Katello 3.15 Documentation

3.15

Katello 3.15 Installation

> After installation of Katello, navigate to the `/pub` directory and trust Katello’s CA certificate for identifying web sites (e.g. `http://katello.example.com/pub/katello-server-ca.crt`).

Important Note for Existing Installations

Katello does not currently support installation on existing Foreman deployments. **DO NOT attempt to install Katello on an existing Foreman deployment**, unless you are a Foreman developer and willing to debug the broken configuration that will result from attempting an install on existing system.

Hardware Requirements

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

- Two Logical CPUs
- 12 GB of memory (16 GB or greater may be required depending on the amount of managed content)
- The filesystem holding /var/lib/pulp needs to be large, but may vary depending on how many different Operating Systems you wish to syncronize:
    - 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 syncronized later.
- The path /var/spool/squid/ is used as a temporary location for some types of repository syncs and may grow to consume 10s of GB of space before the files are migrated to /var/lib/pulp. You may wish to put this on the same partition as /var/lib/pulp.
- 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 syncronize:
    - 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 Smart Proxy 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 operating systems:

- CentOS 7 (x86_64)
- Red Hat Enterprise Linux 7 (x86_64)

Installation may be done manually or via our recommended approach of using forklift.

Required Repositories

Select your Operating System: Red Hat Enterprise Linux 7 ▾

```
subscription-manager repos --disable "*"
subscription-manager repos --enable rhel-7-server-rpms
subscription-manager repos --enable rhel-7-server-optional-rpms
subscription-manager repos --enable rhel-7-server-extras-rpms
yum install -y yum-utils
```

```
yum -y localinstall https://yum.theforeman.org/releases/2.0/el7/x86_64/foreman-release.rpm
yum -y localinstall https://fedorapeople.org/groups/katello/releases/yum/3.15/katello/el7/x86_64/katello-repos-latest.rpm
yum -y localinstall https://yum.puppet.com/puppet6-release-el-7.noarch.rpm
yum -y localinstall https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
yum -y install foreman-release-scl
```

Installation

After setting up the appropriate repositories, update your system:

```
yum -y update
```

Then 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>
```

Multiple subnets and domains

The installer only supports one subnet and one DNS domain via command line arguments. Multiple entries can be entered via /etc/foreman-installer/custom-hiera.yaml file:

```
dhcp::pools:
 isolated.lan:
   network: 192.168.99.0
   mask: 255.255.255.0
   gateway: 192.168.99.1
   range: 192.168.99.5 192.168.99.49
dns::zones:
  # creates @ SOA $::fqdn root.example.com.
  # creates $::fqdn A $::ipaddress
  example.com: {}

  # creates @ SOA test.example.net. hostmaster.example.com.
  # creates test.example.net A 192.0.2.100
  example.net:
    soa: test.example.net
    soaip: 192.0.2.100
    contact: hostmaster.example.com.

  # creates @ SOA $::fqdn root.example.org.
  # does NOT create an A record
  example.org:
    reverse: true

  # creates @ SOA $::fqdn hostmaster.example.com.
  2.0.192.in-addr.arpa:
    reverse: true
    contact: hostmaster.example.com.
```

Tuning options

The Foreman installer supports automatic tuning of your environment using predefined tuning profiles. These tuning profiles are the result of a culmination of extensive learning from Foreman environments deployed at scale in large user environments.

When the foreman-installer is run, it is deployed with a `default` predefined tuning profile. Other than the default tuned profile, foreman-installer supports 4 different tuning profiles:

- medium
- large
- extra-large
- extra-extra-large

Based on your environment needs, use one of the tuning profiles (`medium` , `large` , `extra-large` , `extra-extra-large`) in the installer. For example, `medium` profile can be applied like:

```
foreman-installer --tuning medium
```

To reset to the default profile:

```
foreman-installer --tuning default
```

Use `foreman-installer --help | grep tuning` to identify the current tuning level.

Sample output for `medium` tuning:

```
foreman-installer --help | grep tuning
    --tuning INSTALLATION_SIZE    Tune for an installation size. Choices: default, medium, large, extra-large, extra-extra-large (defau
lt: "medium")
```

Sample output for default tuning:

```
foreman-installer --help | grep tuning
    --tuning INSTALLATION_SIZE    Tune for an installation size. Choices: default, medium, large, extra-large, extra-extra-large (defau
lt: "default")
```

Note

- Definitions of various tuning profiles can be found in this directory /usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/ . Note that common.yaml is always applied and the selected tuning profile (e.g., medium) is applied on top and takes precedence.
- Using the --tuning option does not update /etc/foreman-installer/custom-hiera.yml , instead it directly updates the required configuration as specified in the corresponding tuning profile. You can still use custom-hiera.yml to override any configuration if really needed.
- If you had already used custom-hiera.yml and starting to use the tuned profiles, you may want to review the definition of tuned profiles (/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml and /usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/) and remove the duplicated configuration entries from your custom-hiera.yml .
- You can also optionally use foreman-installer --tuning <profile> --noop to run the installer in a test mode and identify what configurations will be changed before actually running the installer.

Which tuning profile should you choose?

It is difficult to find the exact tuning profile for a specific environment in the first attempt because it depends on various factors like the number of managed hosts, the features used in scale (E.g., Remote Execution), the bulk actions on hosts, the total amount of content, amount of host traffic to foreman, etc. Our recommendation is that you start with the tuning profile guidance as shown in the below table based on the number of managed hosts and scale up your environment as needed.

Note

- The information in the table below is just a guidance. It is strongly recommended that you monitor the foreman environment regularly and tune up as required.
- The RAM and CPU Cores check is also integrated into the foreman-installer now. Use disable-system-checks if you like to skip this check in the installer.

Tuned profile	Number of Managed hosts	Minimum Recommended RAM	Minimum Recommended CPU Cores
default	up-to 5000	20G	4
medium	5000 - 10000	32G	8
large	10000 - 20000	64G	16
extra-large	20000 - 60000	128G	32
extra-extra-large	20000 - 60000	256G	48

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.

Katello 3.15 Documentation

3.15 ▾

Smart Proxy Installation

Hardware Requirements

The Smart Proxy server is only supported on x86_64 Operating Systems

- 2 Two Logical CPUs
- 8 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 Smart Proxy services and required ports.

Installation

Install needed packages:

The same yum repositories need to be configured on the Smart Proxy server as the main Katello server. See the installation guide for the list of required repositories.

Once you get the repositories configured, install the formean-proxy-content package on the Smart Proxy

```
yum install -y foreman-proxy-content
```

Generate Certificates for the Smart Proxies

Prior to installing the Smart Proxy, we need to generate certificates on the main **Katello** server:

```
foreman-proxy-certs-generate --foreman-proxy-fqdn "myproxy.example.com"\
                             --certs-tar     "~/myproxy.example.com-certs.tar"
```

In the above example, replace 'myproxy.example.com' with your Smart Proxy'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 Smart Proxy on using whatever method you prefer (e.g. SCP).

The foreman-proxy-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 ~/myproxy.example.com-certs.tar to the system myproxy.example.com
  3. Run the following commands on the Smart Proxy (possibly with the customized
     parameters, see foreman-installer --scenario foreman-proxy-content --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 foreman-proxy-content\
                    --foreman-proxy-content-parent-fqdn   "katello.example.com"\
                    --foreman-proxy-register-in-foreman   "true"\
                    --foreman-proxy-foreman-base-url      "https://katello.example.com"\
                    --foreman-proxy-trusted-hosts         "katello.example.com"\
                    --foreman-proxy-trusted-hosts         "myproxy.example.com"\
                    --foreman-proxy-oauth-consumer-key    "UVrAZfMaCfBiiWejoUVLYCZHT2xhzuFV"\
                    --foreman-proxy-oauth-consumer-secret "ZhH8p7M577ttNU3WmUGWASag3JeXKgUX"
\
                    --certs-tar-file                      "/root/myproxy.example.com-certs.tar"
  The full log is at /var/log/foreman-proxy-certs-generate.log
```

Install Smart Proxy

Use the provide installation command from `foreman-proxy-certs-generate`, and tailor for your own purposes as needed. The defaults will give you a Smart Proxy 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.

Katello 3.15 Documentation

3.15

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.

Select your Operating System: Enterprise Linux 7 (CentOS, etc.) ▾

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
yum install -y https://yum.theforeman.org/client/2.0/el7/x86_64/foreman-client-release.rpm
```

Now you are ready to install the client package:

The `katello-host-tools` package reports errata & package profile information, but does not allow you to run remote actions on the clients.

```
yum install katello-host-tools
```

We generally recommend using Foreman Remote Execution or Ansible for remote actions, but we also offer a messaging bus based client that does have some limitations when used with a large number of clients.

```
yum install katello-agent
```

Optionally you can also install `katello-host-tools-tracer` and the client will report processes that need restarting after an update back to the Katello server.

```
yum install katello-host-tools-tracer
```

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.

When provisioning new clients that should use Puppet 5, set a parameter called ‘enable-puppet5’ to ‘true’, so the templates know which package to install and where to place the configuration. This parameter can be placed at the host, host group, or another appropriate level of the hierarchy.

Katello 3.15 Documentation

3.15

Katello Upgrade

Katello supports upgrades from the previous two versions only. Upgrades should be performed sequentially without skipping versions in between.

Pre-upgrade

Before upgrading, run the upgrade check script that will check for any active tasks:

```
foreman-rake katello:upgrade_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
```

Step 3 - Repositories

Update the Foreman and Katello release packages:

- RHEL7 / CentOS 7:

```
yum update -y https://fedorapeople.org/groups/katello/releases/yum/3.15/katello/el7/x86_64/katello-repos-latest.rpm
yum update -y https://yum.theforeman.org/releases/2.0/el7/x86_64/foreman-release.rpm
yum update -y foreman-release-scl
```

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 adjust the configuration to reflect what's new in Katello 3.15.

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

Step 6 - Reboot if necessary

If kernel packages are updated during Step 2 the system must be rebooted to ensure the new kernel and SELinux policy are loaded. If there are no kernel or selinux updates then this step can be omitted.

Congratulations!

You have now successfully upgraded your Katello to 3.15.

For a rundown of what was added, please see the release notes.

If the above steps failed, please review /var/log/foreman-installer/katello.log and let us know about it if unable to resolve.

Pulp3 Migration

Katello 3.15 introduces Pulp 3 alongside Pulp 2. For this release only Docker and File content types can be migrated to Pulp 3 via the migration process. This process is optional for this release, but will need to be done by a future release. Depending on how much File and Docker content is present, this can take a very long time. The largest part of the process can be run without downtime.

1. Migrate the content:

```
foreman-maintain advanced procedure run content-prepare
```

Note that this is run without downtime, while content and repositories are migrated in the background. This can be run multiple times.

1. Perform the switchover to pulp3:

```
foreman-maintain advanced procedure run content-switchover
```

This will require some downtime to do the final switchover, but this will:

- Perform one last content migration
- Perform a check to verify everything that is needed has been migrated
- Re-run the installer to ‘switch’ over to pulp3

If you run into any issues with this process, please file a support issue with ‘[ContentMigration]’ in the subject.

Katello 3.15 Documentation

3.15

Smart Proxy 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:

- RHEL7 / CentOS 7:

```
yum update -y https://fedorapeople.org/groups/katello/releases/yum/3.15/katello/el7/x86_64/katello-repos-latest.rpm
yum update -y https://yum.theforeman.org/releases/2.0/el7/x86_64/foreman-release.rpm
```

Step 3 - Update Packages

Clean the yum cache

```
yum clean all
```

Update packages:

```
yum update -y
```

```
yum install foreman-proxy-content
```

Step 4 - Regenerate Certificates

On the Katello server, regenerate the certificates tarball for your Smart Proxy:

```
foreman-proxy-certs-generate --foreman-proxy-fqdn "myproxy.example.com"\
                   --certs-update-all\
                   --certs-tar     "~/myproxy.example.com-certs.tar"
```

And copy them to your Smart Proxy:

```
scp ~/myproxy.example.com-certs.tar myproxy.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 3.15

```
foreman-installer --scenario foreman-proxy-content --upgrade\
               --certs-tar-file ~/myproxy.example.com-certs.tar\
               --certs-update-all --certs-regenerate true --certs-deploy true
```

Congratulations! You have now successfully upgraded your Smart Proxy to 3.15 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/foreman-proxy.log – if any of the "Upgrade step" tasks failed, you may try to run them manaully below to aid in troubleshooting.

Katello 3.15 Documentation

3.15

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: Enterprise Linux 5 (RHEL, CentOS, etc.) ▾

```
yum update -y https://yum.theforeman.org/client/2.0/el5/x86_64/foreman-client-release.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 3.15 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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15

Katello 3.15 Release Notes

For the full release notes, see the Changelog.

Features

- Improved experience for Simple Content Access users with clarified messaging and UI tweaks
- Errata can be searched by their issued date
- hammer gpg has been renamed to hammer content-credentials to match the UI
- Host updates will no longer use Dynflow for increased performance

- Repository support for pulpcore rpm content types.
- Improved visibility of pulp2 and pulpcore service responsibility in smart proxy service details
- Improved performance of Subscriptions list API
- Content count information is now stored on Content View Versions directly for increased performance

Deprecations

- Puppet and OSTree repository types will no longer be available in Katello 3.16
- katello-agent will be removed in Katello 3.16

Bug Fixes

Katello 3.15 includes 77 bug fixes, which can be seen here

Contributors

- Avi Sharvit
- Brad Buckingham
- Chris Roberts
- Dhananjay Gupte
- Eric Helms
- Ian Ballou
- James Jeffers
- Jeremy Lenz
- John Mitsch
- Jonathon Turel
- Justin Sherrill
- Marek Hulán
- Michael Moll
- Partha Aji
- Ranjan Kumar
- Ron Lavi
- Samir Jha
- Shimon Shtein
- Shira Maximov
- Swetha Seelam Lakshmi Narayanan
- Walden Raines

Katello 3.15 Documentation

3.15 ▾

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: Red Hat Enterprise Linux 7

```
yum -y  --disablerepo="*" --enablerepo=rhel-7-server-rpms install yum-utils wget
yum-config-manager --disable "*"
yum-config-manager --enable rhel-7-server-rpms
yum-config-manager --enable rhel-7-server-optional-rpms
yum-config-manager --enable rhel-7-server-extras-rpms
```

```
yum -y localinstall https://fedorapeople.org/groups/katello/releases/yum/3.15/katello/el7/x86_64/katello-repos-latest.rpm
yum -y localinstall https://yum.theforeman.org/releases/2.0/el7/x86_64/foreman-release.rpm
yum -y localinstall https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

