy a Host Collection](#)
- [Perform actions on a Host Collection](#)

Create a Host Collection

To create a new collection,

- navigate to: Hosts > Host Collections
- click **New Host Collection**

Note the following option:

- *Content Host Limit*: This option will control how many Content Hosts are allowed to be added to the collection.

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with the 'FOREMAN' logo, user information ('Admin User'), and links for 'Default_Organization', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', and 'Administer'. Below the navigation bar, the main area is titled 'Host Collections'. On the left, there's a sidebar with a search bar and a 'Name' filter. The main content area has a title 'New Host Collection'. It contains fields for 'Name*' (with 'research' typed in), 'Content Host Limit' (with 'Unlimited Content Hosts' checked), and a 'Description' field. At the bottom are 'Cancel' and 'Save' buttons. The status bar at the bottom indicates '0 Selected | Deselect All' and a '+ New Host Collection' button.

Add Content Hosts to a Host Collection

To add Content Hosts to a collection:

- navigate to: Hosts > Host Collections
- select the desired collection from the list

- click **Content Hosts**
- click **Add**
- select the Content Hosts you would like to add
- click **Add Selected**

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Monitor, Content, Hosts, Configure, Infrastructure, and Administer. On the far right, it shows 'Admin User' and 'Default_Organization'. Below the header, the main content area is titled 'Host Collections'. A search bar and a 'Showing 1 of 1 (1 Total)' message are at the top left. On the right, there are buttons for '0 Selected | Deselect All' and '+ New Host Collection'. A sidebar on the left lists collections, with 'research' highlighted. The main panel shows a 'Host Collection: research' view. It has tabs for Details, Content Hosts (which is selected), and Collection Actions. Under 'Content Hosts', there are buttons for 'List/Remove' and 'Add'. A search bar and a 'Showing 2 of 2 (2 Total) Content Hosts' message are present. The list of hosts includes 'host1.example.com' and 'host2.example.org', each with its environment and content view information.

Name	Environment	Content View
host1.example.com	Library	Default Organization View
host2.example.org	Library	Default Organization View

Copy a Host Collection

Copying a Host Collection allows a user to quickly create a new collection that is a copy of an existing one.

To copy a Host Collection:

- navigate to: Hosts > Host Collections
- select the desired collection from the list
- click **Copy Collection**
- enter a name for the new collection
- click **Create**

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default_Organization, Monitor, Content, Hosts, Configure, Infrastructure, Administer, and a user icon for Admin User. The main area is titled "Host Collections". A search bar at the top left shows "Search..." and a result count of "Showing 1 of 1 (1 Total)". On the right, there are buttons for "0 Selected | Deselect All" and "+ New Host Collection".

A sidebar on the left has a "Name" dropdown with "research" selected. The main content area is titled "Host Collection: research". It has tabs for "Details", "Content Hosts" (which is active), and "Collection Actions". Under "Content Hosts", there are buttons for "List/Remove" and "Add". A search bar shows "Showing 2 of 2 (2 Total) Content Hosts".

A modal window titled "Copy" is open, asking for a new name: "development". It has "Create" and "Cancel" buttons.

Name	Environment	Content View
host1.example.com	Library	Default Organization View
host2.example.org	Library	Default Organization View

Perform Actions on a Host Collection

To perform an action on Content Hosts within a collection:

- navigate to: Hosts > Host Collections
- select the desired collection from the list
- click **Collection Actions**
- click on the action that you would like to perform

 FOREMAN

Default_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Host Collections

Search...  Showing 2 of 2 (2 Total) 0 Selected | Deselect All 

Name
<input type="checkbox"/> development
<input checked="" type="checkbox"/> research ➔

Host Collection: research

Details Content Hosts **Collection Actions**

The following actions can be performed on content hosts in this host collection:

- o [Package Installation, Removal, and Update](#)
- o [Errata Installation](#)
- o [Host Collection Membership](#)
- o Change assigned Environment or Content View

Note: clicking on an action will take the user to the appropriate Content Hosts Bulk Actions page, where all Content Hosts associated with the collection have been selected. [Click here, for more information on performing Content Host Bulk Actions](#)

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](#)

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Lifecycle Environments

What can a Lifecycle Environments be used for?

- Hold content view versions.
- To manage the lifecycle of Content Hosts.
- Establish workflow containers and promote content views.

Definitions

- Lifecycle Environment - containers for content view versions which are consumed by content hosts.
- Library - a special kind of Lifecycle Environment that does not have a parent. The library serves as the main container for synced content such as products, puppet modules, and published content views. Every organization has a library. Subsequent environments are derived from the library. The first node of an environment is the Library, all future environments are derived from the library and follow the library in promotion order.
- Lifecycle Environment Path - Sequence of lifecycle environments that form the content promotion order.

General Workflow

First create a lifecycle environment connected to the library life cycle environment and promote content views to the new lifecycle environment. A [Content Host](#) can now register directly to the promoted content view in the promoted environment or library therein. Updates will be available as soon as new content is synced and promoted.