After setting up the appropriate repositories, install Katello:

```
yum -y install tfm-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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15

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
- System Purpose

Definitions

- Content Host
- Host Collection - A statically defined group of Content Hosts.
- Subscription - The right to receive the associated content from Katello.
- System Purpose - Helps auto-attach find a subscription that satisfies the intended use of the system

General Features

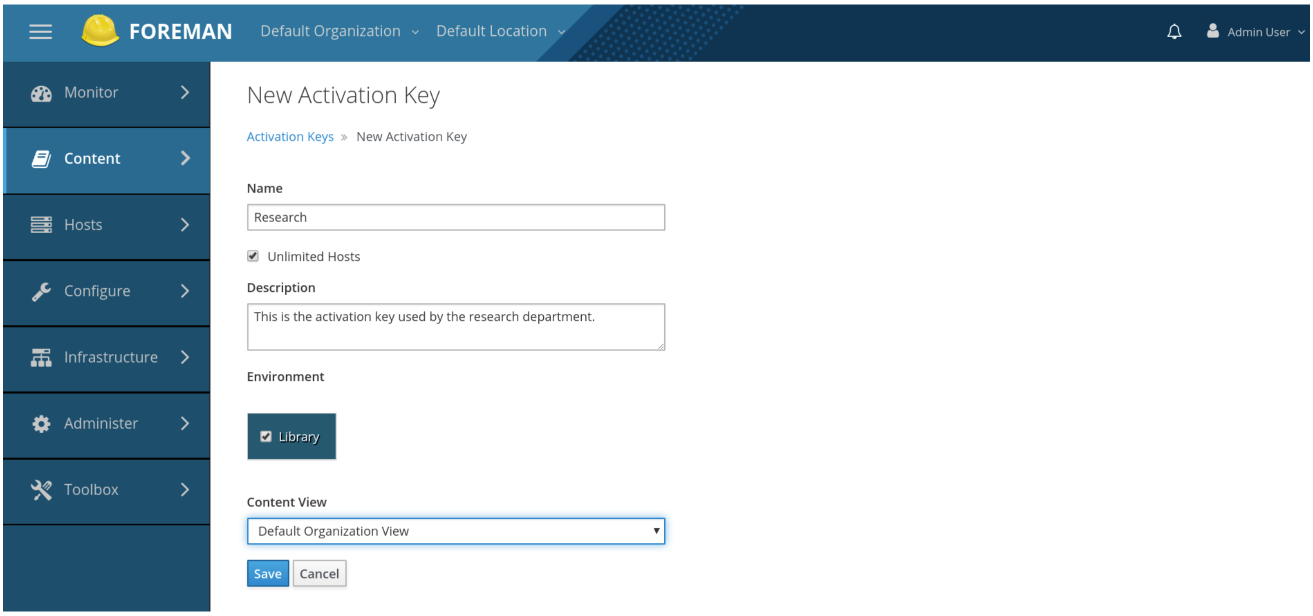
The following is a high-level summary of the Activation Key features:

- Create an Activation Key
- Add subscriptions to an Activation Key
- Change repository enablement for an Activation Key
- Add Host Collections to an Activation Key
- Add System Purpose details to an Activation Key
- Register a Content Host using an Activation Key
- View Content Hosts registered with an Activation Key

Create an Activation Key

To create a new key,

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



New Activation Key

Name: Research

☑ Unlimited Hosts

Description: This is the activation key used by the research department.

Environment: ☑ Library

Content View: Default Organization View

- *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**

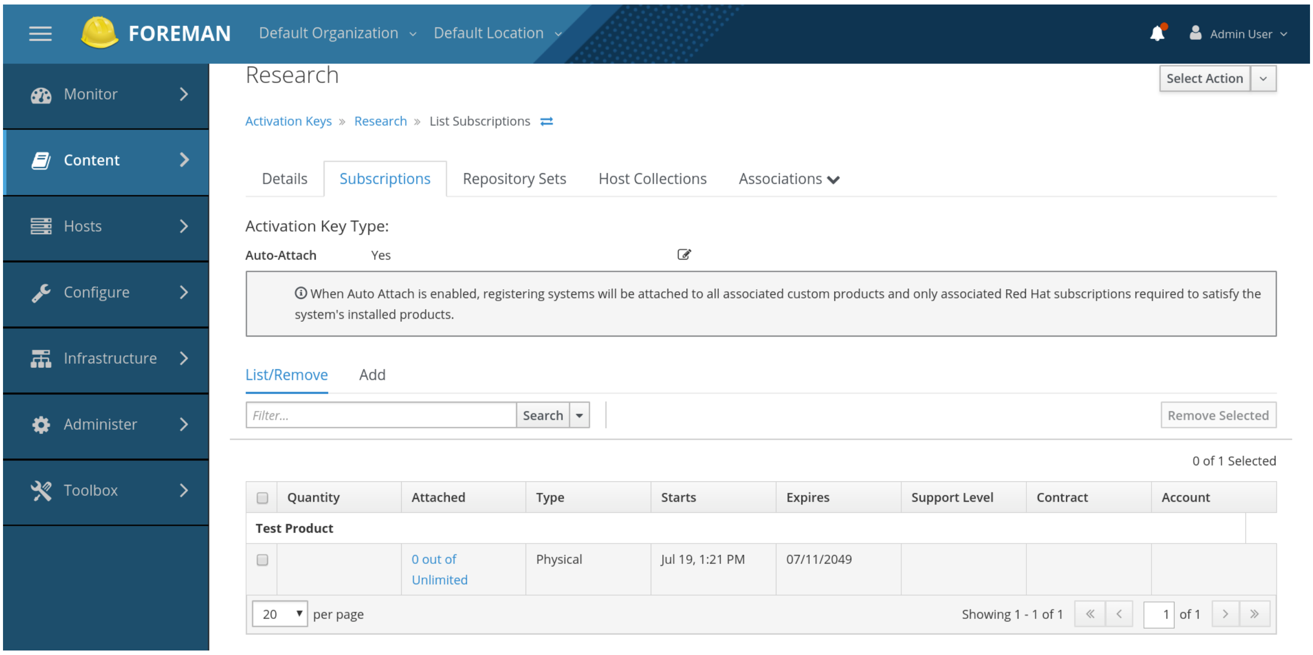
The **Auto-Attach** setting controls how the group of subscriptions are processed during registration.

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 attached 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.



Activation Key Type:

Auto-Attach Yes

ⓘ When Auto Attach is enabled, registering systems will be attached to all associated custom products and only associated Red Hat subscriptions required to satisfy the system's installed products.

0 of 1 Selected

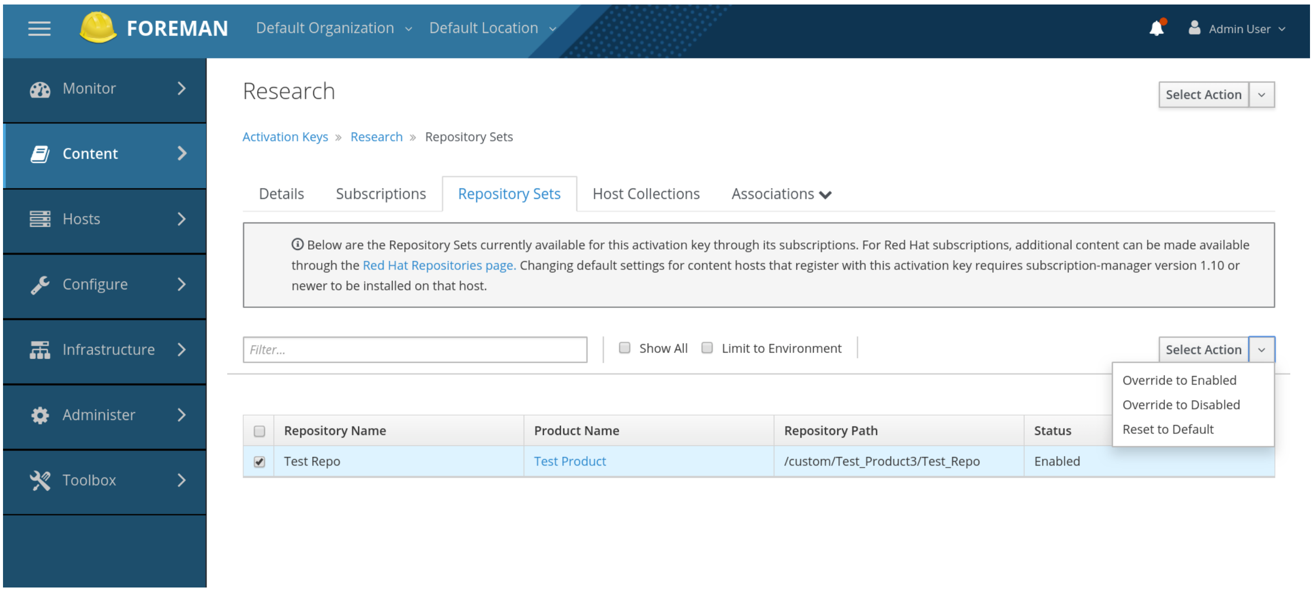
□	Quantity	Attached	Type	Starts	Expires	Support Level	Contract	Account
Test Product								
□		0 out of Unlimited	Physical	Jul 19, 1:21 PM	07/11/2049			

Change Repository Enablement for an Activation Key

To change repository enablement settings using a key:

- navigate to: Content > Activation Keys
- select the key you want from the list
- click **Repository Sets**
- select the repository you want to modify
- select the value you want to change from the **Select Action** menu(‘Override to Enabled’, ‘Override to Disable’, ‘Reset to Default’)

View and change settings:



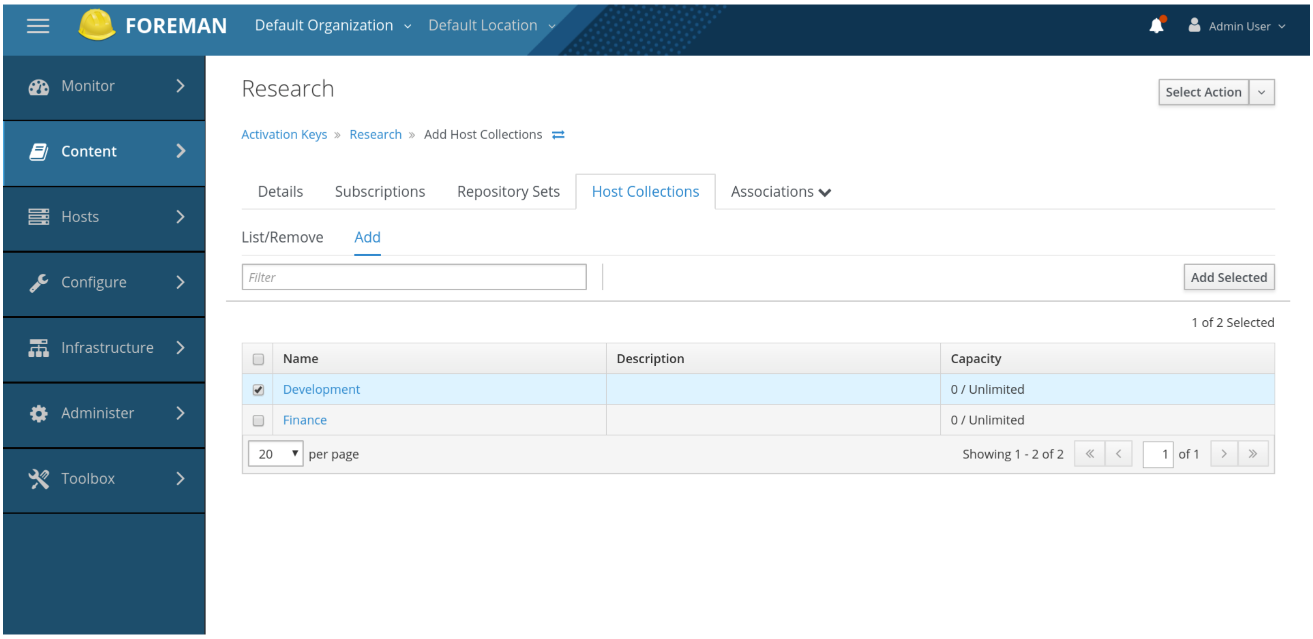
ⓘ Below are the Repository Sets currently available for this activation key through its subscriptions. For Red Hat subscriptions, additional content can be made available through the Red Hat Repositories page. Changing default settings for content hosts that register with this activation key requires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

□	Repository Name	Product Name	Repository Path	Status
☑	Test Repo	Test Product	/custom/Test_Product3/Test_Repo	Enabled

Add Host Collections to an Activation Key

To add Host Collections to a key:

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

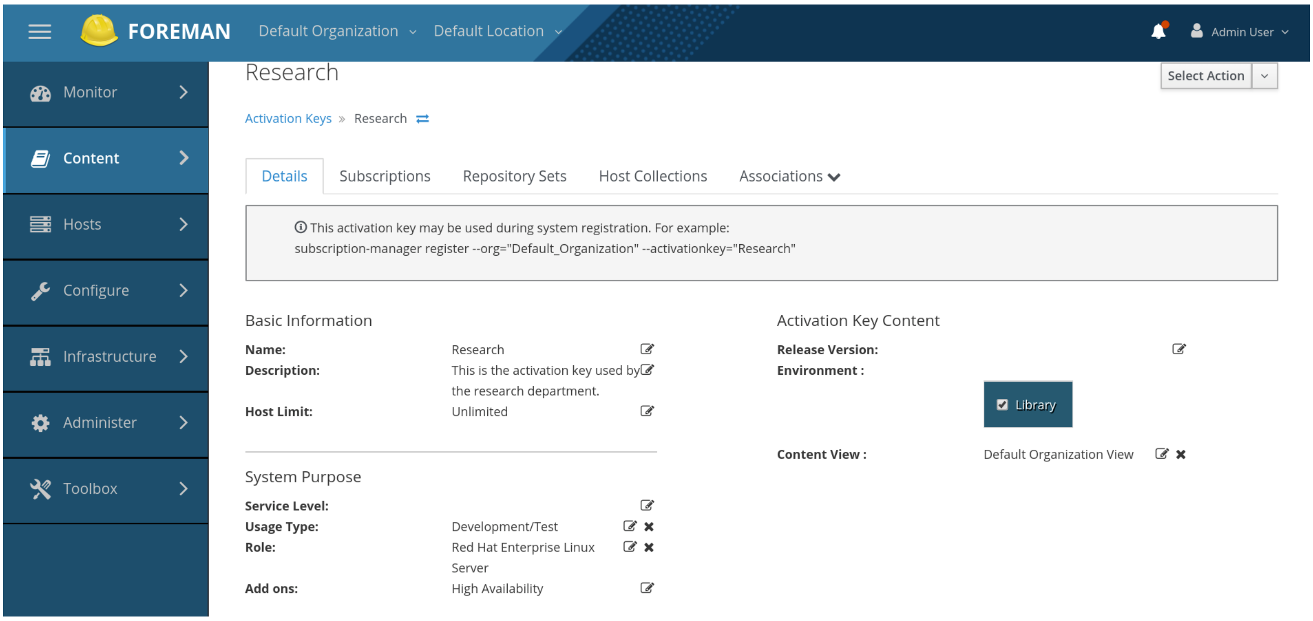


□	Name	Description	Capacity
☑	Development		0 / Unlimited
□	Finance		0 / Unlimited

Add System Purpose details to an Activation Key

To add System Purpose to a key:

- navigate to: Content > Activation Keys
- select the key you want from the list
- There is a section for System Purpose where you can set the Service Level, Usage Type, Role, and Add-ons for a system.



Research

Activation Keys » Research

This activation key may be used during system registration. For example:
subscription-manager register --org="Default_Organization" --activationkey="Research"

Basic Information

Name: Research
Description: This is the activation key used by the research department.
Host Limit: Unlimited

System Purpose

Service Level:
Usage Type: Development/Test
Role: Red Hat Enterprise Linux Server
Add ons: High Availability

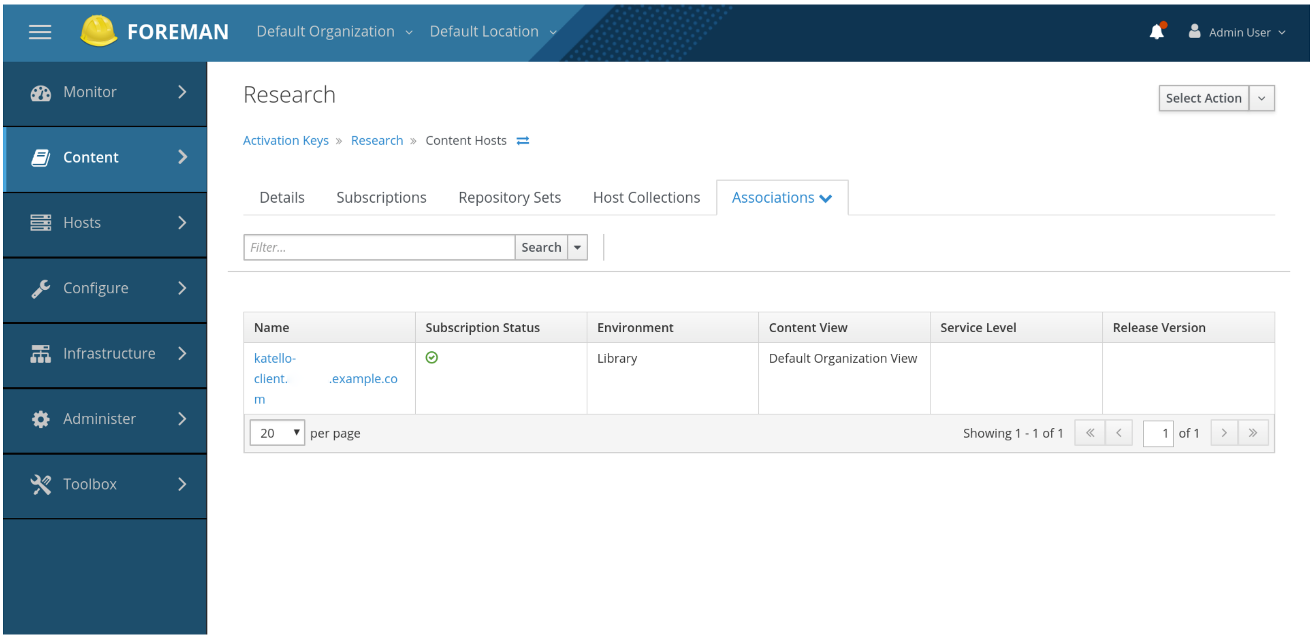
Activation Key Content

Release Version:
Environment : ☑ Library
Content View : Default Organization View

View Content Hosts registered with the Activation Key

To view Content Hosts registered with a particular Activation Key:

- navigate to: Content > Activation Keys
- select the key you want from the list
- select **Content Hosts** under **Associations**



Research

Activation Keys » Research » Content Hosts

Name	Subscription Status	Environment	Content View	Service Level	Release Version
katello-client. .example.com	✓	Library	Default Organization View		

20 per page Showing 1 - 1 of 1 1 of 1

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15

Backup

Please use Foreman Maintain for backup and restore functionality. Foreman-maintain is a dependency of Katello starting in 3.7.

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15

Smart Proxies

What are Smart Proxies?

The Smart Proxy server is a Katello component that provides federated services to discover, provision, control, and configure hosts. Each Katello server includes a Default Smart Proxy, and you may deploy additional Smart Proxies to remote data centers. A Smart Proxy 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 Smart Proxy server is a means to scale out the Katello installation. Organizations can create various Smart Proxies 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 Smart Proxy servers subscribed to that environment. Hosts pull content and configuration from the Katello Smart Proxy servers in their location and not from the central server.

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

What is a Foreman Proxy with Content?

A Katello Smart Proxy 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 Smart Proxy. Larger deployments would have a single Katello server with multiple Smart Proxies attached, with these remote Smart Proxies deployed to various datacenters. Smart Proxies can also be used to scale the number of hosts attached to a single Katello server.

Installation

See Smart Proxy Installation

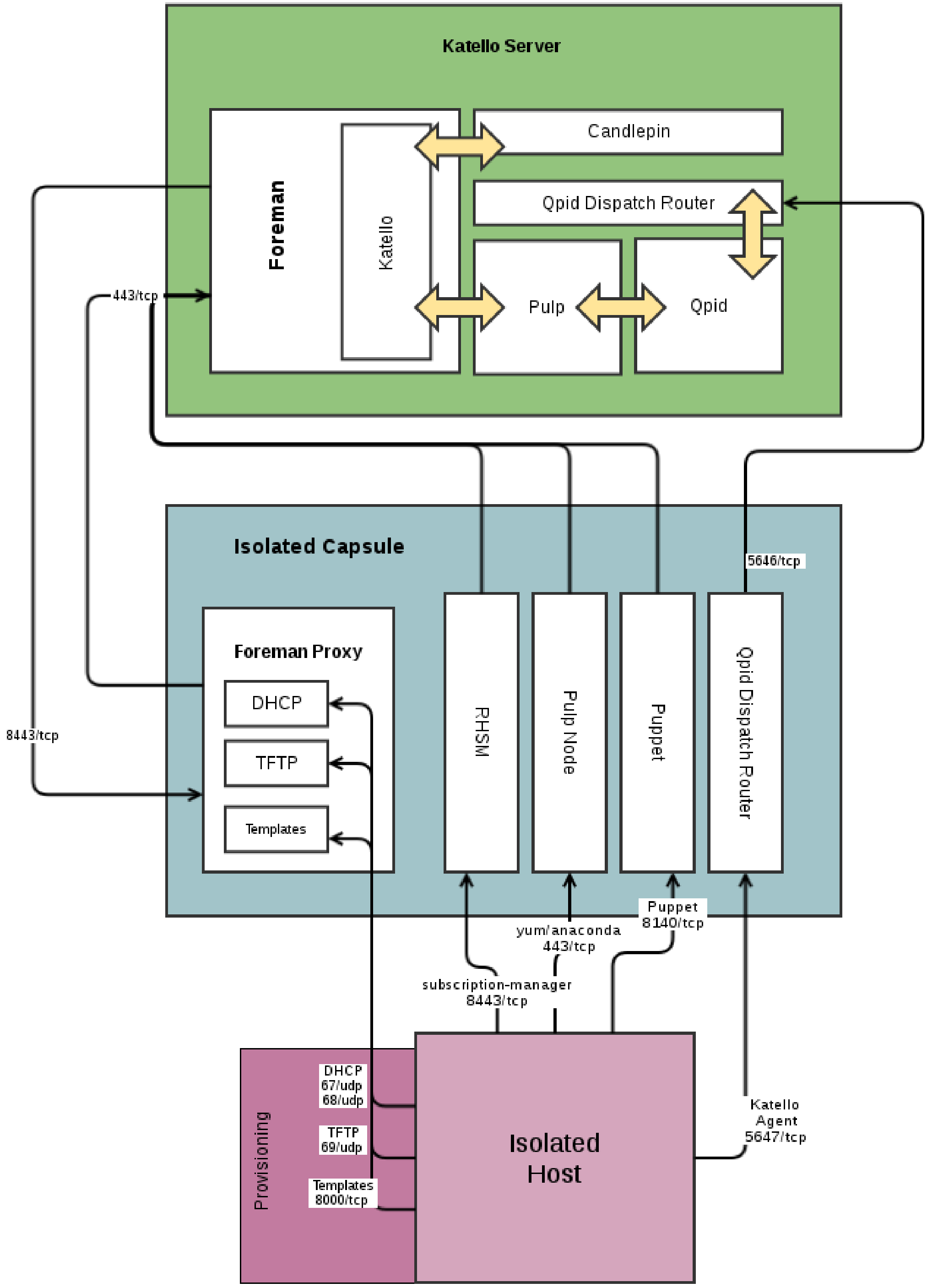
Removal

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