Viewing the list of lifecycle environments

From the web UI, navigate to:

Content -> Lifecycle Environments



The screenshot shows the Foreman web interface with the 'Lifecycle Environment Paths' page selected. The top navigation bar includes 'Default_Organization', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', 'Admin User', and 'Administrator'. The main content area shows a single path entry: 'Library'. Below the path entry is a '+' button to add more paths. The URL in the browser is [https://foreman.example.com/lifecycle_environments_paths](#).

Creating a lifecycle environment

Click on the + next to the Library or the prior environment to add a new path



Lifecycle Environment Paths

Environment Path



+ New Environment Path

Close

Create Environment

Name*

Label*

Description

Creating a lifecycle environment path

Click on the New Environment Path



Lifecycle Environment Paths

Environment Path



+ New Environment Path

Close

Create Environment

Name*

Label*

Description



FOREMAN

Default_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾

Admin User ▾ Administer ▾

Lifecycle Environment Paths

Create Successful.

Environment Path

Library > Dev2 > +

Library > Dev > QA > Prod > +

View/Updating environment name

Click on the name of the environment.

FOREMAN

Default_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾

Admin User ▾ Administer ▾

Lifecycle Environment Paths

+ New Environment Path

Environment Path

Library > Dev2 > +

Library > Dev > QA > Prod > +

Environment Dev

Name: Dev

Label: Dev

Description:

Close

Foreman 2.3.3 has been released! Follow the quick start to install it.

Foreman 2.2.2 has been released! Follow the quick start to install it.

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

1. Installation

1.1 Katello

1.2 Capsule

1.3 Clients

2. Upgrade

2.1 Katello

2.2 Capsule

2.3 Clients

3. Release Notes

4. CLI

5. User Guide

5.1 Activation Keys

5.2 Backup

5.3 Capsules

5.4 Content

5.5 Content Hosts

5.6 Content Views

5.7 Disconnected

5.8 Docker Management

5.9 Email Notifications

5.10 Errata

5.11 Glossary

5.12 GPG Keys

5.13 Host Collections

5.14 Lifecycle Environments

5.15 Provisioning

5.16 Puppet Integration

5.17 Red Hat Content

6. Troubleshooting

Provisioning

See [the Foreman manual](#) for general information on configuring provisioning.

Templates

Katello ships a number of templates in addition to the standard Foreman ones. When using these templates, if a host has a Host group with an [Activation Key](#), it will register as a Content Host automatically.

- **Katello Kickstart Default** - Kickstart template for Fedora, CentOS, RHEL, and other Red Hat-compatible operating systems.
- **Katello Kickstart Default Finish** - image-based provisioning
- **Katello Kickstart Default User Data** - cloud-init template for EC2 and OpenStack
- **subscription_manager_registration** - Snippet for registering a host for content

To customize any of the above templates, simply clone them and add your changes.

When you synchronize a repository with a distribution such as Fedora or CentOS, Katello will automatically create the operating system and assign these default templates. You may change the defaults by going to Administer > Settings, and selecting the Katello tab.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Managing Puppet Content

Importing the Puppet Forge

The [Puppet Forge](#) is a collection of puppet modules written by the community which can be used to manage hosts in Katello. These modules can be used in content views as described in the [content views guide](#) in order to configure the running hosts.

To import the puppet forge navigate to

Content > Products

Click on the *+New Product* button.

Once the product is created, select the product and click the *Create Repository* button. Fill out the repository as shown:

The screenshot shows the Foreman web interface. In the top navigation bar, 'Default Organization' is selected. On the right, 'Admin User' and 'Administrator' are visible. Below the navigation, a search bar and a message 'Showing 1 of 1 (1 Total)' are present. A toolbar with buttons for 'Bulk Actions', 'Repo Discovery', and '+ New Product' is at the top right. The main area is titled 'Products' and shows a single entry named 'Puppet'. A modal window is open for 'Product Puppet'. It has tabs for 'Details', 'Repositories', and 'Tasks', with 'Repositories' selected. The 'Repositories' tab contains a sub-section titled '<< Add New Repository'. It includes fields for 'Name*' (set to 'forge'), 'Label*' (set to 'forge'), 'Type*' (set to 'puppet'), and 'URL' (set to 'https://forge.puppetlabs.com/'). At the bottom of this section are 'Cancel' and 'Save' buttons. The 'Save' button is highlighted in blue.

This can be done via the CLI:

```
hammer product create
--organization "Default Organization"
--name Puppet

hammer repository create
--organization "Default Organization"
--product Puppet
--name forge
--content-type puppet
--url "https://forge.puppetlabs.com/"
```

The repository can [now be synced](#).

Importing Puppet Modules from Git

In order to allow users to import puppet modules from Git repositories, Katello comes with a tool called 'pulp-puppet-module-builder' from the pulp-puppet-tools RPM. This utility will be available on the Katello server but it can also be installed on another machine if desired. By running the 'pulp-puppet-module-builder' against a Git repository, it will checkout the repository, build all of the modules, and publish them in a structure Katello can synchronize.

The most common method is to run the utility on the Katello server itself and publish to a local file system directory and sync against that directory.

```
mkdir /modules
chmod 755 /modules
pulp-puppet-module-builder --output-dir=/modules --url=git@mygitserver.com:mymodules.git --branch=develop
```

This will checkout the 'develop' branch of the Git repository located at 'git@mygitserver.com:mymodules.git' and publish them to the `/modules` directory. Then, from within Katello, simply set the url on your Puppet Repository to 'file://modules'. You can now sync the Repository just like any other Repository.