```
katello-remove
```

Smart Proxy Isolation

The goal of Smart Proxy 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 Smart Proxy itself. The following section details the communication clients need to have with a Smart Proxy. 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 Smart Proxies 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 Smart Proxy 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 Smart Proxy 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 Smart Proxy on port 5647/tcp

3 - Puppet & Puppet CA

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

Install Option:

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

Required Connectivity:

- Clients need to communicate with the Smart Proxy 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 Smart Proxy 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 Smart Proxy, and a DHCP server available. While not required, the Smart Proxy can provide the DHCP service. In order for the installer to obtain its kickstart template from the Smart Proxy, 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 Smart Proxy 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.

Katello 3.15 Documentation

3.15

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 bandwith 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

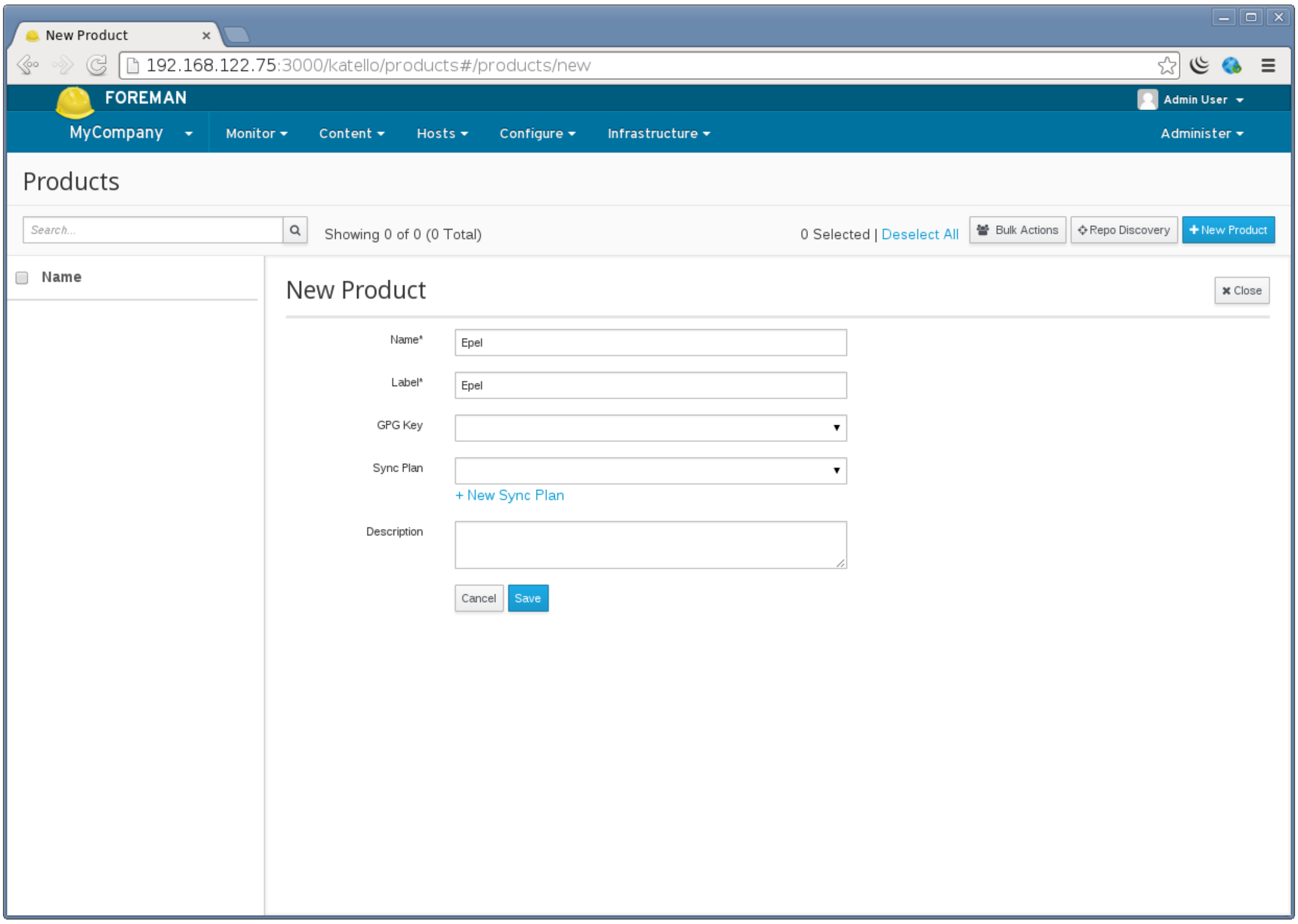
Definitions

- 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)



New Product

Name* Epel

Label* Epel

GPG Key

Sync Plan

Description

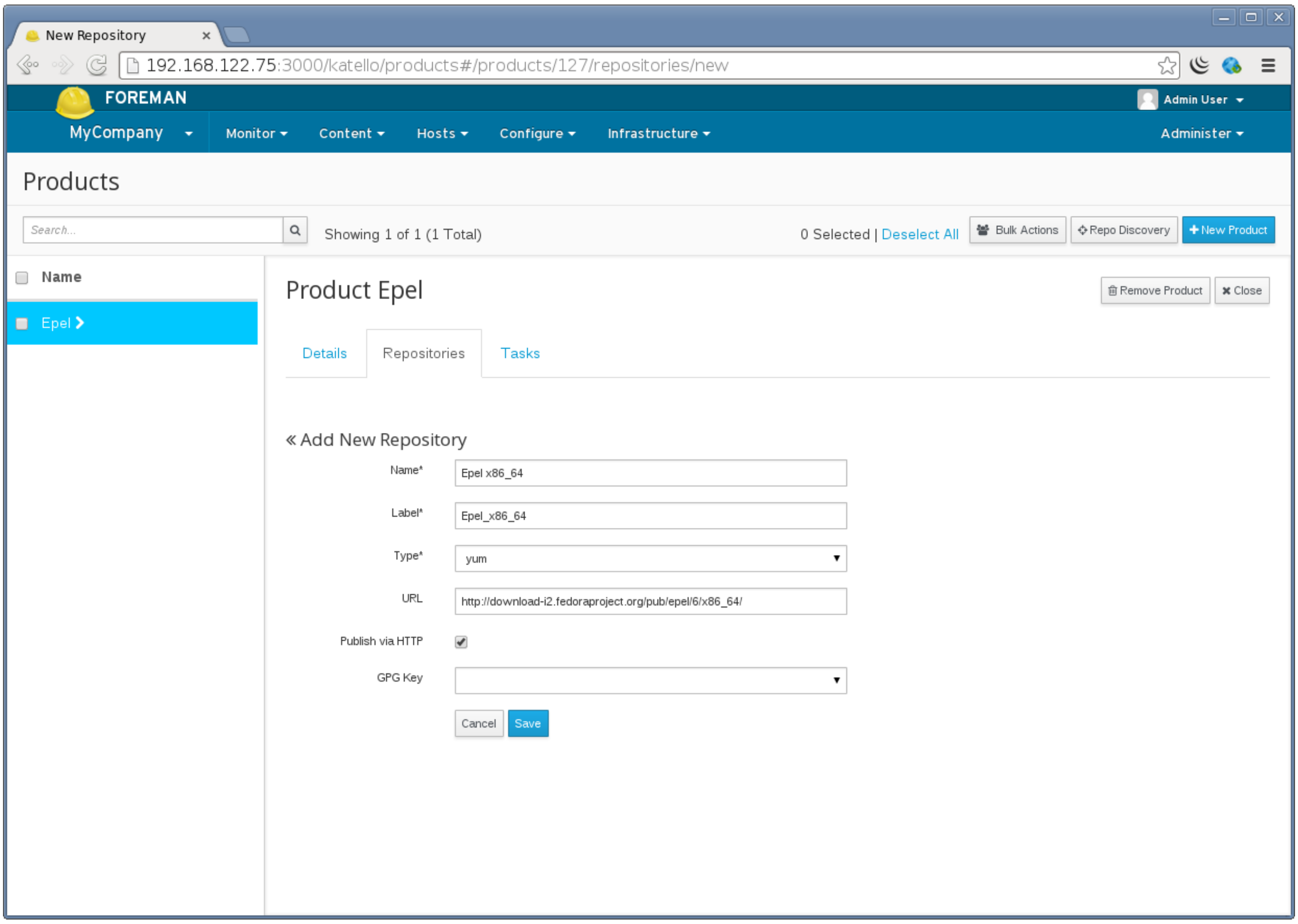
Creating a Repository

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/'.



Products

Showing 1 of 1 (1 Total)

0 Selected | Deselect All

Product Epel

« Add New Repository

Name* Epel x86_64

Label* Epel_x86_64

Type* yum

URL http://download-i2.fedoraproject.org/pub/epel/6/x86_64/

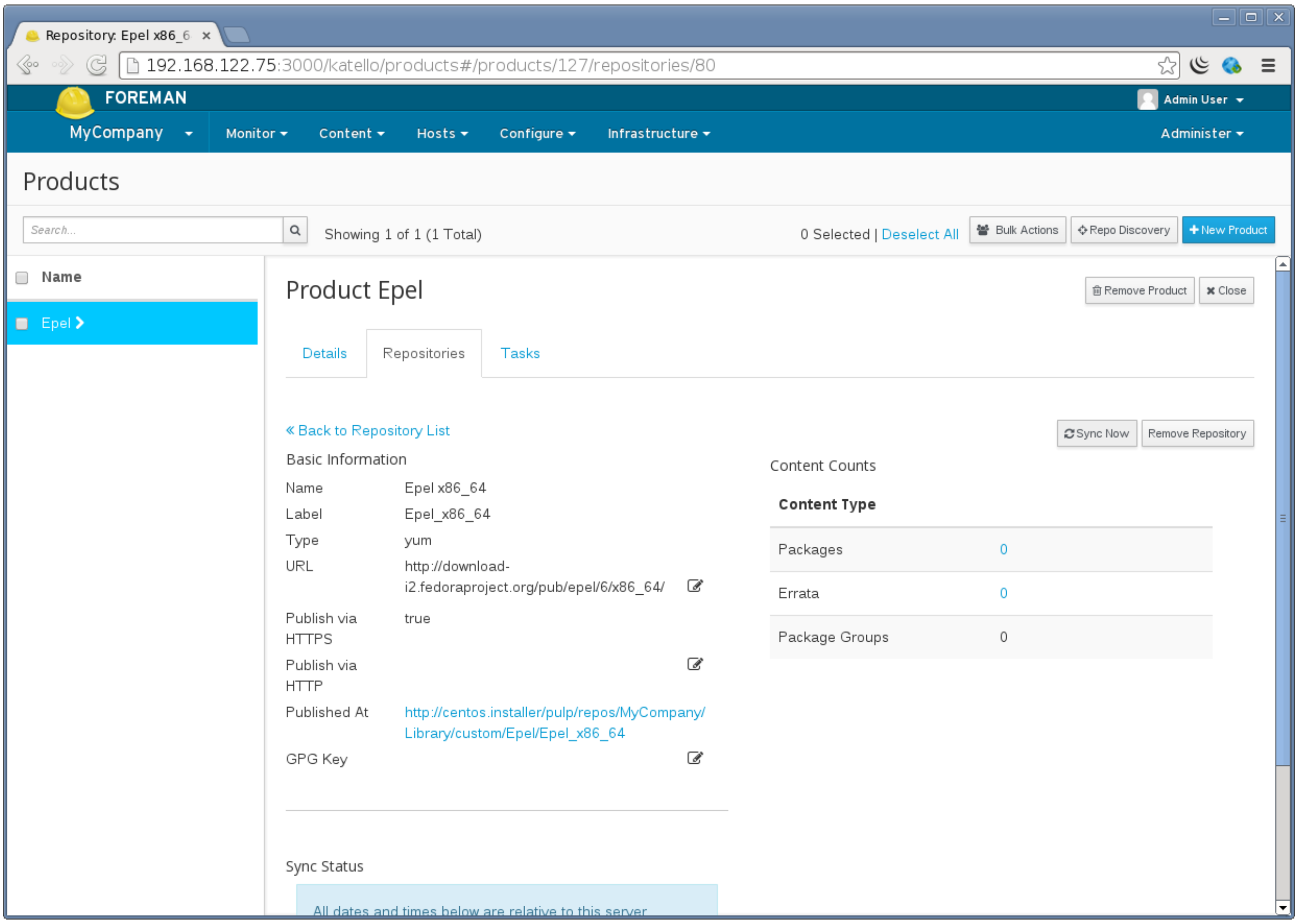
Publish via HTTP ☑

GPG Key

Syncing a Repository

From the web UI, navigate to:

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



192.168.122.75:3000/katello/products#/products/127/repositories/80

Products

Showing 1 of 1 (1 Total)

0 Selected | Deselect All

Name

Epel

Product Epel

« Back to Repository List

Basic Information

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	
Published At	http://centos.installer/pulp/repos/MyCompany/Library/custom/Epel/Epel_x86_64
GPG Key	

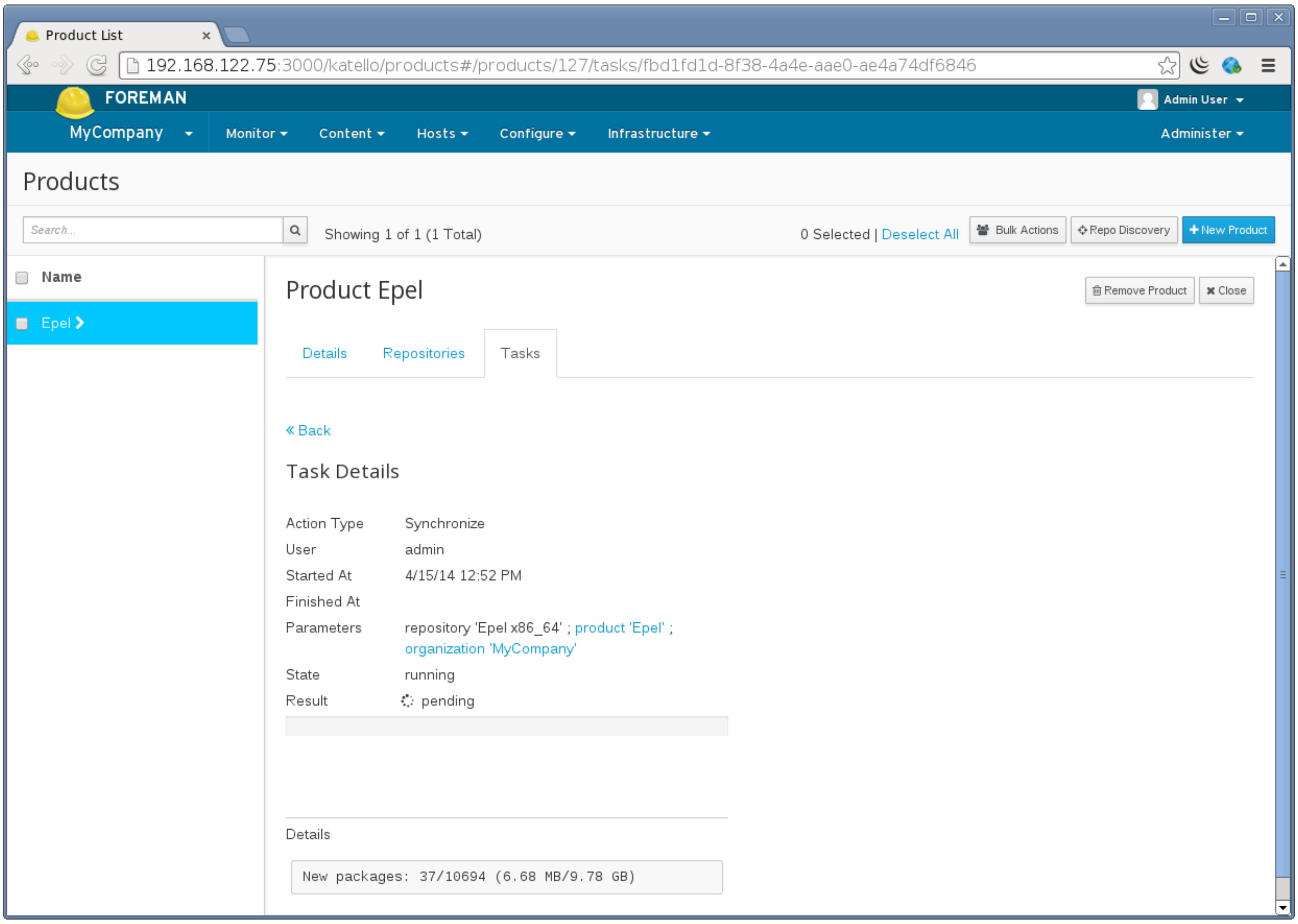
Content Counts

Content Type	
Packages	0
Errata	0
Package Groups	0

Sync Status

All dates and times below are relative to this server

The progress will be displayed:



192.168.122.75:3000/katello/products#/products/127/tasks/fbd1fd1d-8f38-4a4e-aae0-ae4a74df6846

Products

Showing 1 of 1 (1 Total)

0 Selected | Deselect All

Name

Epel

Product Epel

« Back

Task Details

Action Type	Synchronize
User	admin
Started At	4/15/14 12:52 PM
Finished At	
Parameters	repository 'Epel x86_64' ; product 'Epel' ; organization 'MyCompany'
State	running
Result	pending

Details

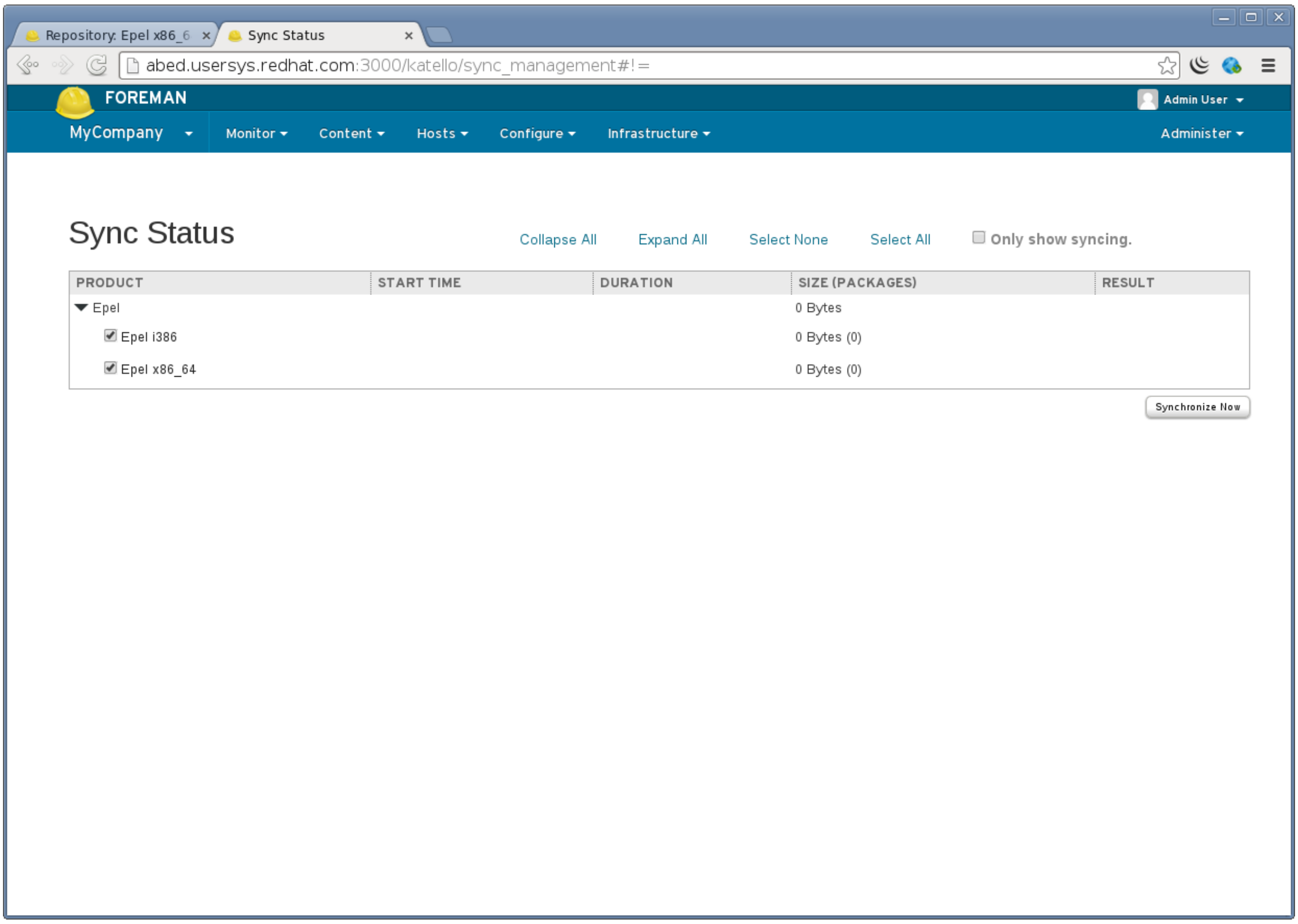
```
New packages: 37/10694 (6.68 MB/9.78 GB)
```

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.



Sync Status

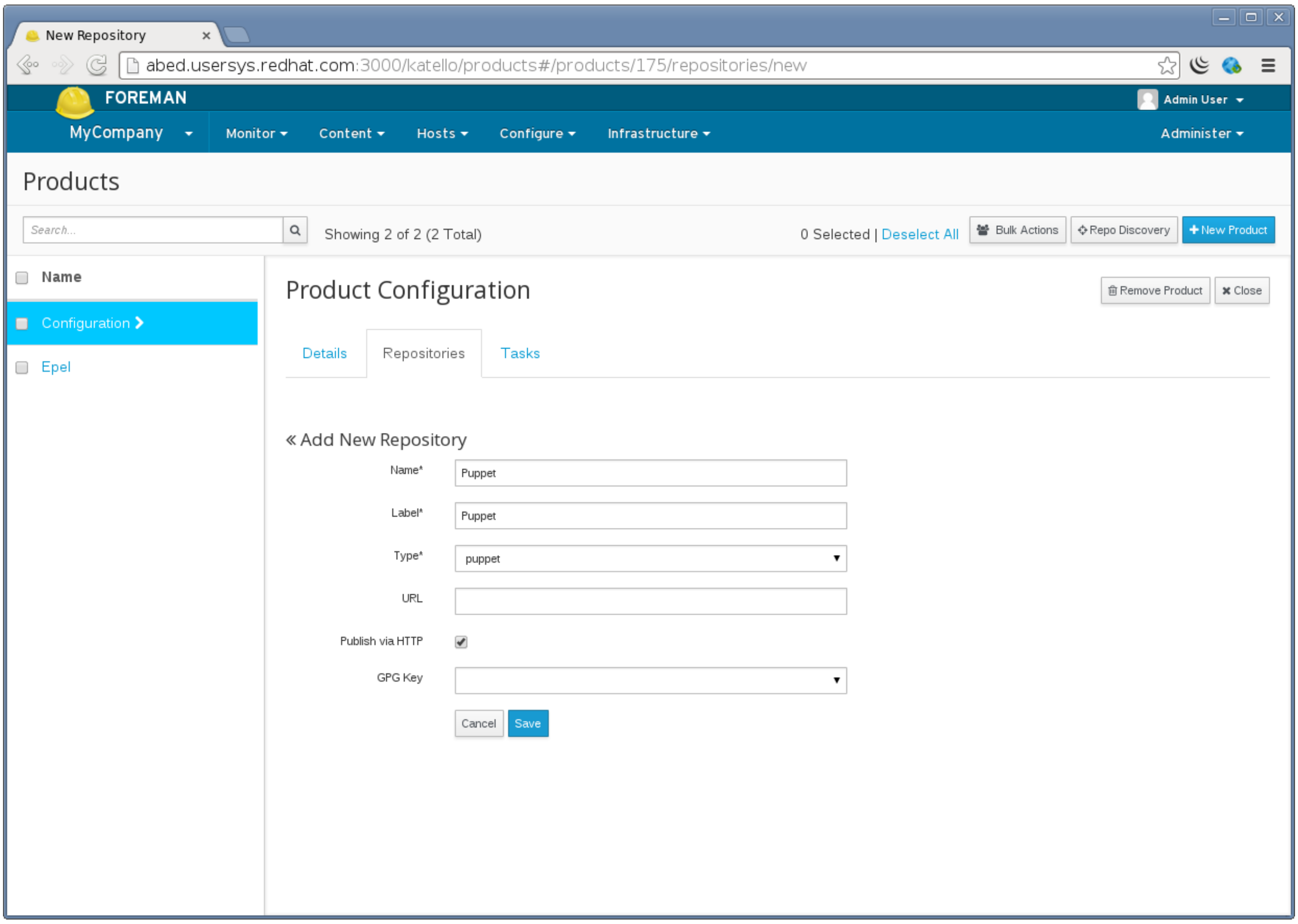
PRODUCT	START TIME	DURATION	SIZE (PACKAGES)	RESULT
Epel			0 Bytes	
☑ Epel i386			0 Bytes (0)	
☑ Epel x86_64			0 Bytes (0)	

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):



Products

Showing 2 of 2 (2 Total)

□ Name
□ Configuration
□ Epel

Product Configuration

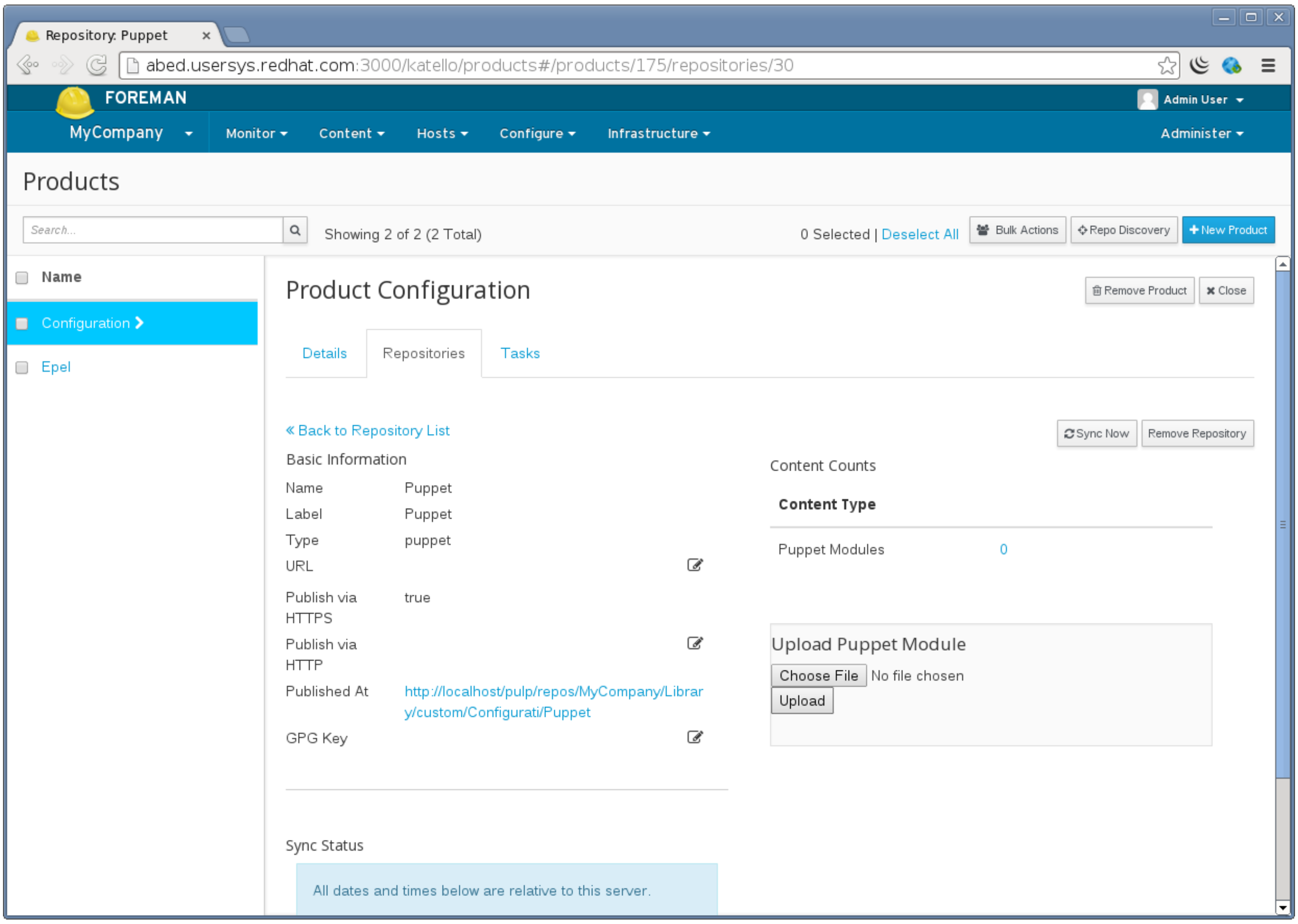
« Add New Repository

Name*	Puppet
Label*	Puppet
Type*	puppet
URL	
Publish via HTTP	☑
GPG Key	

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



Products

Showing 2 of 2 (2 Total)

□ Name
□ Configuration
□ Epel

Product Configuration

Basic Information

Name	Puppet
Label	Puppet
Type	puppet
URL	
Publish via HTTPS	true
Publish via HTTP	
Published At	http://localhost/pulp/repos/MyCompany/Library/custom/Configurati/Puppet
GPG Key	

Content Counts

Content Type	
Puppet Modules	0

Upload Puppet Module

Sync Status

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

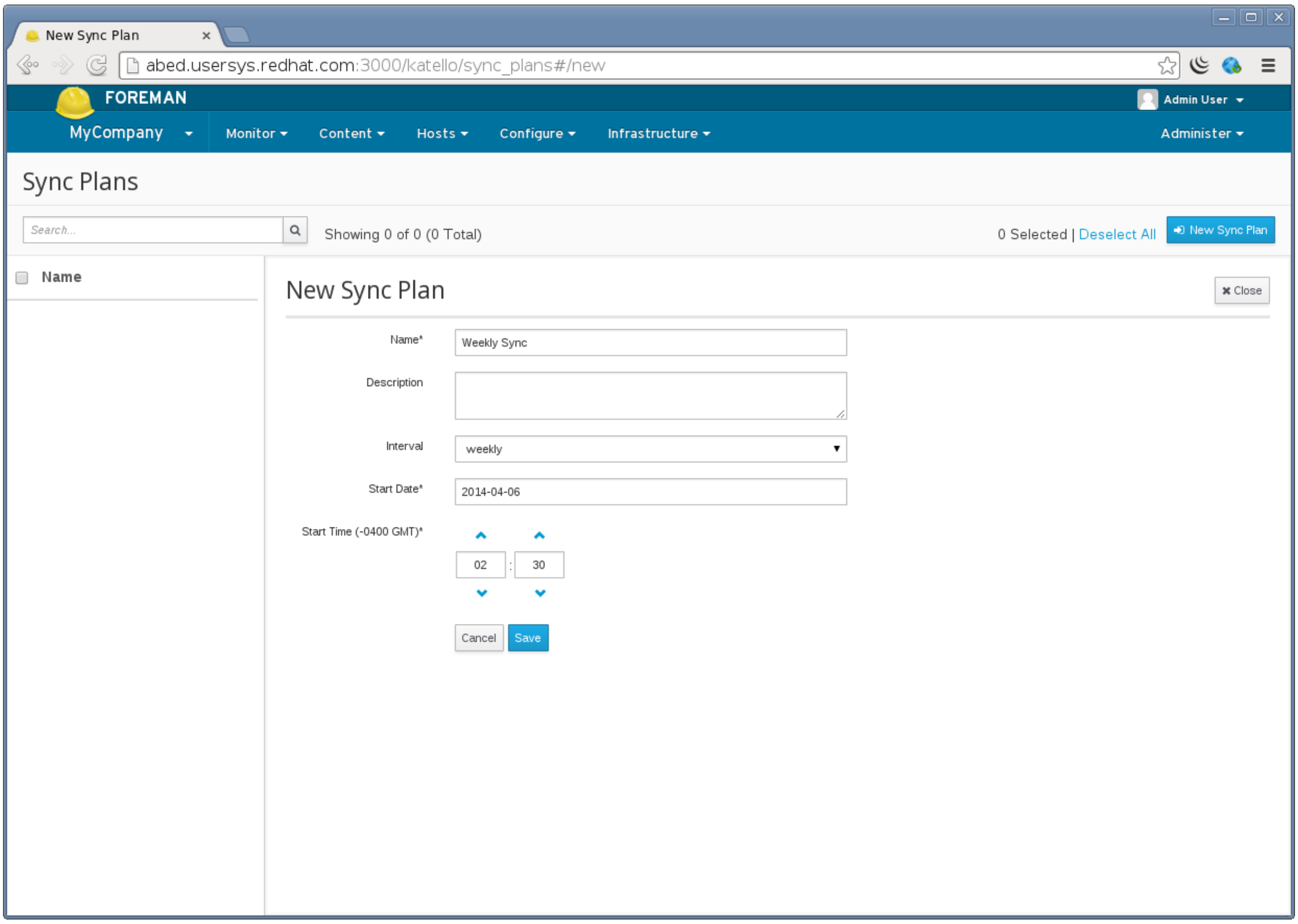
If you would like to schedule certain repositories to sync on a hourly, daily or weekly basis, Sync Plans give you this capability.

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-occuring 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.



abed.usersys.redhat.com:3000/katello/sync_plans#/new