If you are running this on a remote machine, you will need to publish the containing to folder to a location accessible by HTTP or HTTPS.

```
mkdir /var/www/html/modules/  
chmod 755 /var/www/html/modules/  
pulp-puppet-module-builder --output-dir=/var/www/html/modules --url=git@mygitserver.com:mymodules.git --branch=develop
```

Then in Katello, simply enter 'http://HOSTNAME/modules/' for the Repository url and sync it like you normally would.

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.

[HOME](#)[GET STARTED ▾](#)[GET HELP ▾](#)[GET INVOLVED ▾](#)[NEWS ▾](#)

Katello 3.1 Documentation

3.1 ▾

- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Red Hat Content

Katello can be used to manage content associated with Red Hat products based upon available subscriptions. This includes content such as RPMs, package groups, errata and distributions.

Definitions

- Subscription Manifest - An archive file containing certificates and data that represent the subscriptions that are available. A subscription manifest is created and downloaded from the Red Hat Customer Portal.
- Repository - Collection of content (either rpm or puppet).
- Product - Collection of repositories (content hosts attach to a product).
- Library - The initial lifecycle environment where repositories are created. Content that is synced or uploaded lands in the library.

General Workflow

The following is a high-level summary of the workflow:

- Create a subscription manifest using the Red Hat Customer Portal
- Import the subscription manifest
- Enable Red Hat repositories
- Synchronize repositories
- Schedule repository synchronization
- Attach a content host to a product for Red Hat content

Create a Subscription Manifest Using the Red Hat Customer Portal

If you are a Red Hat customer, you should have access to the Red Hat Customer Portal to create and download a subscription manifest. Once created, the manifest can be imported in to a Katello Organization.

To access the Red Hat Customer Portal, [click here](#)

For details on how to create a subscription manifest, [click here](#)

Import the Subscription Manifest

Importing a subscription manifest will allow for Red Hat content associated with purchased subscriptions to be enabled and synchronized to Katello.

To import a manifest,

- navigate to: Content > Red Hat Subscriptions
- click **Choose File**
- navigate to the file containing the manifest (e.g. manifest.zip)
- click **Open**
- click **Upload**

The screenshot shows the Foreman web interface. At the top, there's a header with the Katello logo, a search bar, and navigation links for 'Monitor', 'Content', 'Hosts', 'Configure', and 'Infrastructure'. On the right, it shows 'Admin User' and 'Administrator' dropdowns. Below the header, the main title is 'Subscriptions'. A search bar and a 'Manage Manifest' button are visible. On the left, a sidebar has a 'Consumed' section. The main content area is titled 'Subscription Manifest' with a 'Close' button. It contains tabs for 'Details', 'Actions', and 'Import History', with 'Import History' currently selected. Under 'Red Hat Provider Details', it shows a repository URL: 'Repository https://cdn.redhat.com' with a checked checkbox. Below that is a section for 'Subscription Manifest' with tabs for 'Upstream', 'Subscription', 'Management', and 'Application', where 'Upstream' is selected. It shows 'No subscription manifest imported'. There's a 'Choose File' button for 'manifest.zip' and an 'Upload' button. At the bottom, there's a 'Manifest History' section with columns for 'Message' and 'Time'.

Enable Red Hat Repositories

Once a subscription manifest is imported, access is available to potentially hundreds of Red Hat Repositories (e.g. Red Hat Enterprise Linux Server, Red Hat Enterprise Virtualization...etc). This process allows you to select only those that you are interested in for your enterprise.

To enable Red Hat repositories,

- navigate to: Content > Red Hat Repositories
- select the content type: RPMs, Source RPMs, Debug RPMs, Beta, ISOs or Other
- select one or more Red Hat products (e.g. Red Hat Enterprise Linux Server)
- select one or more Repository Sets (e.g. Red Hat Enterprise Linux 6 Server (RPMs))
- select one or more Repositories (e.g. Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server)

Note:

- When enabling a RHEL repository, Red Hat recommends selecting the Server repo (e.g. 6Server, 5Server) versus a specific release (e.g. 6.2). When a specific release is necessary, the preferred way is to create a Content View with filters that narrow the content to the desired version (e.g. 6.2)
- If you plan to provision content hosts, be sure to enable both the RPM and Kickstart repositories.

Katello

fortello.devel:3000/katello/providers/redhat_provider#!=

FOREMAN

ECME Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

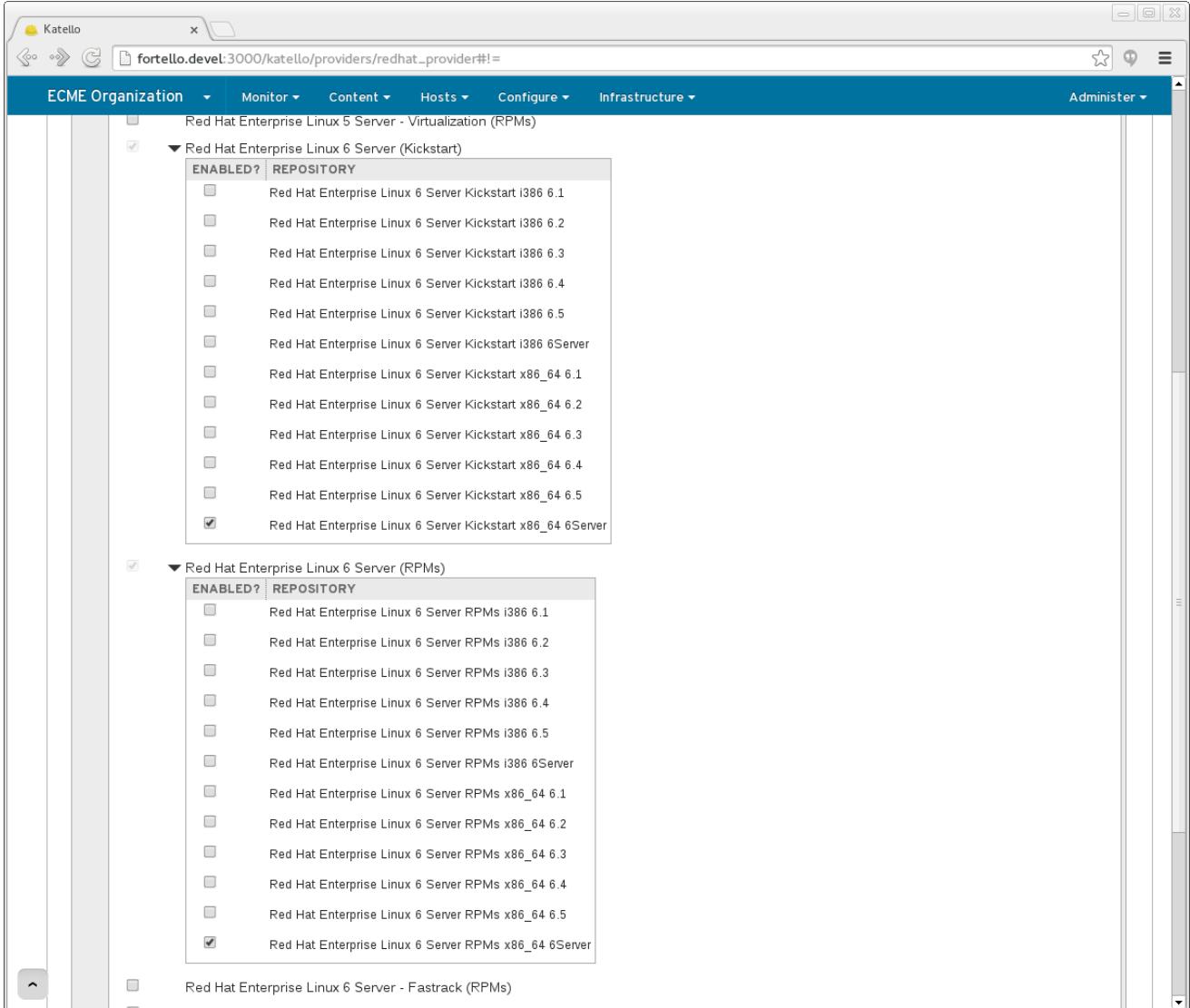
Expand each Red Hat Product below to examine the different repository sets available. When enabling a repository set, the different repositories within are discovered and may be enabled individually. X

Enable Red Hat Repositories

RPMs Source RPMs Debug RPMs Beta ISOs Other

PRODUCT

- ▶ Red Hat Enterprise Linux High Availability for RHEL Server
- ▶ Red Hat Enterprise Linux Load Balancer for RHEL Server
- ▶ Red Hat Enterprise Linux Resilient Storage for RHEL Server
- ▶ Red Hat Enterprise Linux Server
- ▶ Red Hat Enterprise Linux Workstation



Synchronize Repositories

Synchronizing a repository will retrieve all associated content and mirror the content in the Katello Library lifecycle environment.

To sync multiple repositories as well as track their progress,

- navigate to: Content > Sync Status
- expand the desired products
- select the repositories to sync
- click **Synchronize Now**

The screenshot shows the Foreman Sync Status page. At the top, there's a header with the Foreman logo, navigation links for ECME Organization, Monitor, Content, Hosts, Configure, Infrastructure, and Administerer, and a user dropdown for Admin User. Below the header is a section titled "Sync Status" with buttons for Collapse All, Expand All, Select None, Select All, and a checkbox for "Only show syncing." A table follows, with columns: PRODUCT, START TIME, DURATION, SIZE (PACKAGES), and RESULT. The table data is as follows:

PRODUCT	START TIME	DURATION	SIZE (PACKAGES)	RESULT
Red Hat Enterprise Linux Server			0 Bytes	
6Server				
x86_64				
Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server			0 Bytes (0)	
Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server			0 Bytes (0)	

A "Synchronize Now" button is located at the bottom right of the table.

Schedule Repository Synchronization

Creating a Sync Plan

Sync plans provide the ability to schedule repository synchronization on a daily, weekly or a monthly basis. Sync plans can be applied individually or to a set of repositories.