Sync Plans

Showing 0 of 0 (0 Total)

New Sync Plan

Name* Weekly Sync

Description

Interval weekly

Start Date* 2014-04-06

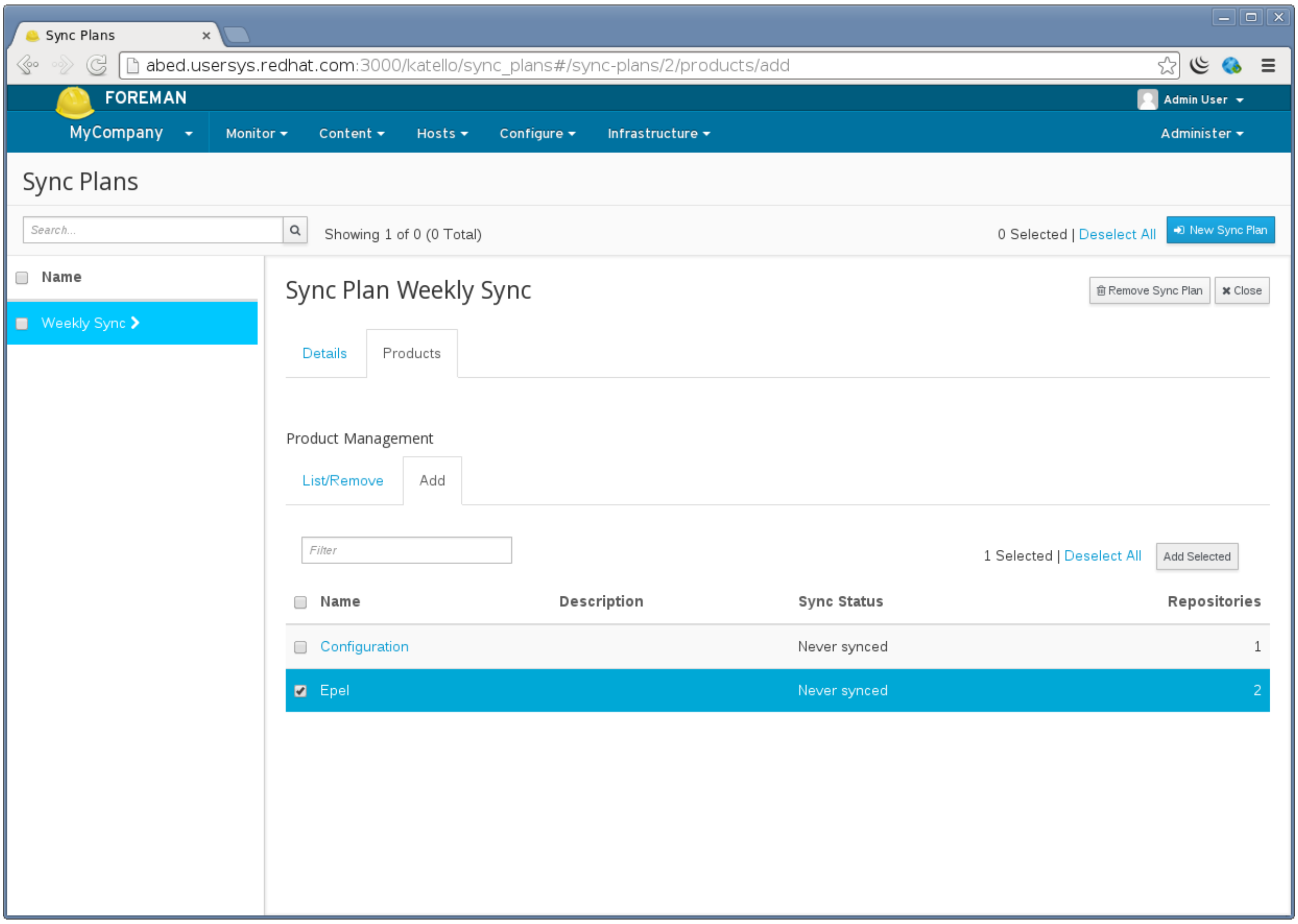
Start Time (-0400 GMT)* 02 : 30

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.



Sync Plans

Showing 1 of 0 (0 Total)

0 Selected | Deselect All

□ Name
□ Weekly Sync

Sync Plan Weekly Sync

Product Management

1 Selected | Deselect All

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

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15 ▾

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
- Stores details about System Purpose

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

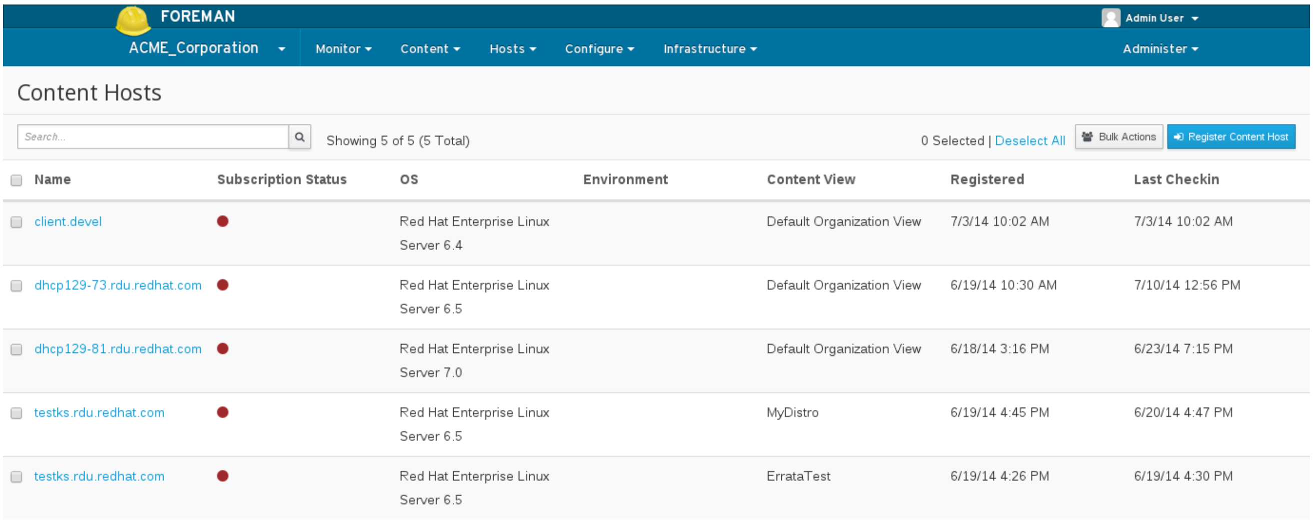
Activation Keys allow you to register and consume content without using a username and password. To create an Activation Key see the Activation Key Guide

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

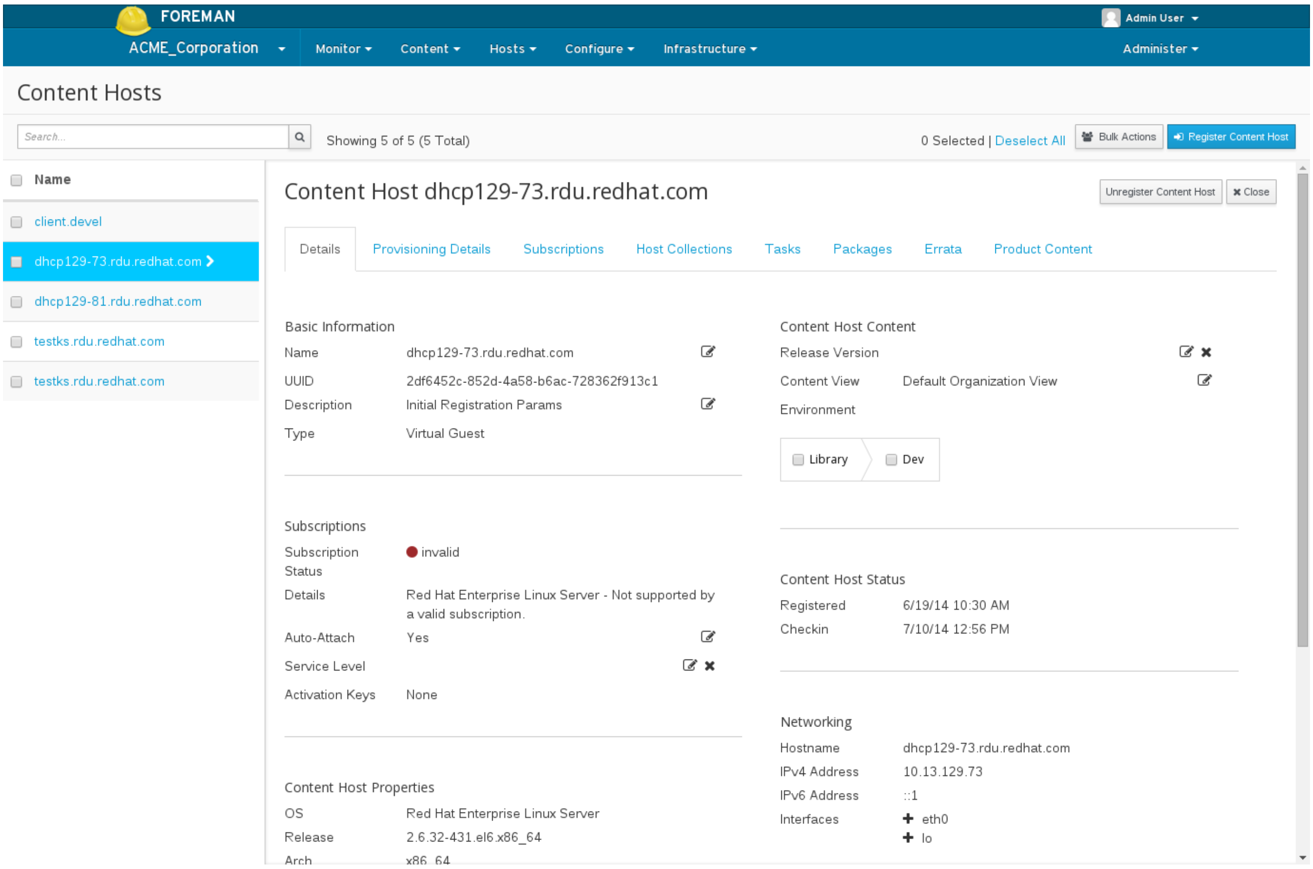
Showing 5 of 5 (5 Total)

0 Selected | Deselect All

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

Showing 5 of 5 (5 Total)

0 Selected | Deselect All

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

Basic Information

Name	dhcp129-73.rdu.redhat.com
UUID	2df6452c-852d-4a58-b6ac-728362f913c1
Description	Initial Registration Params
Type	Virtual Guest

Subscriptions

Subscription Status	● invalid
Details	Red Hat Enterprise Linux Server - Not supported by a valid subscription.
Auto-Attach	Yes
Service Level	
Activation Keys	None

Content Host Properties

OS	Red Hat Enterprise Linux Server
Release	2.6.32-431.el6.x86_64
Arch	x86_64

Content Host Content

Release Version	
Content View	Default Organization View
Environment	□ Library □ Dev

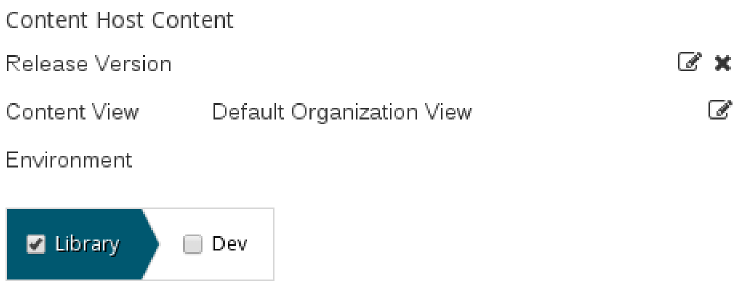
Content Host Status

Registered	6/19/14 10:30 AM
Checkin	7/10/14 12:56 PM

Networking

Hostname	dhcp129-73.rdu.redhat.com
IPv4 Address	10.13.129.73
IPv6 Address	::1
Interfaces	✚ eth0 ✚ lo

Look in the upper right corner for the “Content Host Content”:



Content Host Content

Release Version

Content View   Default Organization View

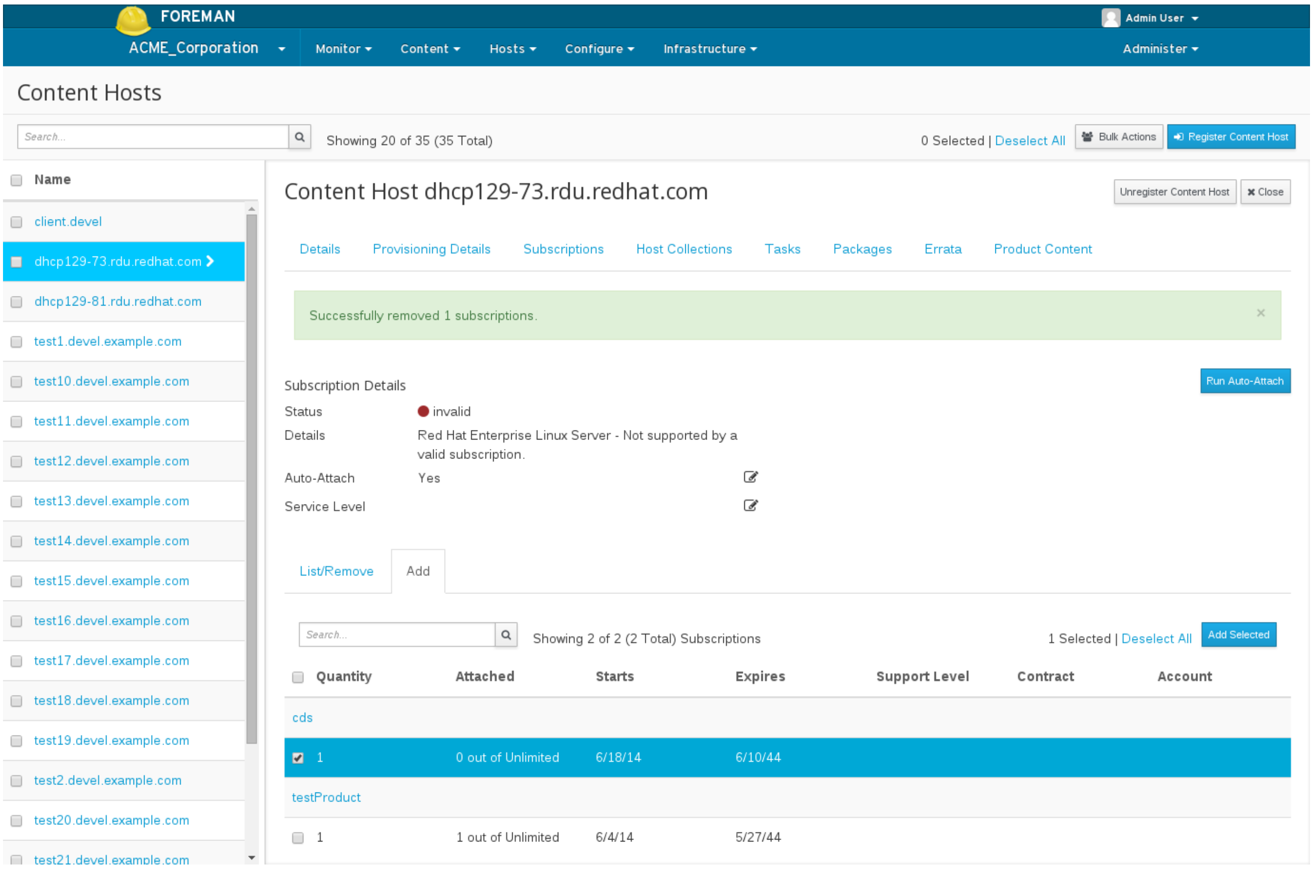
Environment

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



Content Hosts

Showing 20 of 35 (35 Total)

0 Selected | Deselect All

Name
client.devel
dhcp129-73.rdu.redhat.com
dhcp129-81.rdu.redhat.com
test1.devel.example.com
test10.devel.example.com
test11.devel.example.com
test12.devel.example.com
test13.devel.example.com
test14.devel.example.com
test15.devel.example.com
test16.devel.example.com
test17.devel.example.com
test18.devel.example.com
test19.devel.example.com
test2.devel.example.com
test20.devel.example.com
test21.devel.example.com

Content Host dhcp129-73.rdu.redhat.com

Successfully removed 1 subscriptions.

Subscription Details

Status	● invalid
Details	Red Hat Enterprise Linux Server - Not supported by a valid subscription.
Auto-Attach	Yes
Service Level	

Showing 2 of 2 (2 Total) Subscriptions

1 Selected | Deselect All

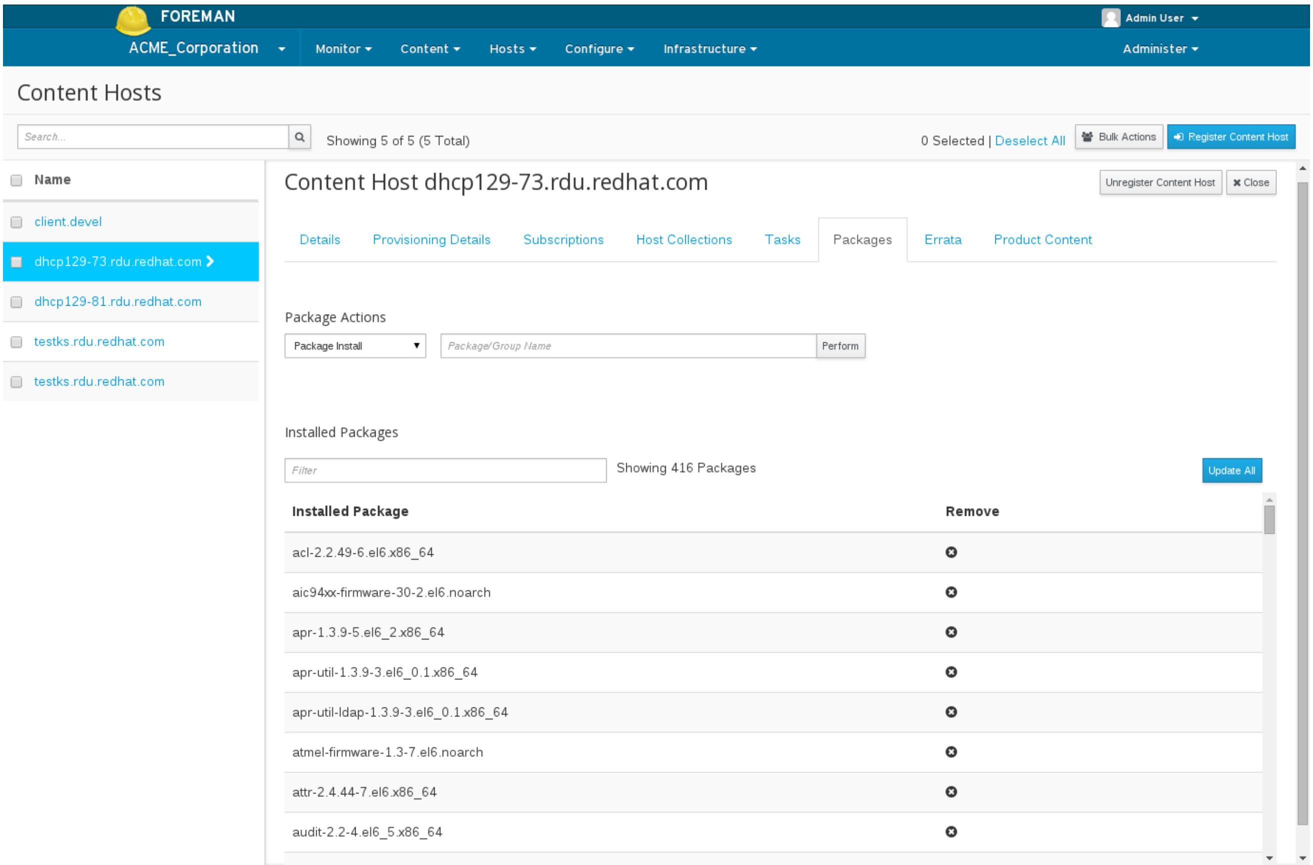
	Quantity	Attached	Starts	Expires	Support Level	Contract	Account
cds							
☑	1	0 out of Unlimited	6/18/14	6/10/44			
testProduct							
□	1	1 out of Unlimited	6/4/14	5/27/44			

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 singleContent Host, navigate to: Hosts > Content Hosts > Select Content Host > Packages



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

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

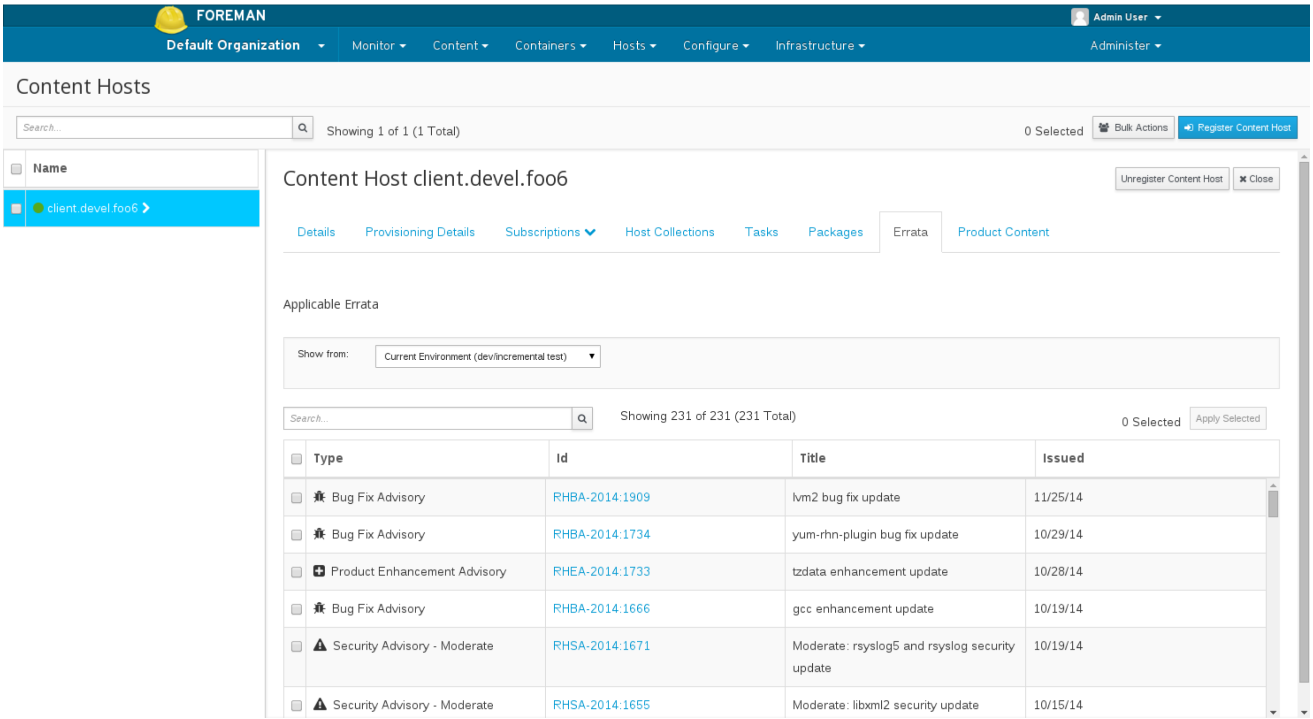
From here you can:

- 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



□	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

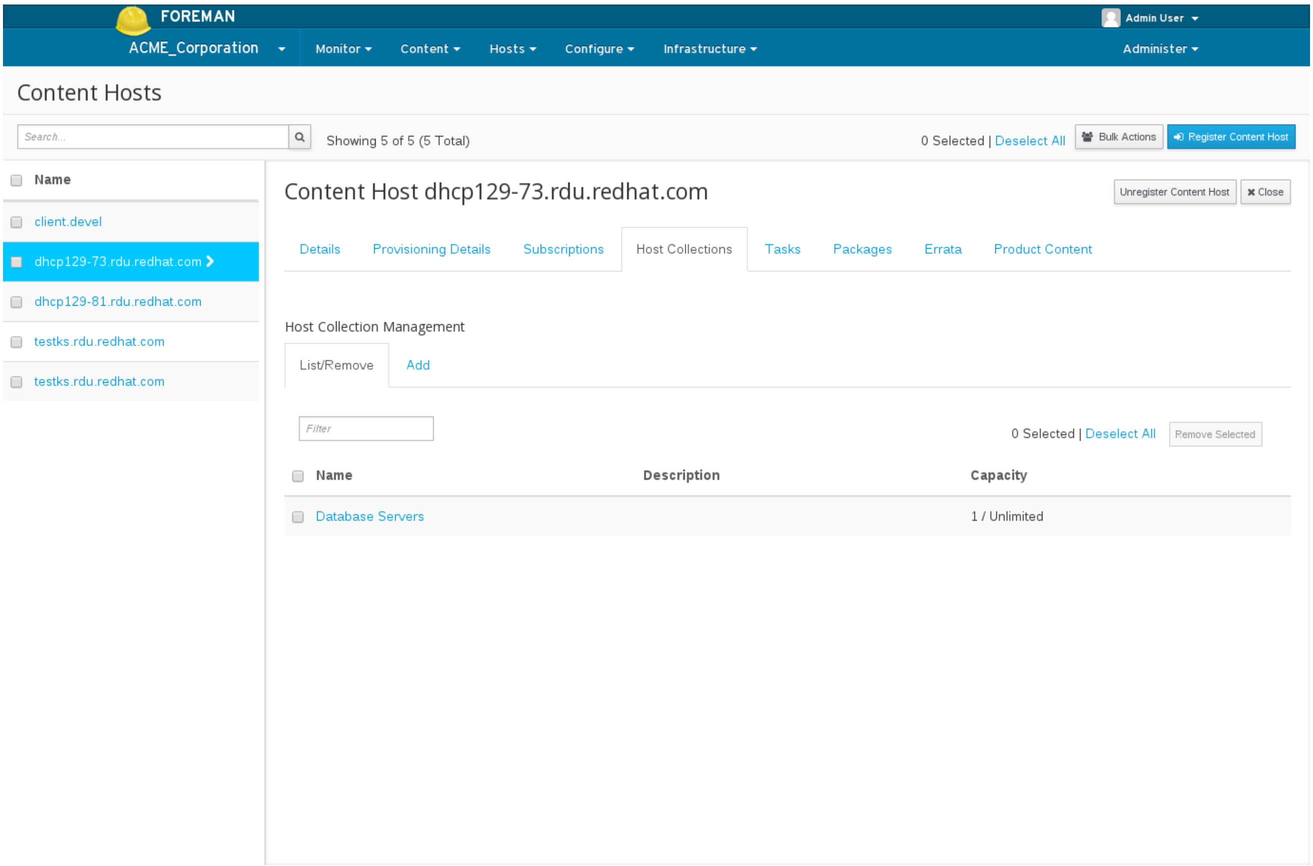
To apply errata, search for the errata you want and select the checkbox beside each errata. Then click “Apply Selected” at the top right.

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



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

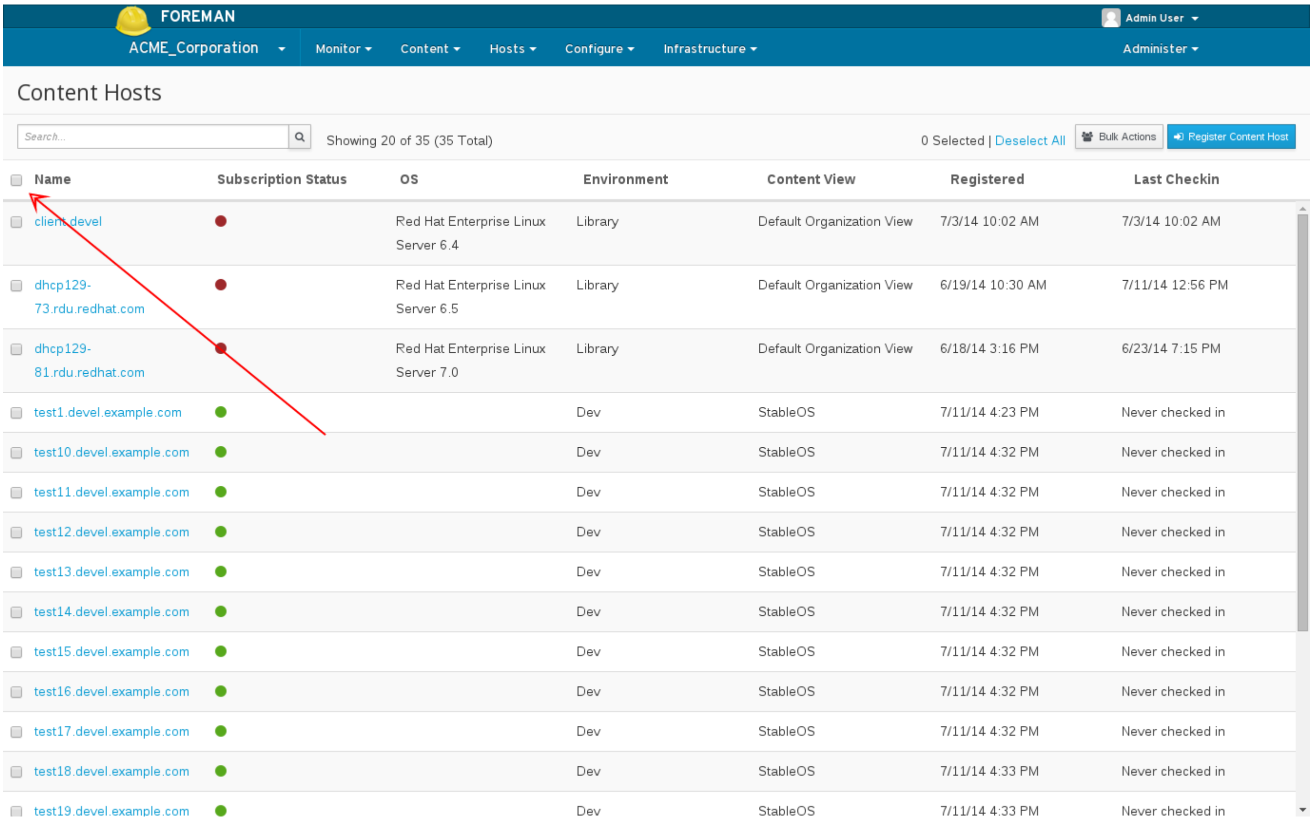
Name	Description	Capacity
Database Servers		1 / Unlimited

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:



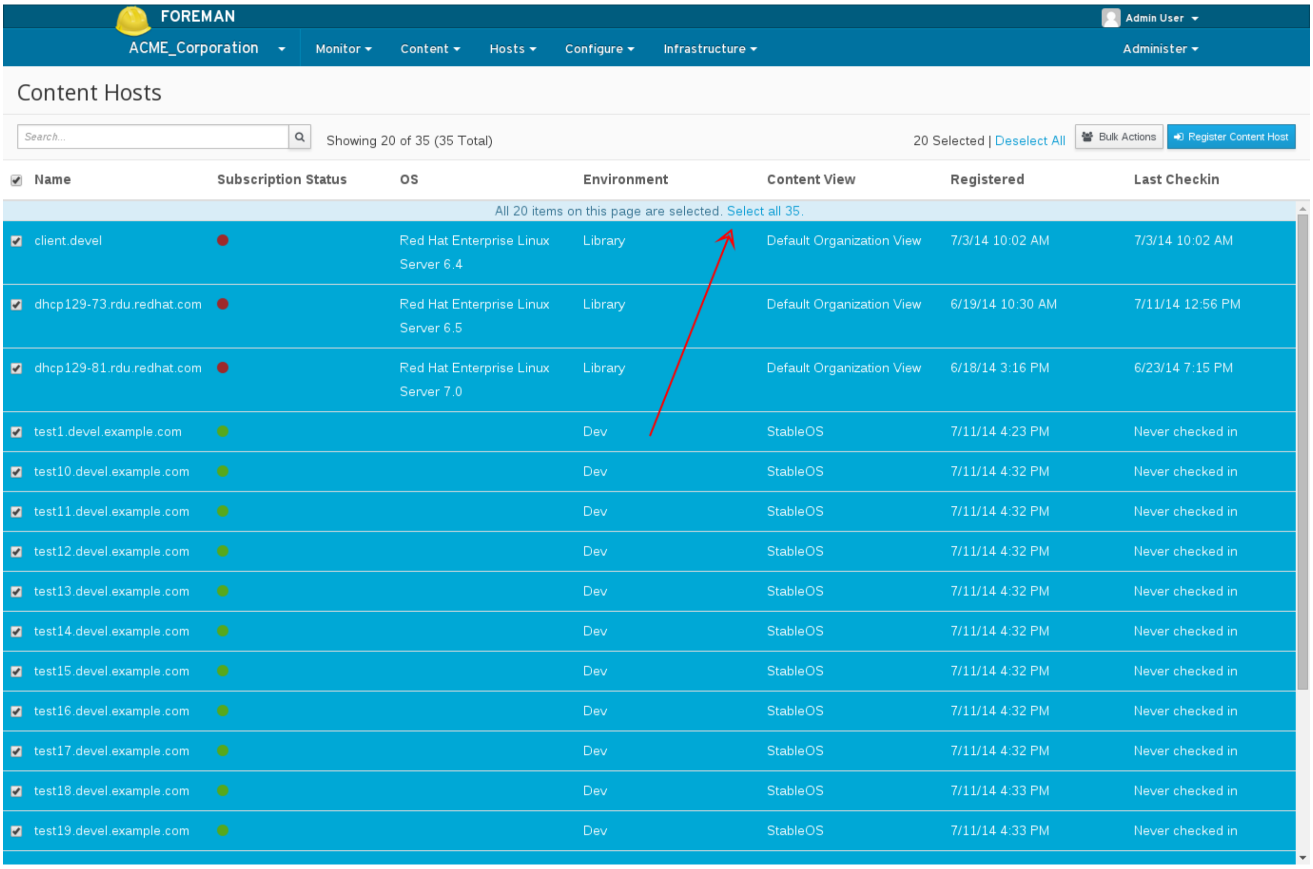
Content Hosts

Showing 20 of 35 (35 Total)

0 Selected | Deselect All

□	Name	Subscription Status	OS	Environment	Content View	Registered	Last Checkin
□	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
□	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
□	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
□	test1.devel.example.com	🟢		Dev	StableOS	7/11/14 4:23 PM	Never checked in
□	test10.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test11.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test12.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test13.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test14.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test15.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test16.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test17.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
□	test18.devel.example.com	🟢		Dev	StableOS	7/11/14 4:33 PM	Never checked in
□	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:



Content Hosts

Showing 20 of 35 (35 Total)

20 Selected | Deselect All

All 20 items on this page are selected. Select all 35.

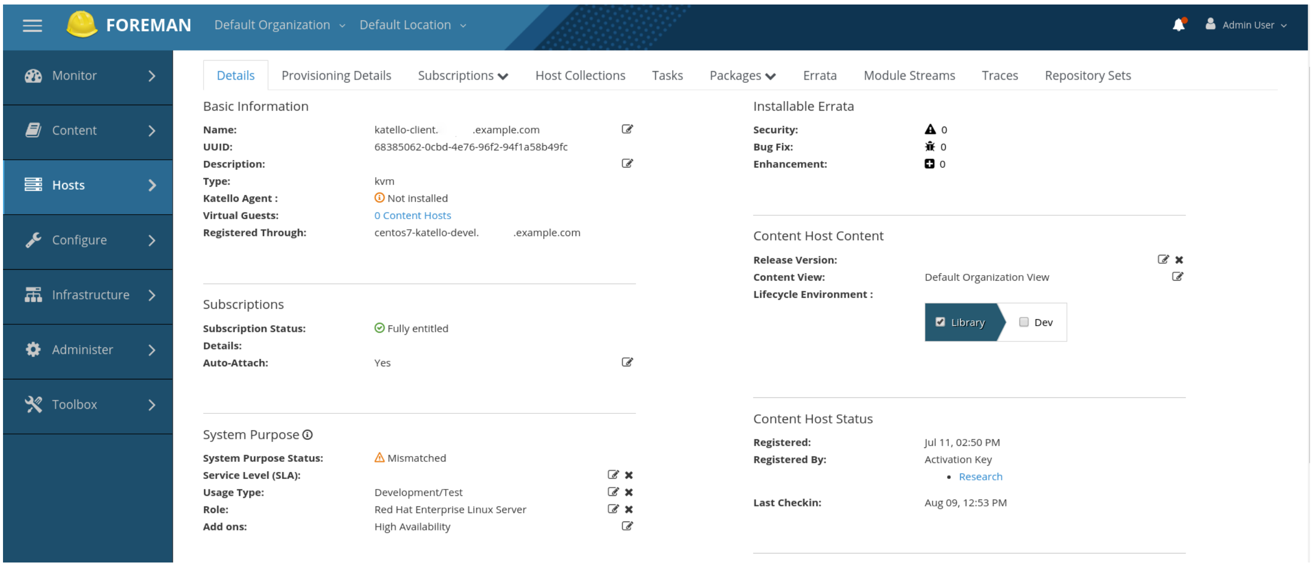
☑	Name	Subscription Status	OS	Environment	Content View	Registered	Last Checkin
☑	client.devel	🔴	Red Hat Enterprise Linux Server 6.4	Library	Default Organization View	7/3/14 10:02 AM	7/3/14 10:02 AM
☑	dhcp129-73.rdu.redhat.com	🔴	Red Hat Enterprise Linux Server 6.5	Library	Default Organization View	6/19/14 10:30 AM	7/11/14 12:56 PM
☑	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
☑	test1.devel.example.com	🟢		Dev	StableOS	7/11/14 4:23 PM	Never checked in
☑	test10.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test11.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test12.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test13.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test14.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test15.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test16.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test17.devel.example.com	🟢		Dev	StableOS	7/11/14 4:32 PM	Never checked in
☑	test18.devel.example.com	🟢		Dev	StableOS	7/11/14 4:33 PM	Never checked in
☑	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.

Set System Purpose attributes

To set System Purpose attributes for a host, Navigate to the Content Host Details page, Host > Content Hosts > Click the name of the desired Content Host. You can edit Service Level, Role, Usage Type, and Add-ons from the System Purpose section.



Basic Information

Name:	katello-client. .example.com
UUID:	68385062-0cbd-4e76-96f2-94f1a58b49fc
Description:	
Type:	kvm
Katello Agent :	Not installed
Virtual Guests:	0 Content Hosts
Registered Through:	centos7-katello-devel. .example.com

Subscriptions

Subscription Status:	Fully entitled
Details:	
Auto-Attach:	Yes

System Purpose

System Purpose Status:	Mismatched
Service Level (SLA):	
Usage Type:	Development/Test
Role:	Red Hat Enterprise Linux Server
Add ons:	High Availability

Installable Errata

Security:	0
Bug Fix:	0
Enhancement:	0

Content Host Content

Release Version:	
Content View:	Default Organization View
Lifecycle Environment :	☑ Library ☐ Dev

Content Host Status

Registered:	Jul 11, 02:50 PM
Registered By:	Activation Key - Research
Last Checkin:	Aug 09, 12:53 PM

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15 ▾

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 snapshoting:

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.

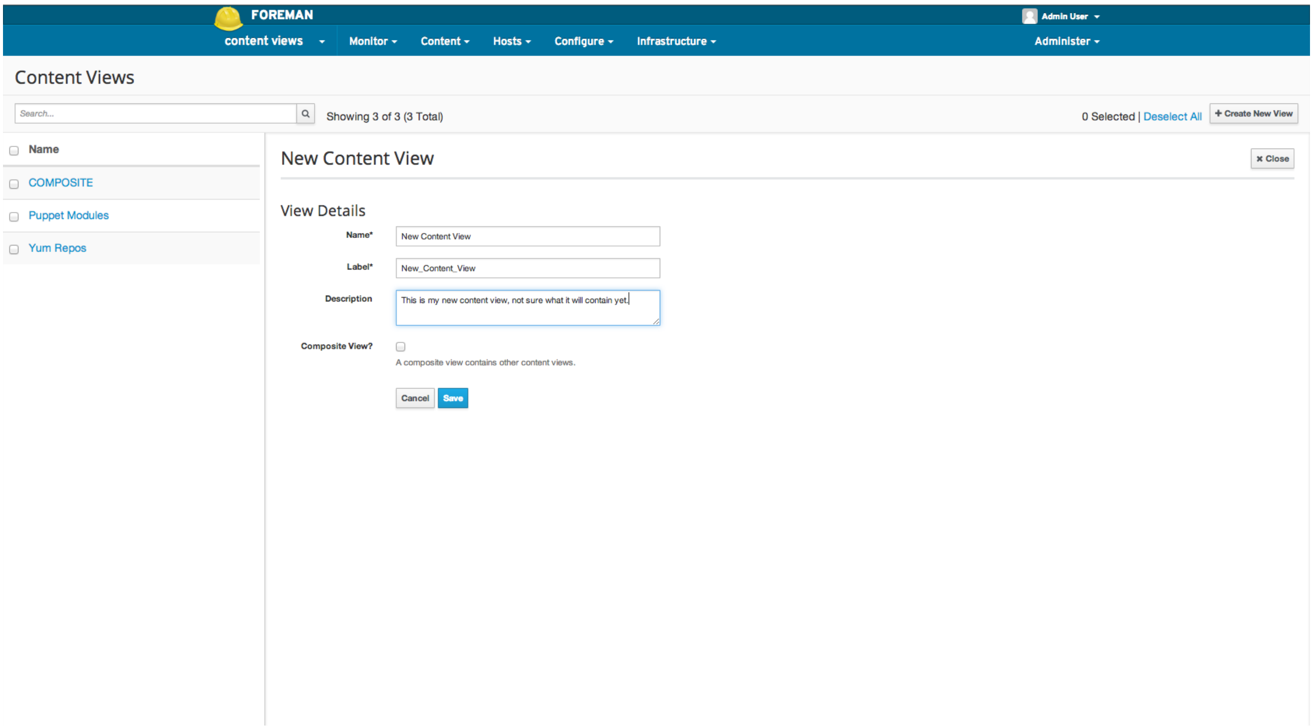
Note that all of the actions below can also done with hammer, the CLI tool, and examples are given at the end of each section.

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.



New Content View

View Details

Name*: New Content View

Label*: New_Content_View

Description: This is my new content view, not sure what it will contain yet.

Composite View? □

A composite view contains other content views.

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:

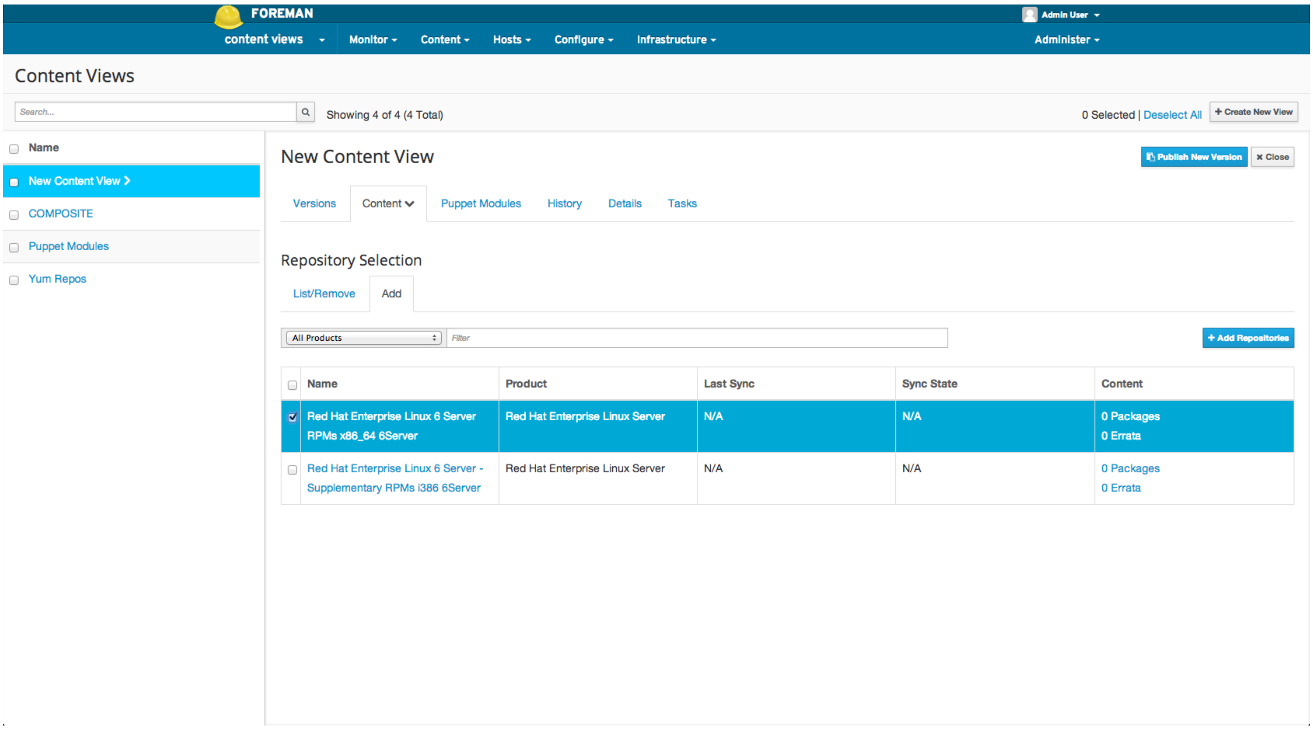
```
hammer content-view create \
  --organization="Default Organization" \
  --name="New Composite Content View" \
  --description="This is my new composite content view." \
  --composite
```

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:

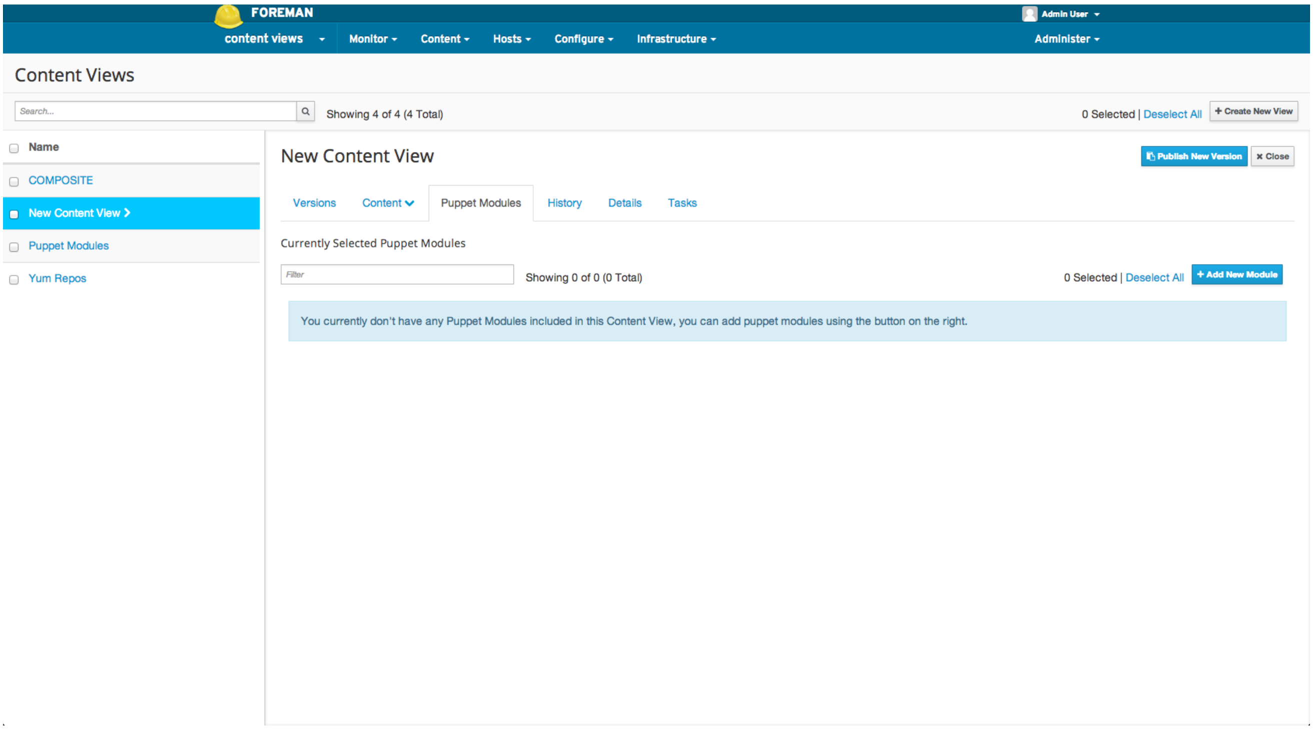
```
hammer content-view add-repository \
  --organization="Default Organization"
  --name="New Content View" \
  --repository="CentOS 6.5"
```

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)



Content Views

Showing 4 of 4 (4 Total)

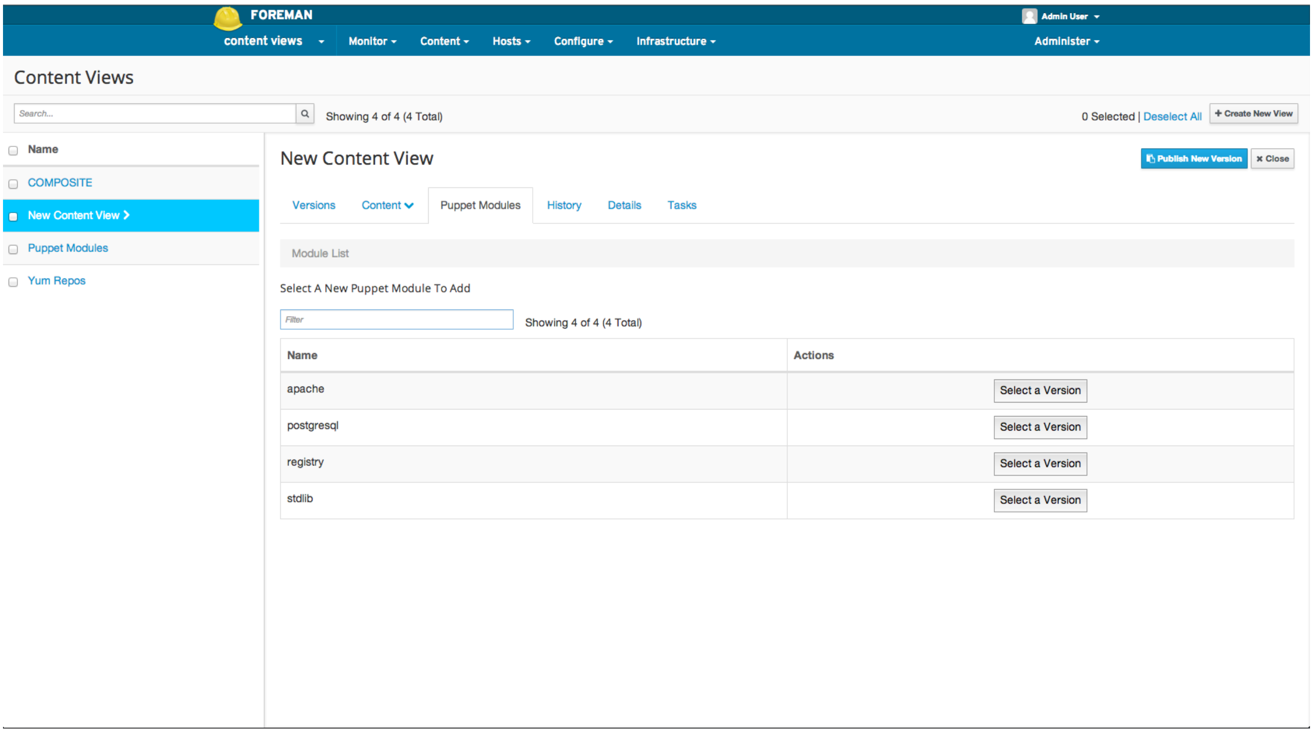
□ Name
□ COMPOSITE
■ New Content View ›
□ Puppet Modules
□ Yum Repos

New Content View