To create a Sync Plan:

- navigate to: Content > Sync Plans
- click **New Sync Plan** on the upper right

Note the following options:

- *Start Date* and *Start Time*: specify the day of the week/month and time of the day to run the recurring syncs. For example, a sync plan that starts on Sunday 2014-04-06 at 2:30 will occur every Sunday at 2:30 every week if it has a weekly interval. If on a monthly interval it would sync every month on the 6th day at 2:30.

New Sync Plan

fortello.devel:3000/katello/sync_plans#/new

FOREMAN

ECME Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

Sync Plans

Search... Showing 0 of 0 (0 Total) 0 Selected | Deselect All New Sync Plan

New Sync Plan

Name* Weekly Sync

Description

Interval weekly

Start Date* 2014-04-15

Start Time (-0400 GMT)*
17 : 06

Cancel Save

Assigning a Sync Plan to a Red Hat Product

To assign a sync plan to a product,

- navigate to Content > Sync Plans
- select your Sync Plan
- click **Products**
- click **Add**
- select the products that you want to add
- click **Add Selected** on the upper right

The screenshot shows the Foreman interface for managing sync plans. The top navigation bar includes links for Sync Plans, Monitor, Content, Hosts, Configure, Infrastructure, and Administer. The main title is "Sync Plans". A search bar and a status indicator ("Showing 1 of 0 (0 Total)") are at the top right. On the left, a sidebar lists "Name" and "Weekly Sync". The main content area is titled "Sync Plan Weekly Sync" and contains tabs for "Details" and "Products". Under "Product Management", there are "List/Remove" and "Add" buttons, and a "Filter" input field. A table lists products with columns for Name, Description, Sync Status, and Repositories. One entry is shown: "Red Hat Enterprise Linux Server" with a status of "Never synced" and 2 repositories.

Name	Description	Sync Status	Repositories
Red Hat Enterprise Linux Server		Never synced	2

Attach a Content Host to a Product for Red Hat Content

To read about registering a content host and subscribing it to a product, click [TODO](#).

Foreman 2.3.3 has been released! [Follow the quick start to install it.](#)

Foreman 2.2.2 has been released! [Follow the quick start to install it.](#)

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: github.com/theforeman/theforeman.org

Thanks to [our sponsors](#) for hosting this website and our project services.

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

Katello 3.1 Documentation

3.1



- 1. Installation
 - 1.1 Katello
 - 1.2 Capsule
 - 1.3 Clients
- 2. Upgrade
 - 2.1 Katello
 - 2.2 Capsule
 - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
 - 5.1 Activation Keys
 - 5.2 Backup
 - 5.3 Capsules
 - 5.4 Content
 - 5.5 Content Hosts
 - 5.6 Content Views
 - 5.7 Disconnected
 - 5.8 Docker Management
 - 5.9 Email Notifications
 - 5.10 Errata
 - 5.11 Glossary
 - 5.12 GPG Keys
 - 5.13 Host Collections
 - 5.14 Lifecycle Environments
 - 5.15 Provisioning
 - 5.16 Puppet Integration
 - 5.17 Red Hat Content
- 6. Troubleshooting

Katello Troubleshooting

For general support information, see [here](#).

Table of Contents

- [Sub-service Status](#)
- [Tasks](#)
- [Debug Certificate](#)

- [FAQ](#)

Sub-services status

Katello uses a set of back-end services to perform the actual job. The status of these services can negatively influence the whole system and it's one of the first things to check when some errors occur.

The status of back-end services can be found either from the Web UI on the `/about` page:

Component	Status	Message
candlepin	FAIL	Connection refused - connect(2)
candlepin_auth	FAIL	A backend service [Candlepin] is unreachable
elasticsearch	OK	
foreman_tasks	OK	
pulp	OK	
pulp_auth	OK	

Alternatively, the `hammer ping` command can be used to get this information.

`katello-service` tool can be used to restart Katello related services. See `man katello-service` for more details.

Tasks

Katello uses [Foreman Tasks](#) for orchestration between the underlying services (local database, Pulp, Candlepin...). The tasks are modeled as [Dynflow](#) processes. When something goes wrong (and there might be many reasons for this happening), Dynflow gives us the tools to recover from these errors to get to the consistent state.

Health checking

There are two properties used for identifying issues with a task:

- **state** - what phase of execution is the task in, possible values are:
 - **planning** - the planning phase of the task is performed: the operations performed in this phase shouldn't modify anything outside Katello's database. The execution of this phase happens in the web-process thread and usually should not take more than few seconds
 - **planned** - the planning phase finished and the task is waiting for the executor process (`foreman-tasks service`) to pick it up
 - **running** - the executor is performing the orchestration action, modifying the state of external services to converge to the final state
 - **paused** - something went wrong during running the task and it's waiting for the resolution (further details below)
 - **stopped** - the execution of the task finished (the success is determined by the `result` value)
 - **result** - how the task ended up (or is going to end up if we already know it)
 - **pending** - task is in the process of executing
 - **success** - no errors occurred during the execution
 - **error** - unresolved errors occurred during the execution
 - **warning** - there were errors during the execution, but they did not prevent the task from finishing or were skipped manually (further details below).