Currently Selected Puppet Modules

Showing 0 of 0 (0 Total)

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



Content Views

Showing 4 of 4 (4 Total)

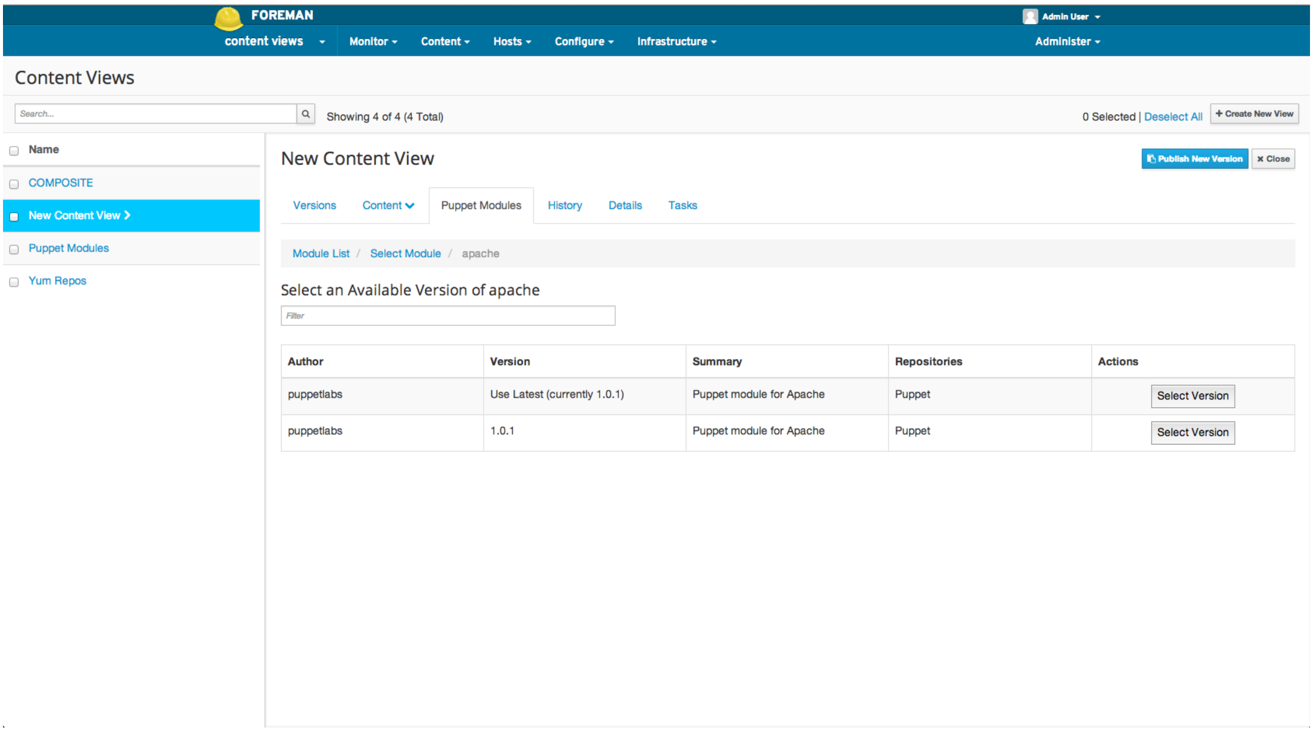
□ Name
□ COMPOSITE
■ New Content View ›
□ Puppet Modules
□ Yum Repos

New Content View

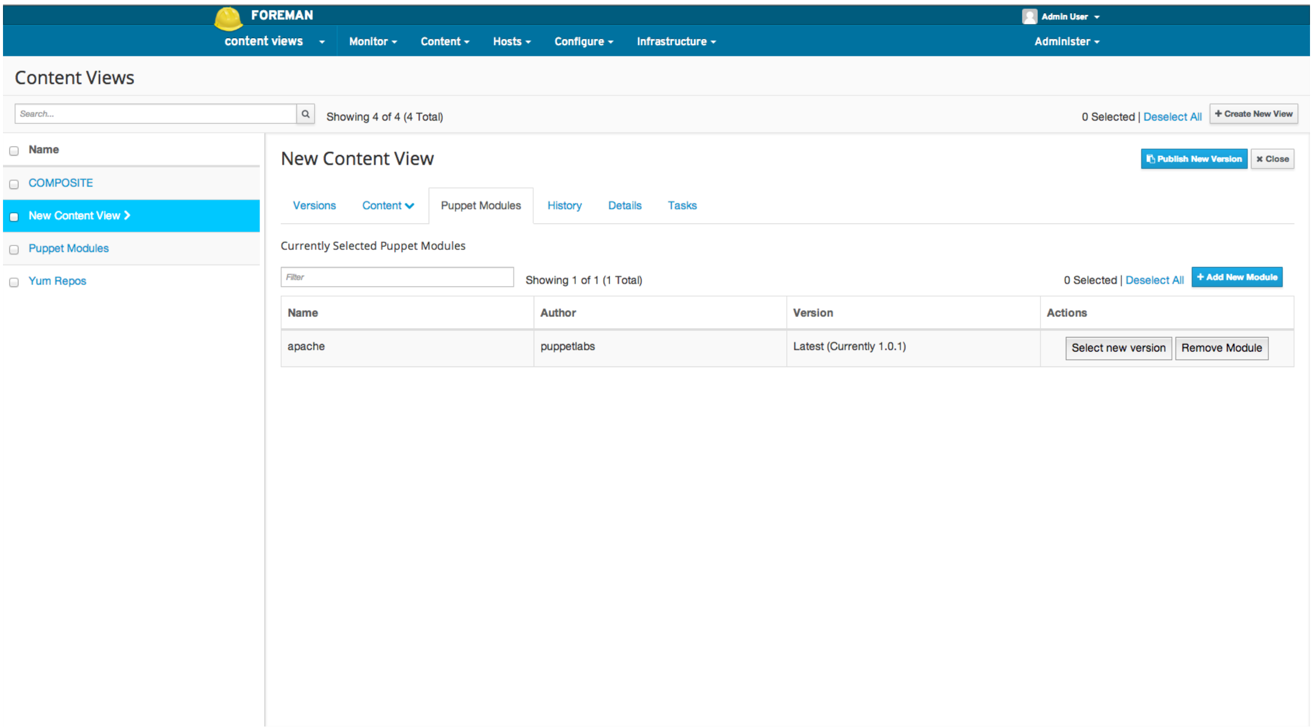
Select A New Puppet Module To Add

Showing 4 of 4 (4 Total)

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



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



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

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:

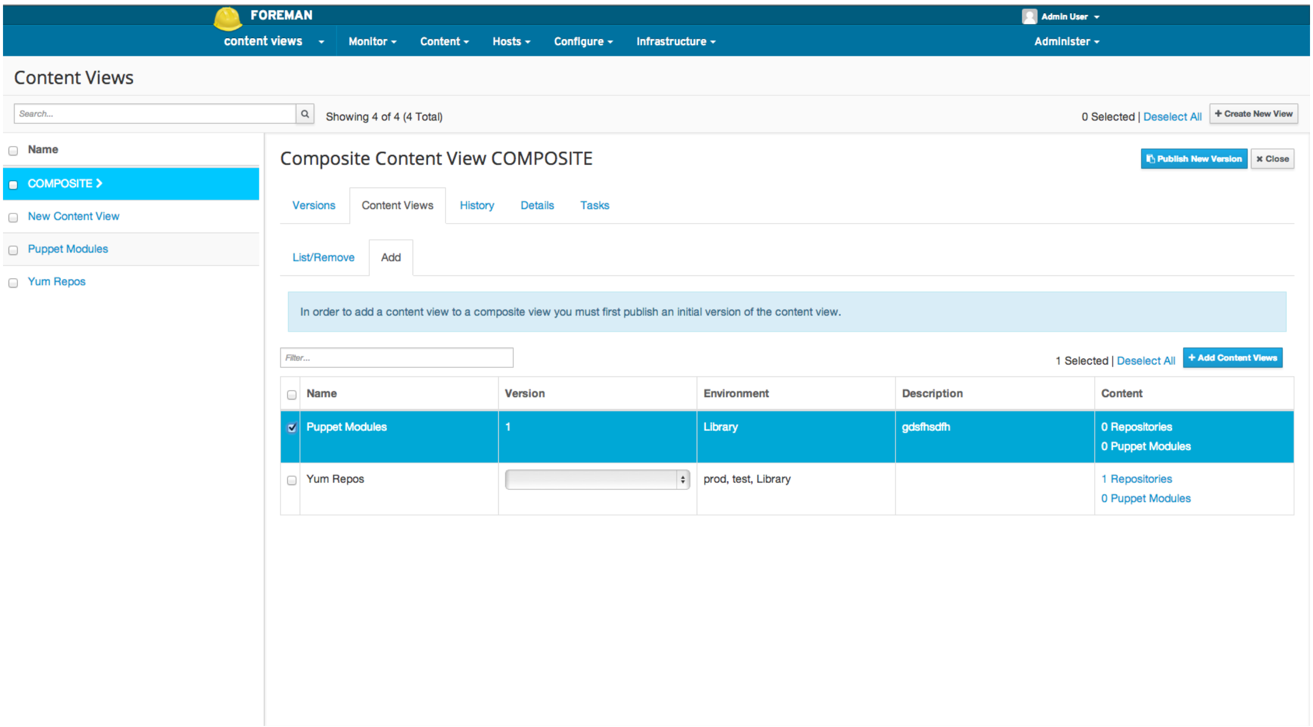
```
hammer content-view puppet-module add \
  --organization="Default Organization" \
  --content-view="New Content View" \
  --uuid=91cc9bb7-dbb3-4798-b50a-45173b763cbb
```

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



Content Views

Showing 4 of 4 (4 Total)

0 Selected | Deselect All

- □ Name
- □ COMPOSITE >
- □ New Content View
- □ Puppet Modules
- □ Yum Repos

Composite Content View COMPOSITE

Versions | Content Views | History | Details | Tasks

List/Remove | Add

In order to add a content view to a composite view you must first publish an initial version of the content view.

1 Selected | Deselect All

□	Name	Version	Environment	Description	Content
☑	Puppet Modules	1	Library	gdsfhsdfh	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:

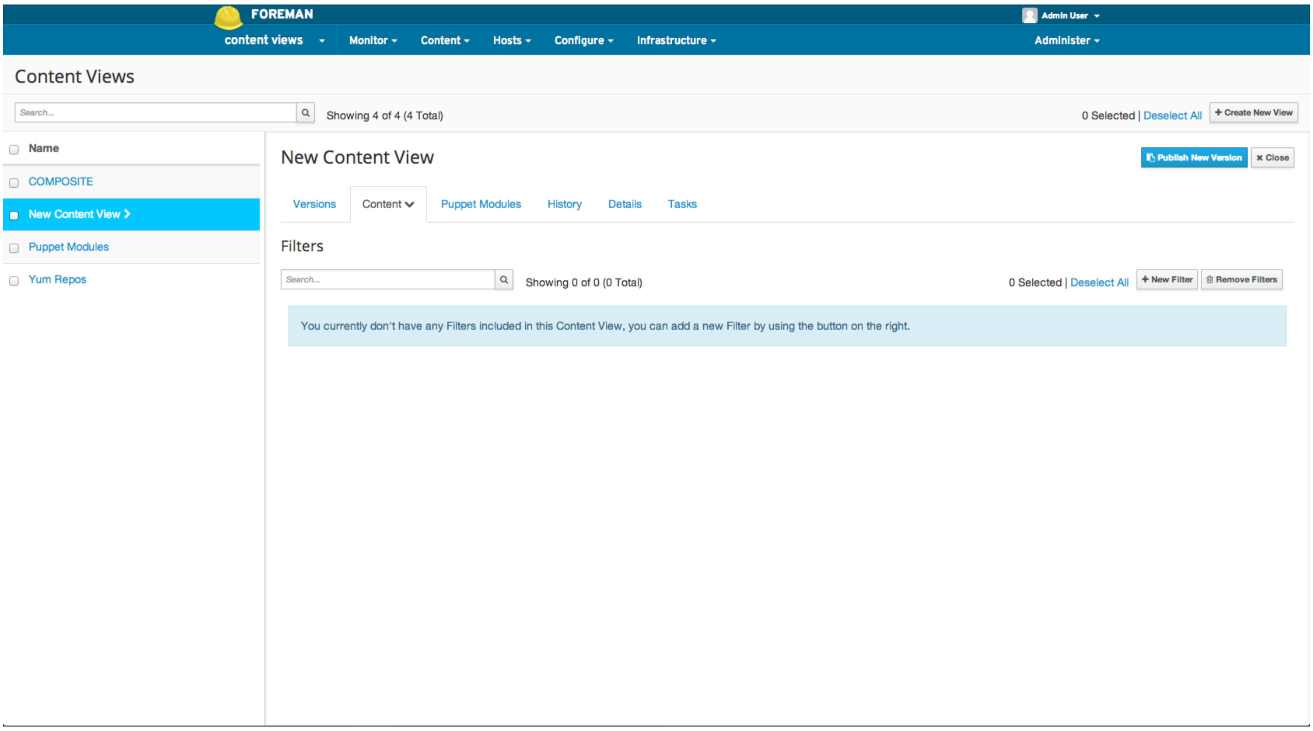
```
hammer content-view update \
  --organization="Default Organization" \
  --content-view="New Composite Content View" \
  --component-ids=2
```

Creating a filter

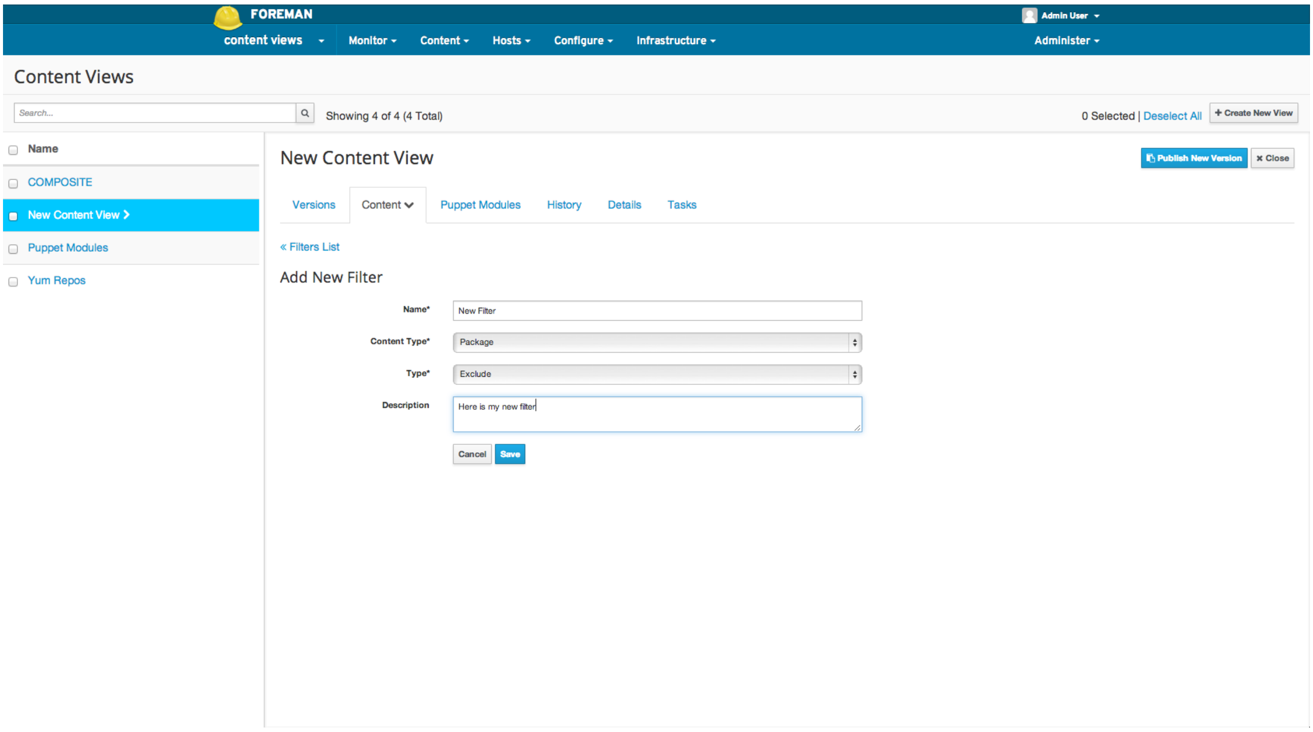
If only using Content Views as snapshots, Filters are unnecessary. If the desire is to filter what content make it into the view, such as blacklisting a package by name or version, or blacklisting errata by date or type, Filters can help accomplish these tasks.

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



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.

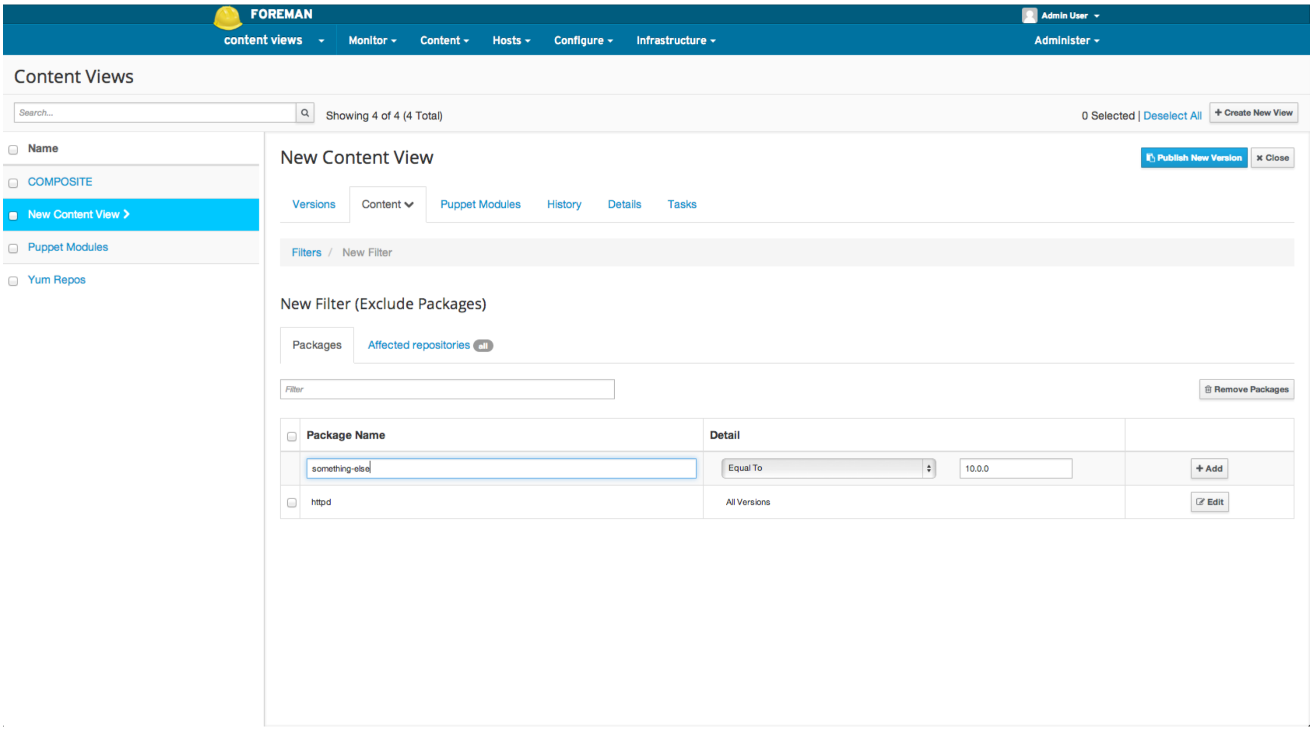


Name*: New Filter

Content Type*: Package

Type*: Exclude

Description: Here is my new filter



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

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:

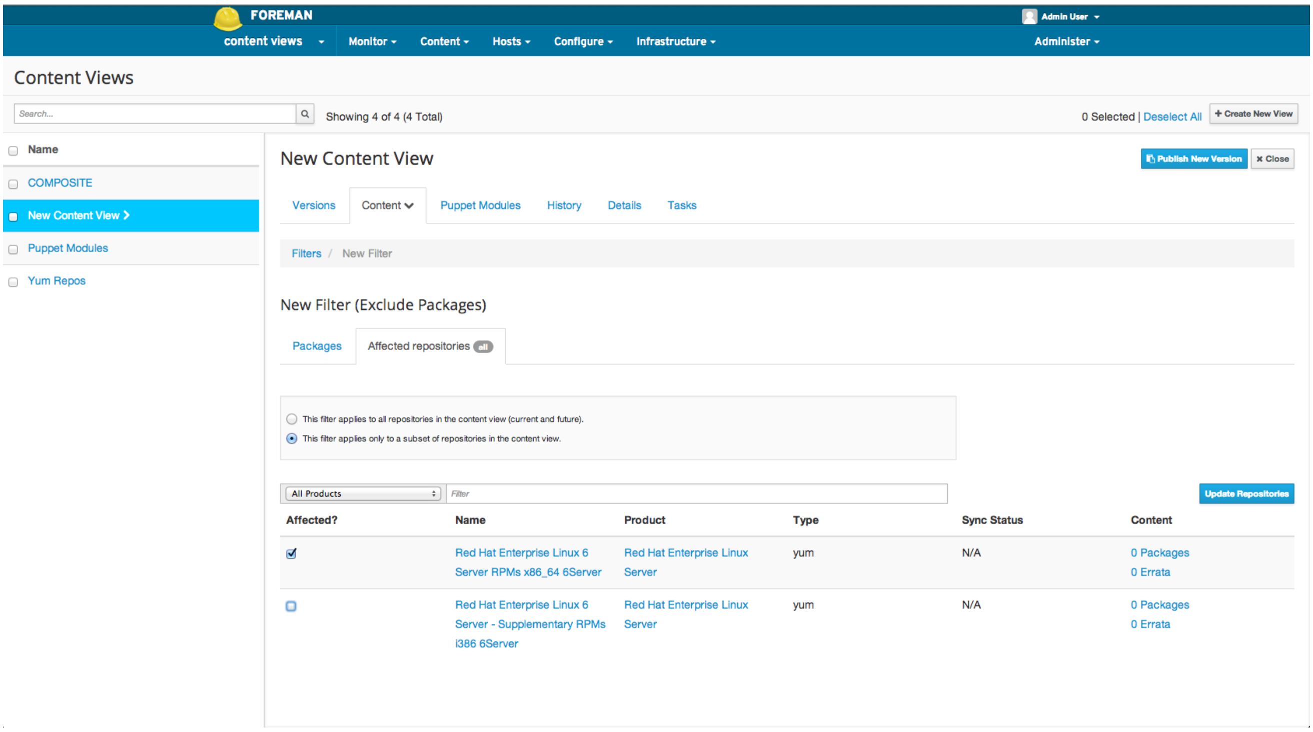
```
hammer content-view filter rule create \
  --organization="Default Organization" \
  --content-view="New Content View" \
  --content-view-filter="New Filter" \
  --name="something-else" \
  --max-version="10.0.0" \
  --min-version="10.0.0"
```

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)



Content Views

Showing 4 of 4 (4 Total) 0 Selected | Deselect All

- Name
- COMPOSITE
- New Content View
- Puppet Modules
- Yum Repos

New Content View

Filters / New Filter

New Filter (Exclude Packages)

Affected repositories all

○ 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.

All Products | Filter

Affected?	Name	Product	Type	Sync Status	Content
☑	Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server	Red Hat Enterprise Linux Server	yum	N/A	0 Packages 0 Errata
□	Red Hat Enterprise Linux 6 Server - Supplementary RPMs i386 6Server	Red Hat Enterprise Linux Server	yum	N/A	0 Packages 0 Errata

From the CLI, adding a Content View Filter:

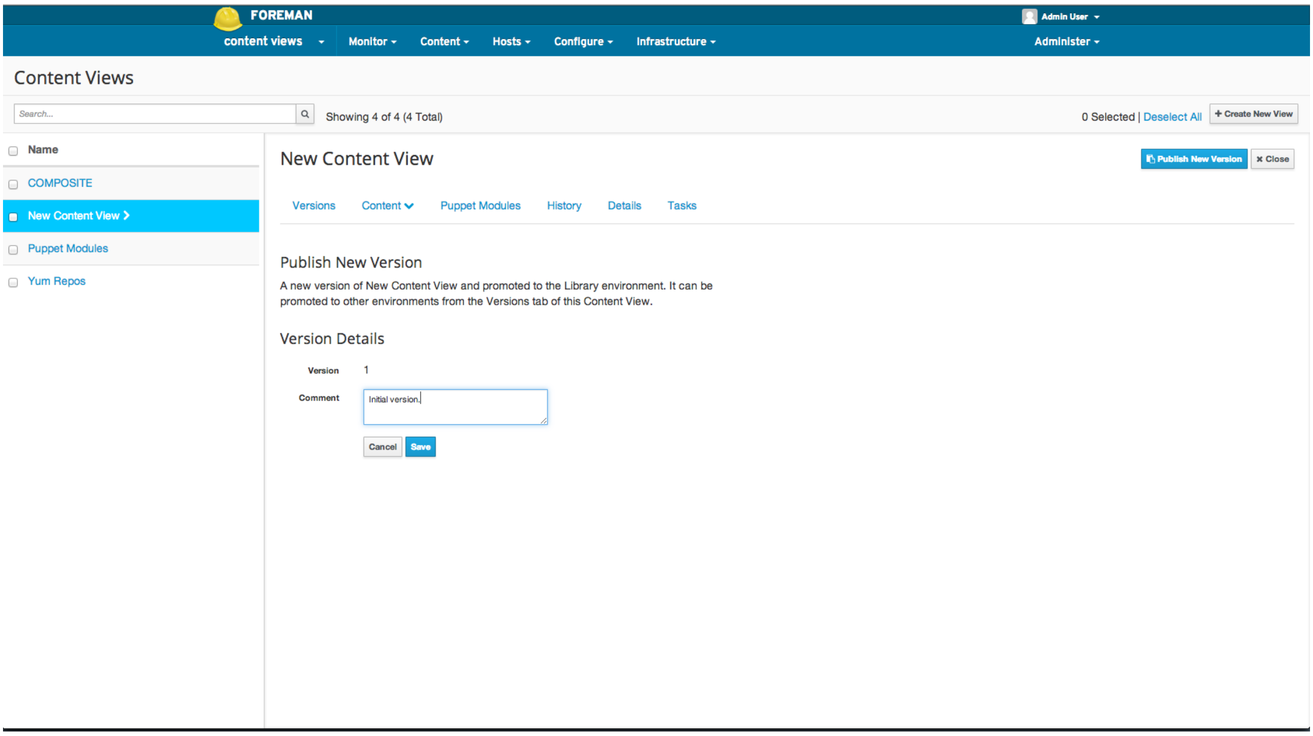
```
hammer content-view filter update \
  --organization="Default Organization" \
  --name="New Filter" \
  --repository-ids=2,3,7
```

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



Content Views

Showing 4 of 4 (4 Total) 0 Selected | Deselect All

- Name
- COMPOSITE
- New Content View
- Puppet Modules
- Yum Repos

New Content View

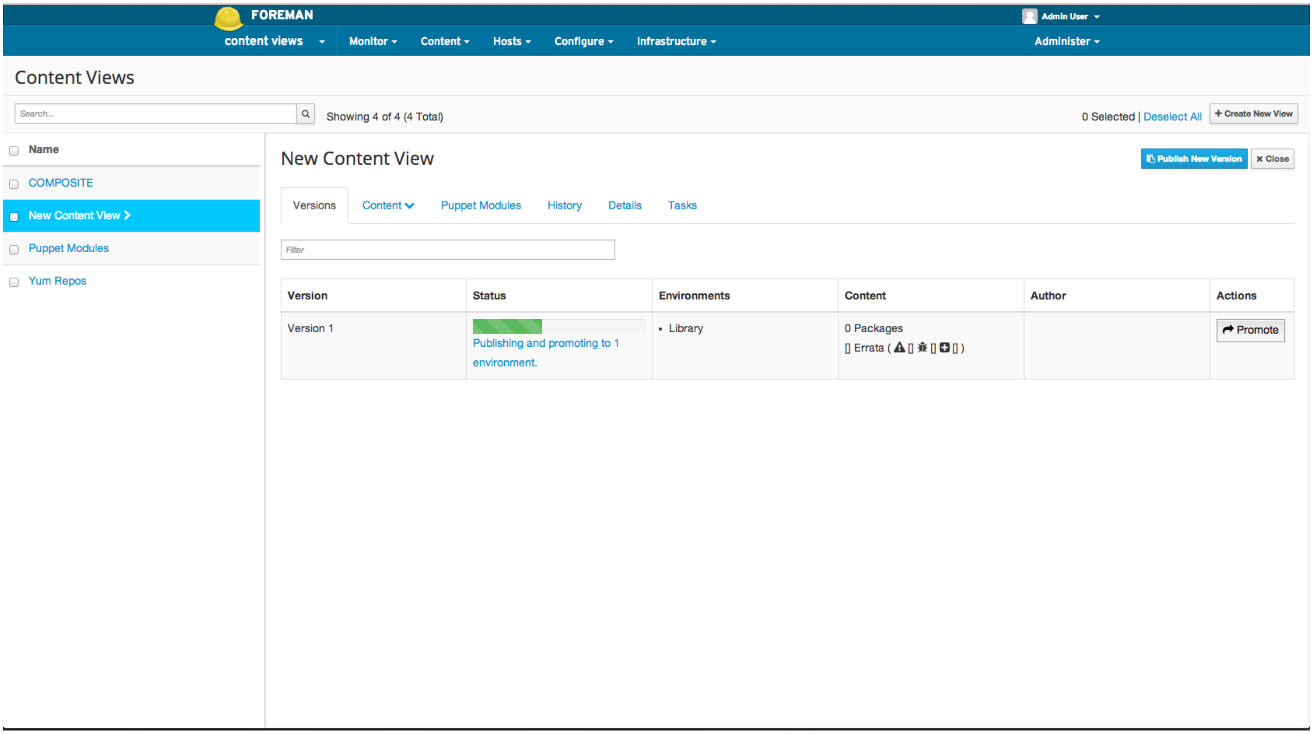
Publish New Version

A new version of New Content View and 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.



Content Views

Showing 4 of 4 (4 Total)

0 Selected | Deselect All

- □ Name
- □ COMPOSITE
- □ New Content View
- □ Puppet Modules
- □ Yum Repos

New Content View

Version	Status	Environments	Content	Author	Actions
Version 1	Publishing and promoting to 1 environment.	• Library	0 Packages [] Errata ([] [] [])		Promote

From the CLI:

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

Registering a Content Host

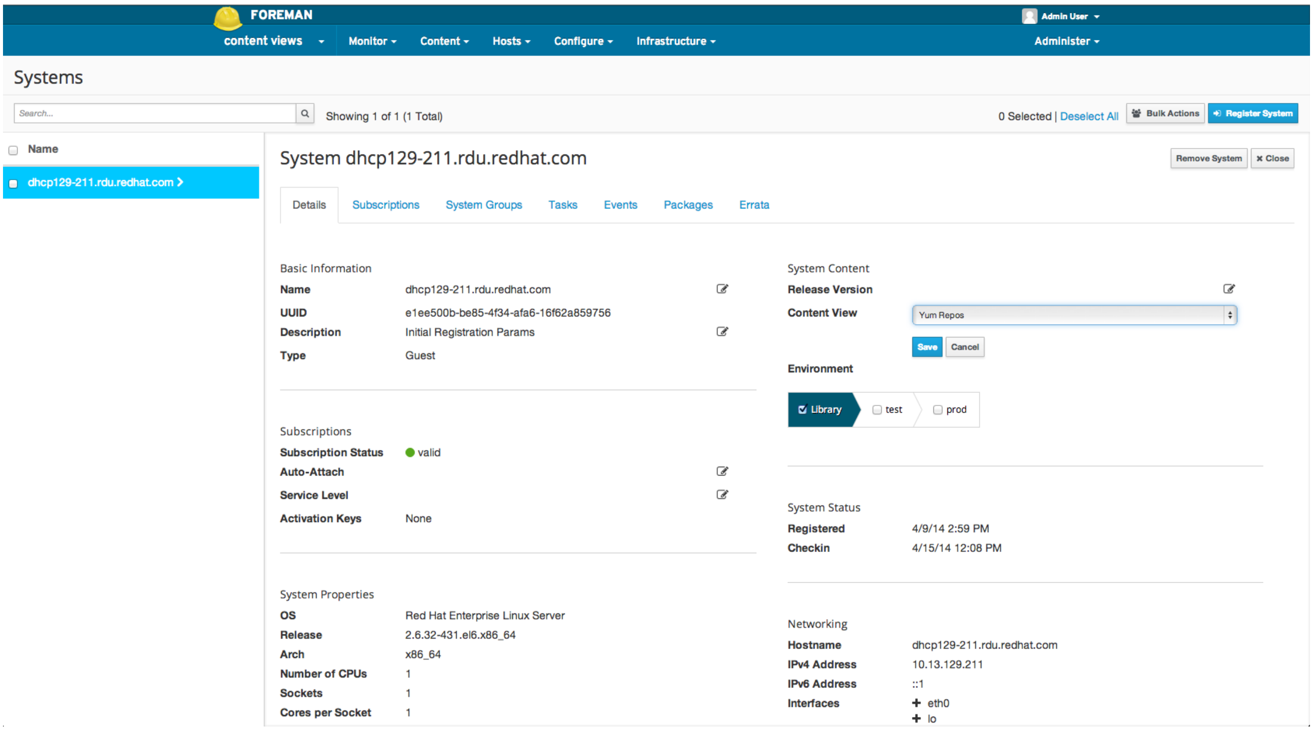
To register a Content Host that is not currently registered to the Content View, simply use subscription manager on the client Content Host and run:

```
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)

Name

dhcp129-211.rdu.redhat.com

System dhcp129-211.rdu.redhat.com

Basic Information

Name dhcp129-211.rdu.redhat.com
UUID e1ee500b-be85-4f34-afa6-16f62a859756
Description Initial Registration Params
Type Guest

Subscriptions

Subscription Status ● valid
Auto-Attach
Service Level
Activation Keys None

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

System Content

Release Version
Content View Yum Repos

Environment

☑ Library ☐ test ☐ prod

System Status

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

Networking

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

From the CLI:

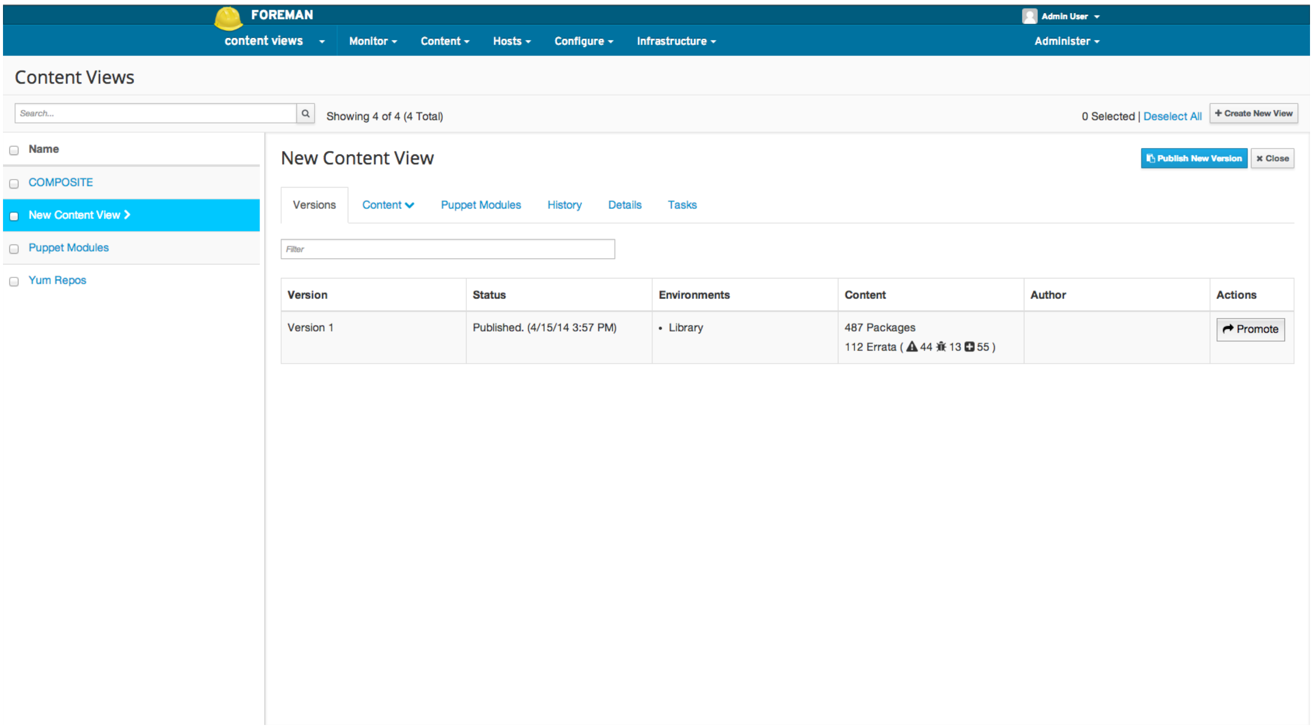
```
hammer content-host update \
  --organization="Default Organization" \
  --name="dhcp129-211.rdu.redhat.com" \
  --content-view="New Content View" \
  --lifecycle-environment="Library"
```

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



Content Views

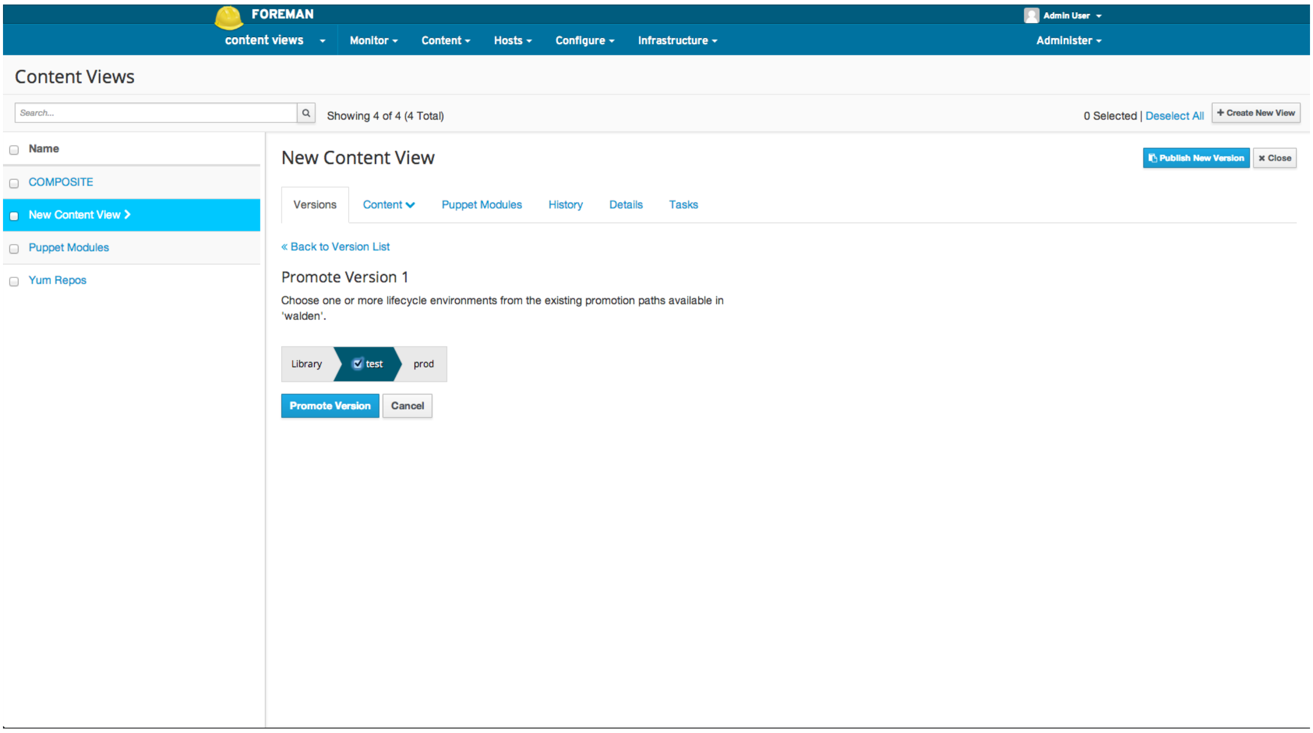
Showing 4 of 4 (4 Total)

0 Selected | Deselect All

□ Name
□ COMPOSITE
□ New Content View
□ Puppet Modules
□ Yum Repos

New Content View

Version	Status	Environments	Content	Author	Actions
Version 1	Published. (4/15/14 3:57 PM)	Library	487 Packages 112 Errata (44 13 55)		Promote



Content Views

Showing 4 of 4 (4 Total)

0 Selected | Deselect All

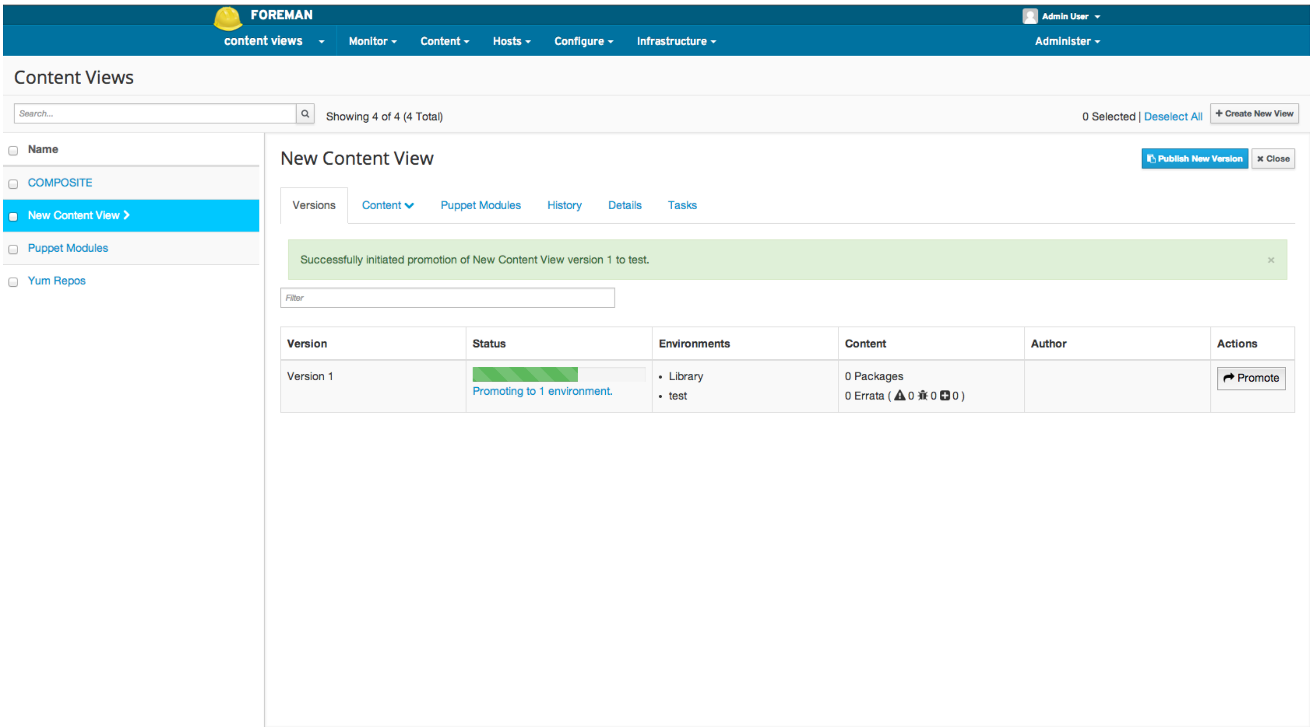
□ Name
□ COMPOSITE
□ New Content View
□ Puppet Modules
□ Yum Repos

New Content View

« Back to Version List

Promote Version 1

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



Successfully initiated promotion of New Content View version 1 to test.

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
```

Katello 3.15 Documentation

3.15 ▾

Katello Inter-Server Sync

NOTE: This feature has been deprecated and will be removed in a future release of Katello. Please consider using Content View Import/Export instead.

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

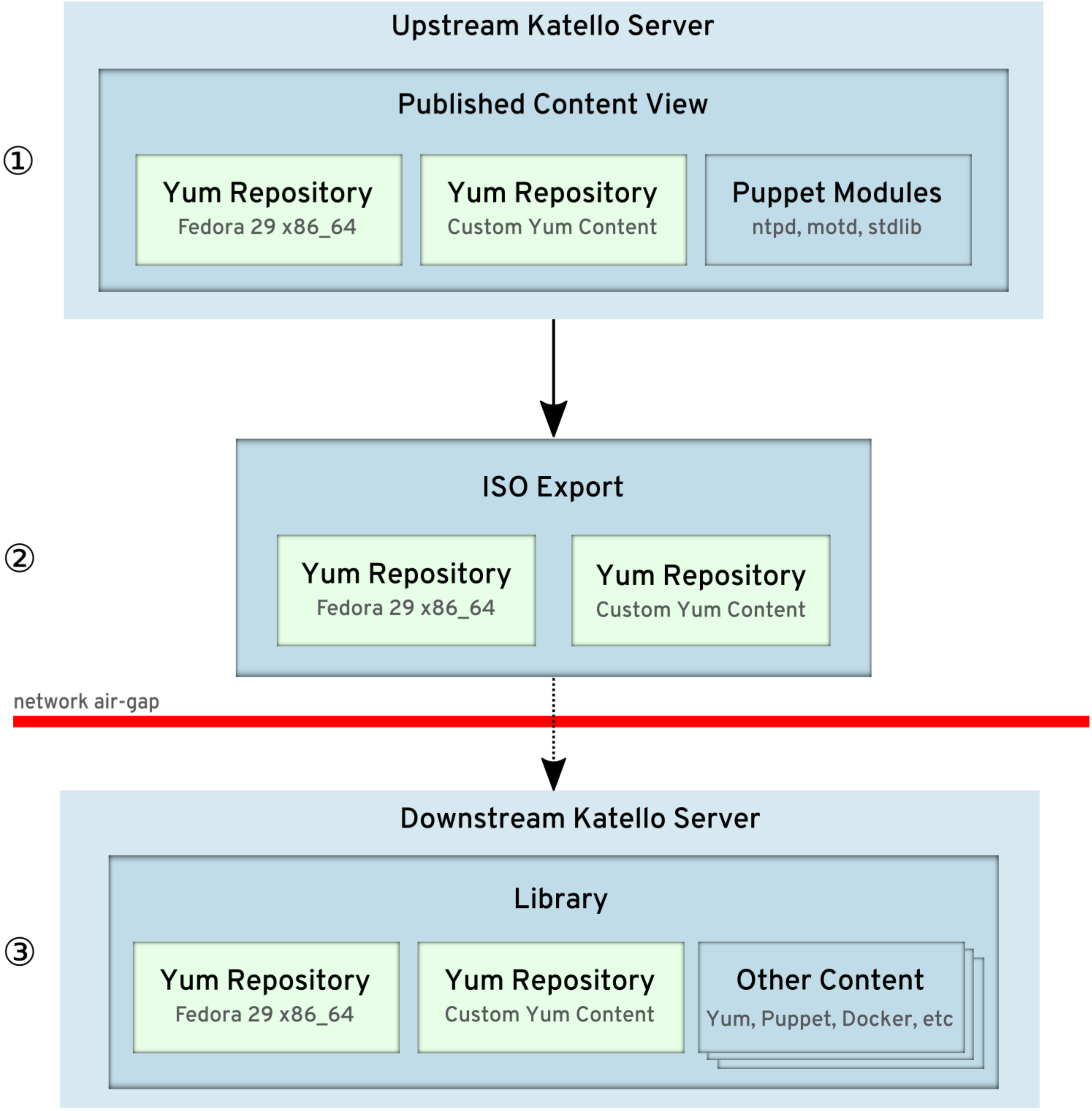
Intro

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 defaults to `/var/lib/pulp/katello_export` . Please be aware that the location needs to be readable and writable by the `foreman` user. SELinux permissions also need to be set on the export location with the type `httpd_sys_rw_content_t` as well as `foreman` user and group ownership.

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 Katello instance. Simply put the export in `/var/www/html/pub/export` , either via copy or symlink. After that, edit your CDN location from the manifest import page to point to “http:///export/path/to/export" and the Red Hat Repos page will then work as expected, using your exported data. Please be sure to use 'http' and not 'https' when altering the CDN url. Katello by default only supports the CA certificate for `cdn.redhat.com`. This is a [known limitation] (http://projects.theforeman.org/issues/16392) that will be addressed in a future version.

The second way is to perform a repository sync via hammer, specifiying 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.

Katello 3.15 Documentation

3.15

Docker Management

Katello can be used to manage and deploy Docker content. Katello can retreive 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 proivisioned 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

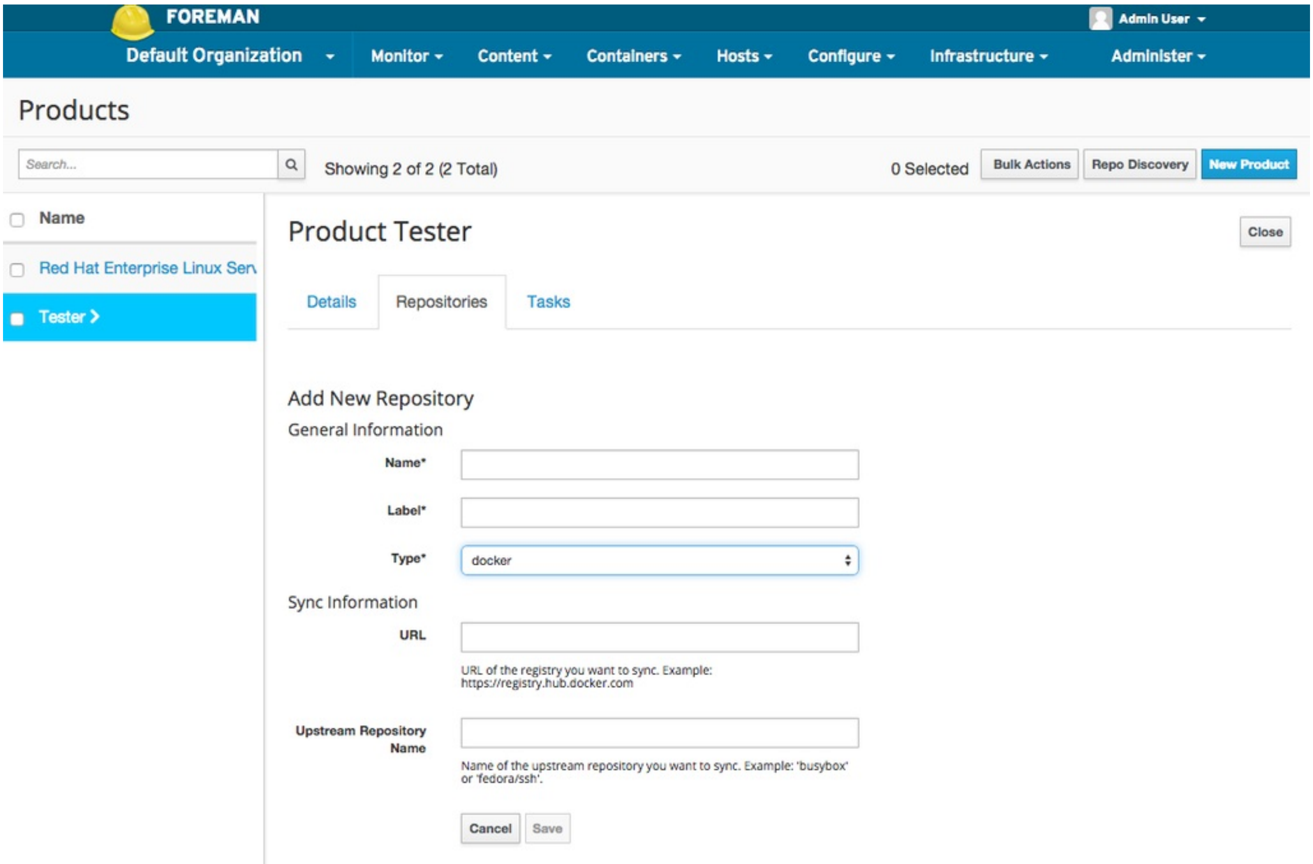
The easiest way to get Docker content into Katello is to sync it in. You can either sync Docker content from the Red Hat CDN (if you have subscriptions for the content) or from a registry such as Docker Hub.

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.



Products

Showing 2 of 2 (2 Total)

0 Selected

Product Tester

Add New Repository

General Information

Name*

Label*

Type* docker

Sync Information

URL

URL of the registry you want to sync. Example: https://registry.hub.docker.com

Upstream Repository Name

Name of the upstream repository you want to sync. Example: 'busybox' or 'fedora/ssh'.

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-1.docker.io/` .

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.

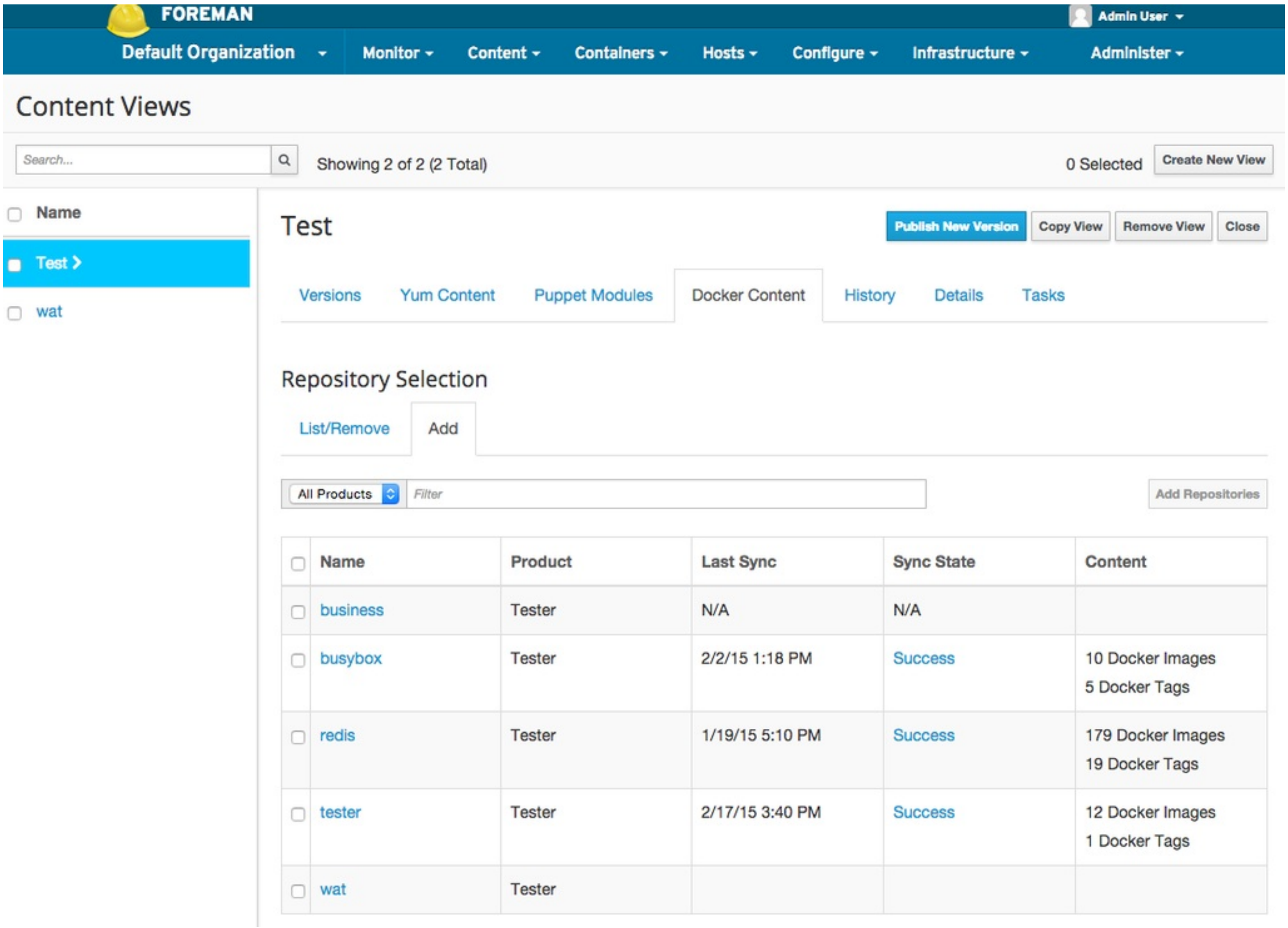
As a workaround, you can create a local Docker registry like so:

```
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.