To see all the tasks in the system, one can go to `/foreman_tasks/tasks` page. To see all the tasks that failed, one can search on `result = error`:

The screenshot shows the Foreman Dynflow Console interface. At the top, there are three tabs: 'Versions for Content' (highlighted), 'Tasks', and 'Dynflow Console'. The URL in the address bar is https://katello.example.com/foreman_tasks/tasks?utf8=%E2%9C%93&search=result+error. The main header includes the Foreman logo, 'Default_Organization', and navigation links for Monitor, Content, Hosts, Configure, Infrastructure, and Administer. A user icon for 'Admin User' is also present.

Tasks

A search bar at the top of the table contains the query 'result = error'. The table lists the following tasks:

Action	State	Result	Started at	User
Publish content view 'My view'; organization 'Default_Organization'	paused	error	2014-09-12 08:51:02 UTC	admin
Synchronize repository 'zoo 1.0'; product 'Zoo'; organization 'Default_Organization'	stopped	error	2014-09-11 12:28:29 UTC	admin
Create	stopped	error	2014-09-10 17:18:23 UTC	admin
Create	stopped	error	2014-09-10 14:28:00 UTC	admin
Destroy organization 'test'	stopped	error	2014-09-10 12:24:40 UTC	admin
Destroy organization 'test'	stopped	error	2014-09-10 11:22:46 UTC	admin
Create	stopped	error	2014-09-03 19:23:00 UTC	admin
Destroy organization 'test'	stopped	error	2014-09-02 10:22:00 UTC	admin

Failed tasks include those in the 'stopped' or 'paused' state. The stopped tasks are already considered as resolved, there is no risk of inconsistency. The tasks in the 'stopped' state and the 'error' result are usually those failed during the planning phase (usually locking error or bad input data).

To see all the tasks requiring further assistance, filter on `state = paused`:

The screenshot shows the Foreman Dynflow Console interface with a different search query: 'state = paused'. The URL in the address bar is https://katello.example.com/foreman_tasks/tasks?utf8=%E2%9C%93&search=state+paused. The table displays one entry:

Action	State	Result	Started at	User
Publish content view 'My view'; organization 'Default_Organization'	paused	error	2014-09-12 08:51:02 UTC	admin

A message at the bottom left indicates 'Displaying 1 entry'.

Dealing with paused task

Once the paused task is identified, one can investigate the problem causing the errors:

The screenshot shows the Foreman web interface with a task details view. The task is currently in a 'paused' state. The action being performed is 'Actions::Pulp::Repository::CopyDistribution'. The input parameters for this action include:

```
{
  "source_pulp_id": "Default_Organization-Foreman-nightly_el6_x86_64",
  "target_pulp_id": "Default_Organization-My_view-1-Foreman-nightly_el6_x86_64",
  "clauses": "nil",
  "remote_user": "admin-283e0a66",
  "remote_cp_user": "admin",
  "locale": "en"
}
```

The resolution of the problem is dependent on the error details. The task may be resolvable by resuming the task: make sure the sub-services are running (see [Sub-services status](#) for more details) and then click 'Resume' within the web interface.

If this still doesn't help, one possible step is going to a Dynflow console (the button from task details takes you there):

The Dynflow Console screenshot shows a task history with the following steps:

- Step 5: Actions::Candlepin::Environment::Create (success) [0.20s / 0.20s]
- Step 9: Actions::Pulp::Repository::CreateInPlan (success) [0.03s / 0.03s]
- Step 12: Actions::Pulp::Repository::CopyRpm (success) [2.15s / 1.14s]
- Step 14: Actions::Pulp::Repository::CopyErrata (success) [0.69s / 0.69s]
- Step 16: Actions::Pulp::Repository::CopyPackageGroup (success) [1.26s / 0.76s]
- Step 18: Actions::Pulp::Repository::CopyYumMetadataFile (success) [71.72s / 3.56s]
- Step 20: Actions::Pulp::Repository::CopyDistribution (error) [291.56s / 5.21s] **Skip**

Started at: 2014-09-12 08:52:23 UTC
Ended at: 2014-09-12 08:57:14 UTC
Real time: 291.56s
Execution time (excluding suspended state): 5.21s

Caution: Dynflow console is considered a low-level tool and should be used very carefully, ideally discussing other options before using its features

If the failed task was taken care of by other means (performing the failed steps manually) or it was identified as not critical to the whole task, one can [skip](#) the failed step and [resume](#) the task to continue. These tasks end up with [warning](#) result at the end, to indicate there was some difficulty during the run.

Dealing with Long Running Tasks

In some cases, there might be an issue with sub-services that make it appear as if the task is running for too long without any obvious evidence that something is occurring within the task.

The first place to look in this case is filtering the tasks on `state = running` and looking at `Running Steps` in the task details:

Action:
Actions::Pulp::Repository::CopyDistribution

Input:
`{"source_pulp_id":>"Default_Organization-Foreman-nightly_el6_x86_64", "target_pulp_id":>"Default_Organization-My_view-1-Foreman-nightly_el6_x86_64", "clauses":>nil, "remote_user":>"admin-283e0a66", "remote_cp_user":>"admin", "locale":>"en"}`

Output:
`{"pulp_tasks":> [{"exception":>nil, "task_type":> "pulp.server.managers.repo.unit_association.associate_from_repo", "_href":>"/pulp/api/v2/tasks/f25da4ae-b974-43e1-b232-9671209a70fd/", "task_id":>"f25da4ae-b974-43e1-b232-9671209a70fd", "tags":> [{"pulp:repository:Default_Organization-My_view-1-Foreman-nightly_el6_x86_64", "pulp:repository:Default_Organization-Foreman-nightly_el6_x86_64", "pulp:action:associate"}], "finish_time":>nil, "start_time":>nil,`

In this case, the "start_time" => nil indicates that the task was not picked up by Pulp, which usually means some issues with running the Pulp workers. See (see [Sub-services status](#) for more details).

One can also go to the Dynflow console for even more details: the `suspended` state means that the step is waiting for the external task to finish - the `suspended` state itself doesn't have to indicate any error:

Status: running
Result: error
Started at: 2014-09-12 08:51:02 UTC
Ended at:

Plan Run Finalize

sequence concurrence

5: Actions::Candlepin::Environment::Create (success) [0.20s / 0.20s]
9: Actions::Pulp::Repository::CreateInPlan (success) [0.03s / 0.03s]
12: Actions::Pulp::Repository::CopyRpm (success) [2.15s / 1.14s]
14: Actions::Pulp::Repository::CopyErrata (success) [0.69s / 0.69s]
16: Actions::Pulp::Repository::CopyPackageGroup (success) [1.26s / 0.76s]
18: Actions::Pulp::Repository::CopyYumMetadataFile (success) [71.72s / 3.56s]
20: Actions::Pulp::Repository::CopyDistribution (suspended) [11976.09s / 6.20s] Cancel

If you're sure the underlying services are running fine, depending on the type of task, there might be a possibility to cancel the running step and possibly follow [dealing with paused tasks](#) instead.

Locking

Foreman tasks provides a locking mechanism which is important to prevent the possibility of operations colliding that are being performed concurrently on the same resource (such as synchronizing and deleting a repository at the same time).

When trying to run an operation on a resource that another task is already running, one can get `Required lock is already taken by other running tasks.`:

The screenshot shows the Foreman interface with the title bar "Publish Content View" and "Tasks". The URL in the address bar is https://katello.example.com/content_views/3/publish. The main content area is titled "Content Views" and shows a single item: "My view". Below the item, there are tabs: Versions, Content (with a dropdown arrow), Puppet Modules, History, Details, and Tasks. A red error message box is displayed, containing the text: "Required lock is already taken by other running tasks. Please inspect their state, fix their errors and resume them. × Required lock: read Conflicts with tasks: - https://katello.example.com/foreman_tasks/tasks/8cb306e7-91a9-49cc-8172-77fbcd152e0c5". Below the message box, there is a section titled "Publish New Version" with the subtext: "A new version of My view and promoted to the Library environment. It can be promoted to other environments from the Versions tab of this Content View."

A locked resource is one where another task that is related to the same resource is already running. Thus, the task being attempted will result in that task being tried in **running or paused state**. This means that the error is triggered also in cases, where there is a task with unresolved failure (see [dealing with paused tasks](#) for more details).

In rare cases, it might be hard to get into the stopped state. There is a possibility to unlock the resource in the `running / paused` task. This will switch the task into `stopped` state, freeing the resources for other tasks. **Caution: unlocking allows running other tasks to run on potentially inconsistent data, which might lead into further errors.** It's still possible to go to the Dynflow console and resume the tasks, even after using the unlock feature. There are two unlock-related buttons: `Unlock` and `Force Unlock`. The only difference between these two is the second one is allowed even when the task is in `running` state, and therefore is potentially even more dangerous than the `Unlock` button. See [dealing with tasks running too long](#) before attempting to use the `Force Unlock` option.

Debug Certificate

Debug certificates (also called Ueber Certificates) can be used to unlock all the content for a given Organization. These are meant to be used by sysadmins who are debugging issues with the Katello install.

Generating a Debug Certificate

To generate a debug certificate for a given Organization from the UI, navigate to the organizations page and click on the organization for which you want a debug certificate. Click on the button to generate and download the certificate as highlighted below:

Edit Default Organization

Name *

Label *

Description

Default System SLA

Debug certificate This certificate allows a user to view the repositories in any environment from a browser.

Users

- [Smart Proxies](#)
- [Subnets](#)
- [Compute Resources](#)
- [Media](#)

All users	<input type="checkbox"/>
Select users	All items Filter +
	Selected items -

To generate a debug certificate using the API see the API docs located on your server running at `/apidoc`.

In either case, you will get the Private Key and Certificate returned to you in a format such as :

```
Key: -----BEGIN RSA PRIVATE KEY-----
<<<DER ENCODED TEXT>>>
-----END RSA PRIVATE KEY-----

Cert: -----BEGIN CERTIFICATE-----
<<<DER ENCODED TEXT>>>
-----END CERTIFICATE-----
```

Using Firefox to browse content

If you wish to use the certificate to browse content via Firefox, do the following:

1. Copy the contents of the above file from `-----BEGIN RSA PRIVATE KEY-----` to `-----END RSA PRIVATE KEY-----` inclusive to a file called `key.pem`
2. Copy the contents of the above file from `-----BEGIN CERTIFICATE-----` to `-----END CERTIFICATE-----` inclusive to a file called `cert.pem`
3. Run the following command to create a pkcs12 file:

```
openssl pkcs12 -keypbe PBE-SHA1-3DES -certpbe PBE-SHA1-3DES -export -in cert.pem -inkey key.pem -out [NAME].pfx -name [NAME]
```

4. Provide a password when prompted.
5. Using the preferences tab, import the resulting pfx file into your browser (Edit->Preferences->Advanced Tab -> View Certificates -> Import)
6. Point your browser at `http://[FQDN]/pulp/repos/[ORG_NAME]`

To use curl to access the repository, you can provide `-cert` and `-key` options. Provided the cert is in `~/cert.pem` and key in `~/key.cert`, the following command will let you access any repository data in the organization. To check the access to a repository, checking the availability of `repodata/repoxml.xml` is usually a good idea (make sure `key.pem` and `cert.pem` are "absolute paths" otherwise it silently fails):

```
curl -k --cert ~/cert.pem --key ~/key.pem https://katello.example.com/pulp/repos/test/Dev/custom/zoo/base-two/repodata/repoxml.xml
```

Frequently Asked Questions

Can I use pulp-admin with Katello?

We do not encourage the use of pulp-admin because it has the potential to get data out of sync. However, pulp-admin can be useful when troubleshooting Katello.

1. Install needed packages

```
yum install -y pulp-admin-client pulp-rpm-admin-extensions
```

2. Edit /etc/pulp/admin/admin.conf
3. Uncomment the 'host:' line and add your server's hostname:

```
host: katello-hostname.example.com
```

4. Run `grep default_password /etc/pulp/server.conf` to lookup the admin password

```
sudo grep default_password /etc/pulp/server.conf
# default_password: default password for admin when it is first created; this
default_password: rGox3G9QhfCRD8fTsNR7FxqdgbvfJfSJ
```

5. Use pulp-admin by specifying the admin username and password:

```
pulp-admin -u admin -p rGox3G9QhfCRD8fTsNR7FxqdgbvfJfSJ repo list
```

Using pulp-admin without password

Using the 'pulp-admin login' command does not function and is not supported with Katello in an attempt to limit access to the certificate authority generated at installation time.

Katello 3.0 generates a client cert at installation time which allows usage of pulp-admin without specifying the username and password. To use this:

1. `mkdir ~/.pulp/`
2. Copy the public client cert and private key to a file together:

```
# sudo cat /etc/pki/katello/certs/pulp-client.crt /etc/pki/katello/private/pulp-client.key > ~/.pulp/user-cert.pem
```

3. Run pulp-admin without username and password:

```
pulp-admin repo list
```

How can I sync a repository like Katello does directly from the console?

Sometimes you want to debug why a synchronization of a repository from Katello is failing and rather than dig through log files and error messages it can often be easier to try to sync the repo with the "grinder" tool which is what Katello uses to download repositories. The tool can be ran from a terminal on your Katello server:

```
$ grinder yum --label=sync-test --url=https://fedorapeople.org/groups/katello/releases/yum/1.0/RHEL/6Server/x86_64/
grinder.RepoFetch: INFO    fetchYumRepo() repo_label = sync-test, repo_url =
https://fedorapeople.org/groups/katello/releases/yum/1.0/RHEL/6Server/x86_64/, basepath = ./, verify_options = {}
grinder.RepoFetch: INFO    sync-test, https://fedorapeople.org/groups/katello/releases/yum/1.0/RHEL/6Server/x86_64/
Calling RepoFetch with: cacert=<None>, clicert=<None>, clikey=<None>, proxy_url=<None>, proxy_port=<3128>, proxy_user=<None>,
proxy_pass=<NOT_LOGGED>, sslverify=<1>, max_speed=<None>, verify_options=<{}>, filter=<None>
...
grinder.ParallelFetch: INFO    5 threads are active. 8 items left to be fetched
grinder.ParallelFetch: INFO    4 threads are active. 4 items left to be fetched
grinder.ParallelFetch: INFO    WorkerThread deleting ActiveObject
grinder.ParallelFetch: INFO    Thread ending
grinder.ParallelFetch: INFO    3 threads are active. 3 items left to be fetched
grinder.ParallelFetch: INFO    WorkerThread deleting ActiveObject
grinder.ParallelFetch: INFO    Thread ending
```

You now have a directory called sync-test off of your current working directory:

```
$ ls sync-test/
converge-ui-devel-0.8.3-1.el6.noarch.rpm
elasticsearch-0.18.4-13.el6.noarch.rpm
katello-1.0.6-1.el6.noarch.rpm
katello-agent-1.0.6-1.el6.noarch.rpm
katello-all-1.0.6-1.el6.noarch.rpm
katello-certs-tools-1.1.7-1.el6.noarch.rpm
lucene3-contrib-3.4.0-2.el6.noarch.rpm
repodata
rubygem-actionmailer-3.0.10-3.el6.noarch.rpm
...
```

Foreman 2.3.3 has been released! Follow the quick start to install it.

Foreman 2.2.2 has been released! Follow the quick start to install it.

[View our Privacy Policy](#)

This web site is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). Source available: [github/theforeman/theforeman.org](https://github.com/theforeman/theforeman.org)

Thanks to [our sponsors](#) for hosting this website and our project services.