Test

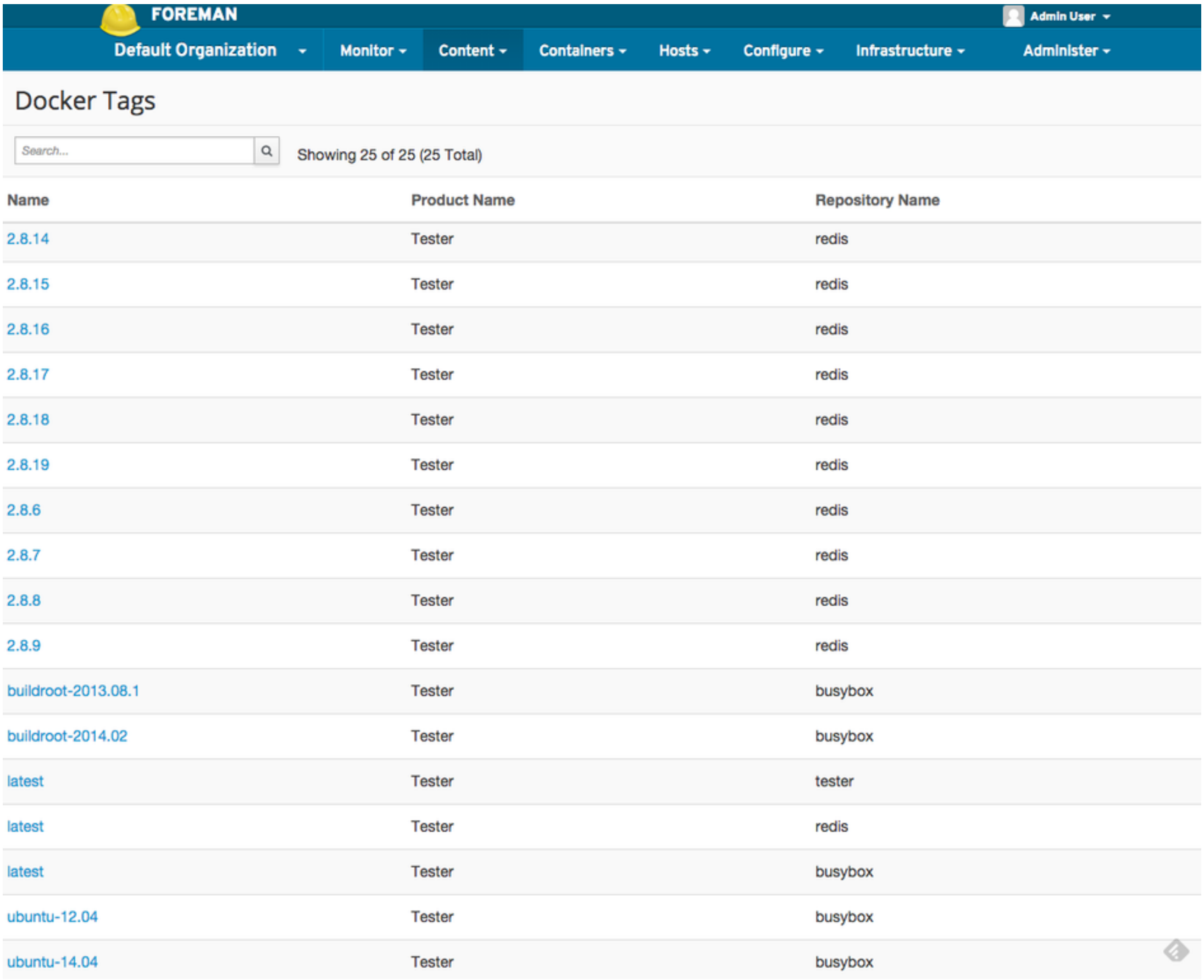
Repository Selection

☐	Name	Product	Last Sync	Sync State	Content
☐	business	Tester	N/A	N/A	
☐	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.

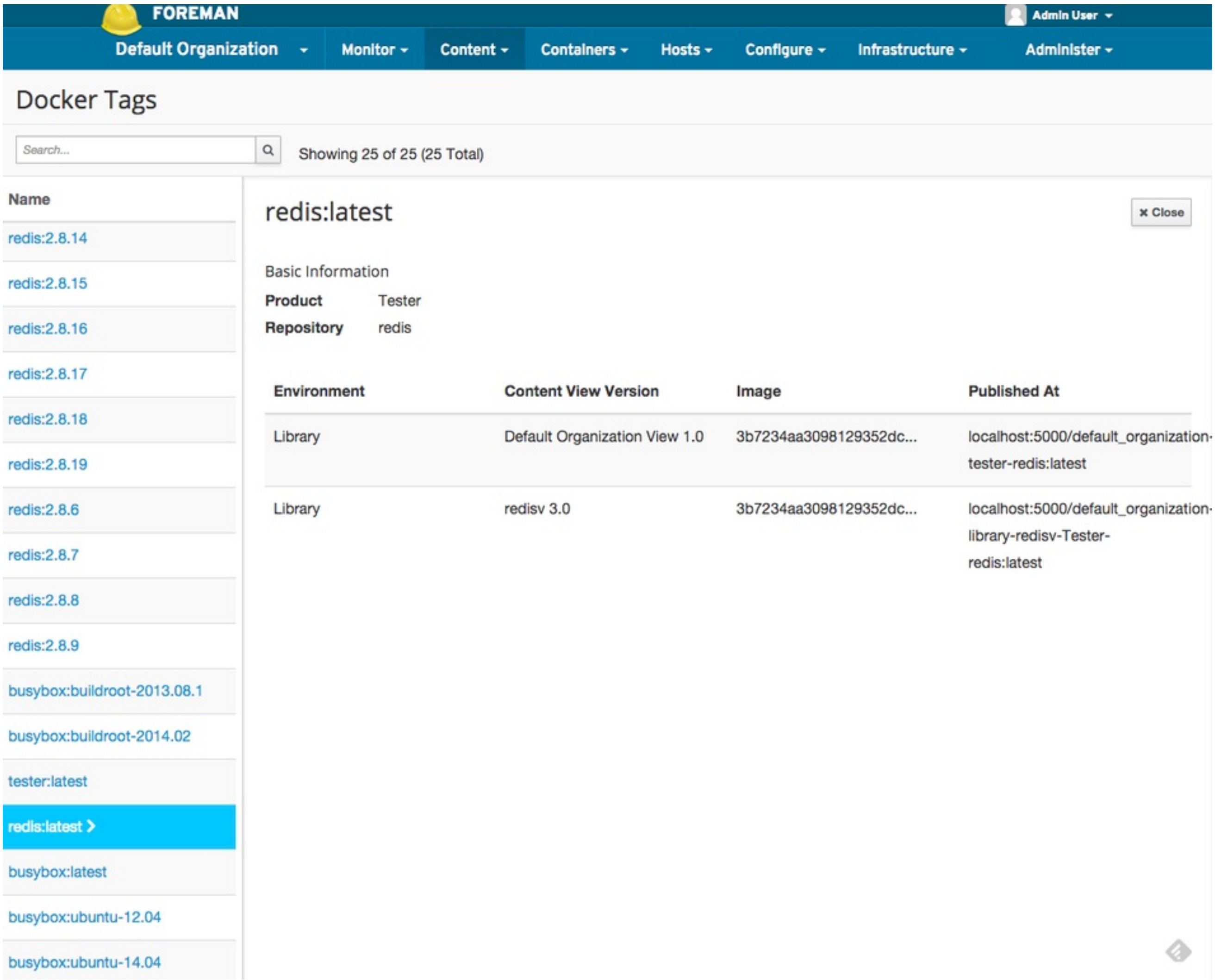


Docker Tags

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.



Docker Tags

Showing 25 of 25 (25 Total)

Name
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

redis:latest

Basic Information

Product Tester

Repository redis

Environment	Content View Version	Image	Published At
Library	Default Organization View 1.0	3b7234aa3098129352dc...	localhost:5000/default_organization-tester-redis:latest
Library	redisv 3.0	3b7234aa3098129352dc...	localhost:5000/default_organization-library-redisv-Tester-redis:latest

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/Smart Proxy. 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.

Katello 3.15 Documentation

3.15

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 of how the Foreman/Katello service sends email is located in `Adminster > Settings > Email`.

For more information see: Email Configuration

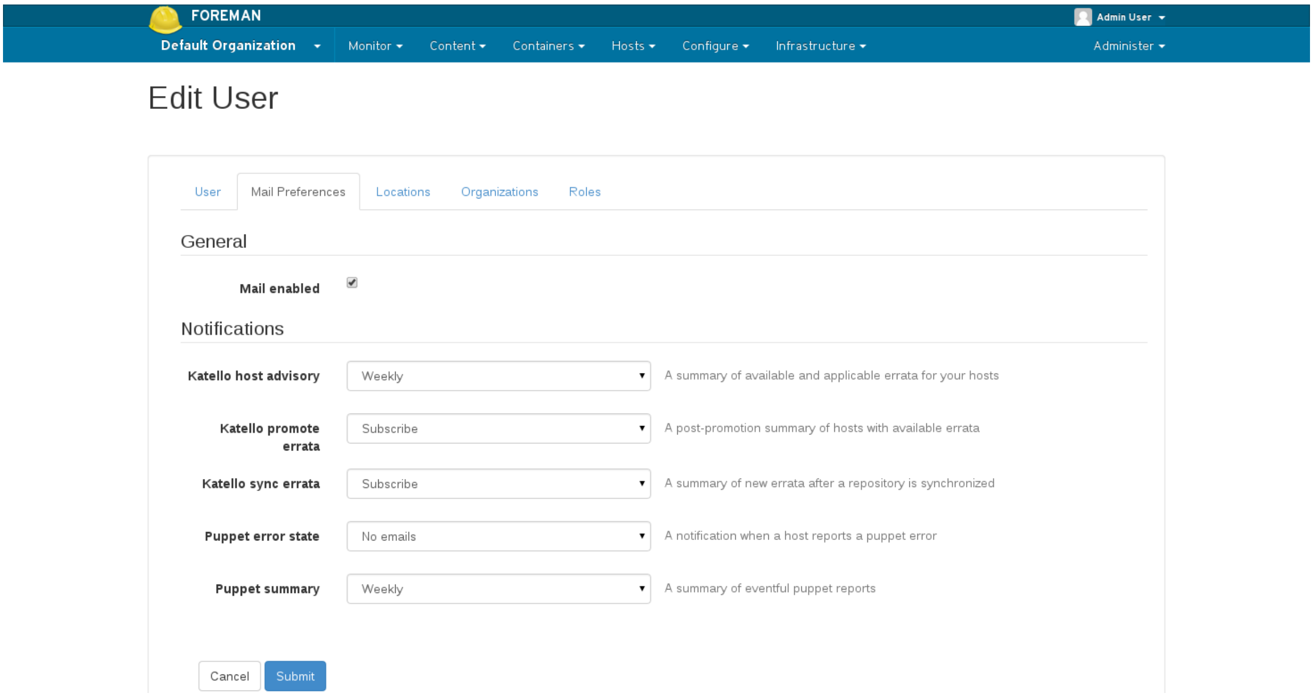
Opting in to the emails

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”.



Edit User

General

Mail enabled ☑

Notifications

Katello host advisory	Weekly	A summary of available and applicable errata for your hosts
Katello promote errata	Subscribe	A post-promotion summary of hosts with available errata
Katello sync errata	Subscribe	A summary of new errata after a repository is synchronized
Puppet error state	No emails	A notification when a host reports a puppet error
Puppet summary	Weekly	A summary of eventful puppet reports

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15 ▾

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

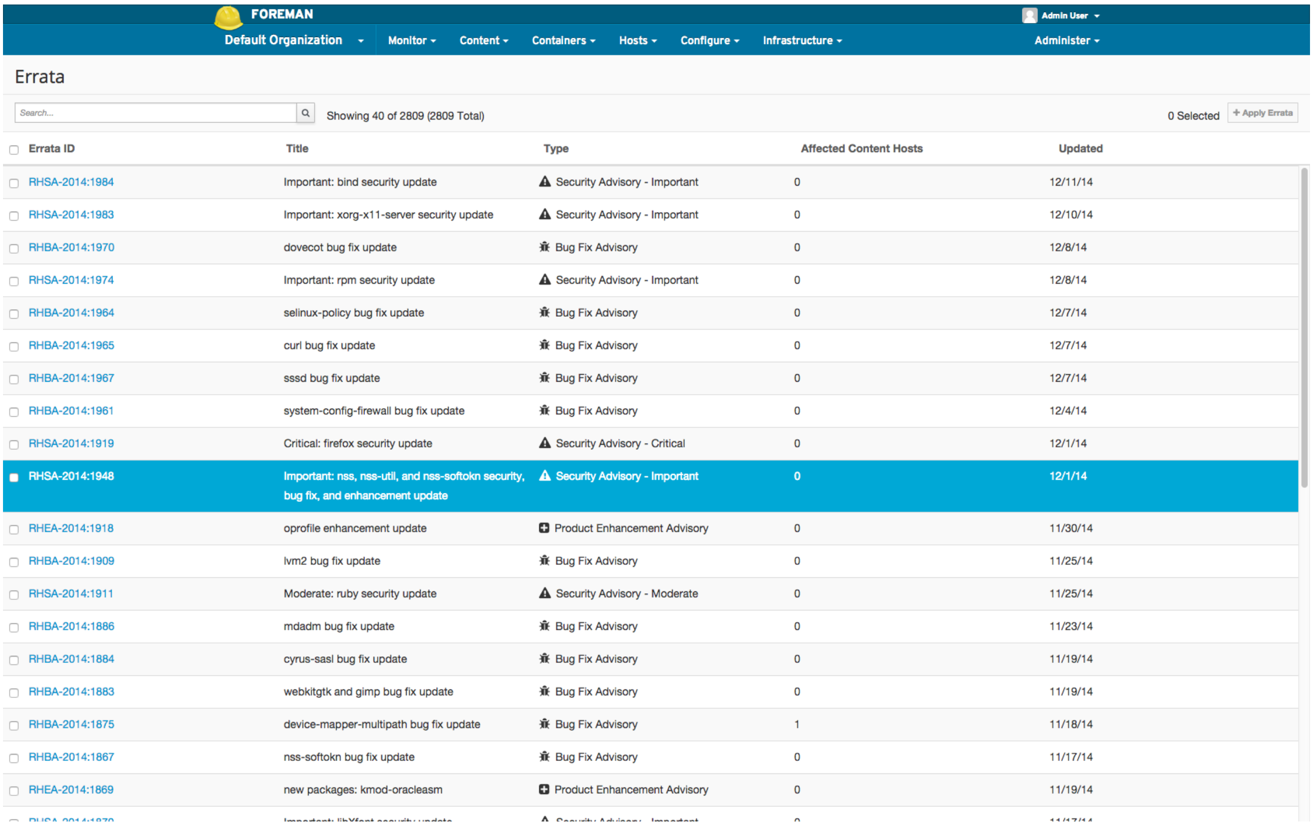
The following is a high-level summary of the Errata features:

- 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

Showing 40 of 2809 (2809 Total)

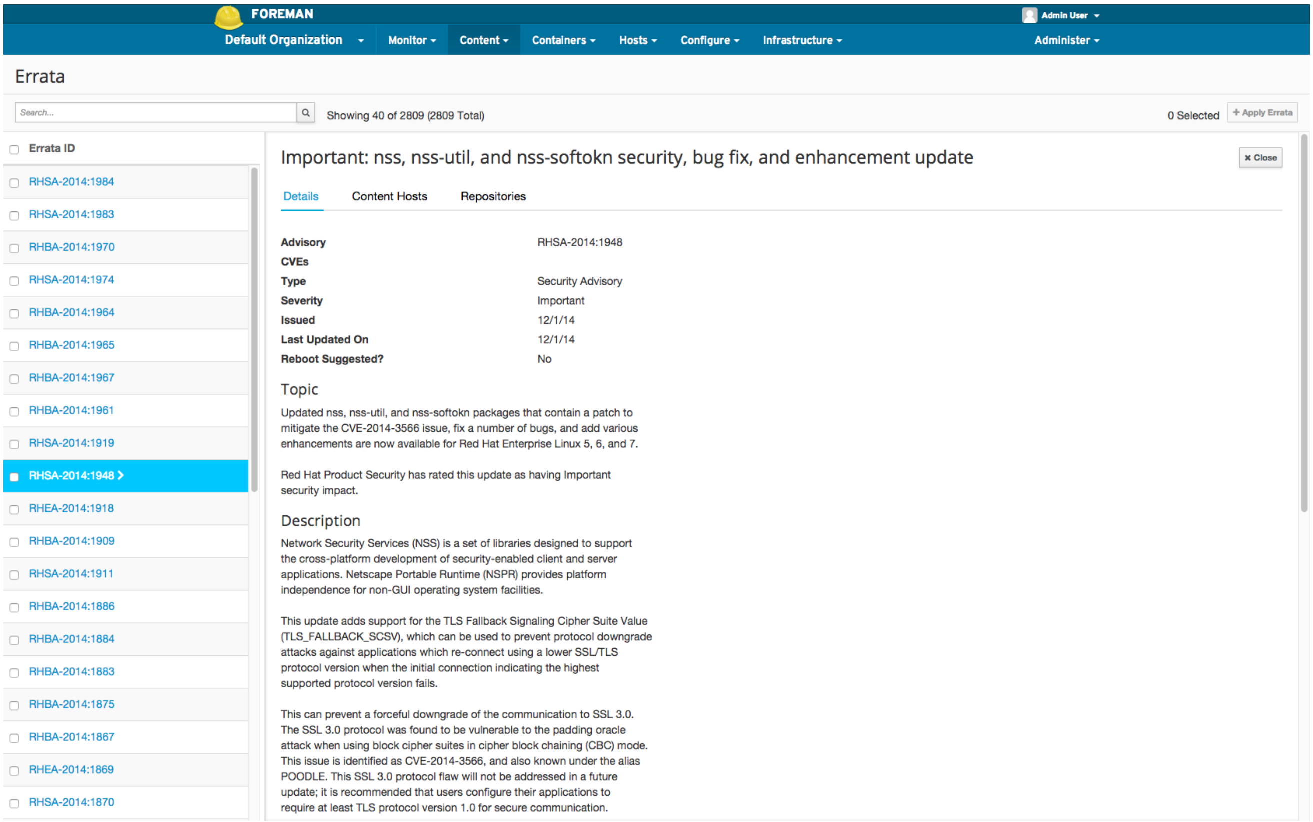
0 Selected

□ Errata ID	Title	Type	Affected Content Hosts	Updated
□ RHSA-2014:1984	Important: bind security update	⚠ Security Advisory - Important	0	12/11/14
□ RHSA-2014:1983	Important: xorg-x11-server security update	⚠ Security Advisory - Important	0	12/10/14
□ RHBA-2014:1970	dovecot bug fix update	Bug Fix Advisory	0	12/8/14
□ RHSA-2014:1974	Important: rpm security update	⚠ Security Advisory - Important	0	12/8/14
□ RHBA-2014:1964	selinux-policy bug fix update	Bug Fix Advisory	0	12/7/14
□ RHBA-2014:1965	curl bug fix update	Bug Fix Advisory	0	12/7/14
□ RHBA-2014:1967	sssd bug fix update	Bug Fix Advisory	0	12/7/14
□ RHBA-2014:1961	system-config-firewall bug fix update	Bug Fix Advisory	0	12/4/14
□ RHSA-2014:1919	Critical: firefox security update	⚠ Security Advisory - Critical	0	12/1/14
□ RHSA-2014:1948	Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update	⚠ Security Advisory - Important	0	12/1/14
□ RHEA-2014:1918	oprofile enhancement update	⊞ Product Enhancement Advisory	0	11/30/14
□ RHBA-2014:1909	lvm2 bug fix update	Bug Fix Advisory	0	11/25/14
□ RHSA-2014:1911	Moderate: ruby security update	⚠ Security Advisory - Moderate	0	11/25/14
□ RHBA-2014:1886	mdadm bug fix update	Bug Fix Advisory	0	11/23/14
□ RHBA-2014:1884	cyrus-sasl bug fix update	Bug Fix Advisory	0	11/19/14
□ RHBA-2014:1883	webkitgtk and gimp bug fix update	Bug Fix Advisory	0	11/19/14
□ RHBA-2014:1875	device-mapper-multipath bug fix update	Bug Fix Advisory	1	11/18/14
□ RHBA-2014:1867	nss-softokn bug fix update	Bug Fix Advisory	0	11/17/14
□ RHEA-2014:1869	new packages: kmod-oracleasm	⊞ Product Enhancement Advisory	0	11/19/14
□ RHSA-2014:1870	Important: libXfont security update	⚠ Security Advisory - Important	0	11/17/14

View Errata Details

To view the details of an Errata:

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



Errata

Showing 40 of 2809 (2809 Total)

- □ 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

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.

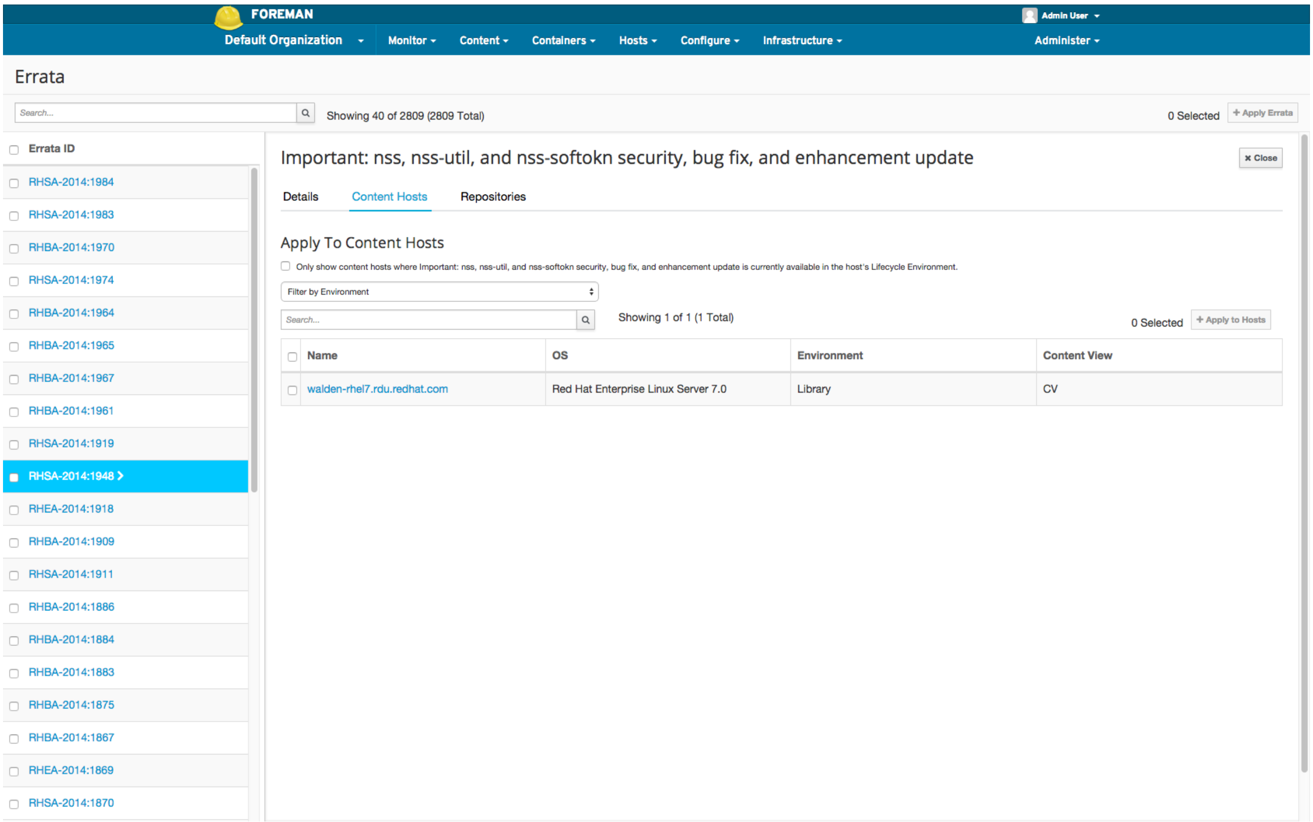
View Affected Content Hosts

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.



Errata

Showing 40 of 2809 (2809 Total)

- □ 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

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.

Showing 1 of 1 (1 Total)

0 Selected

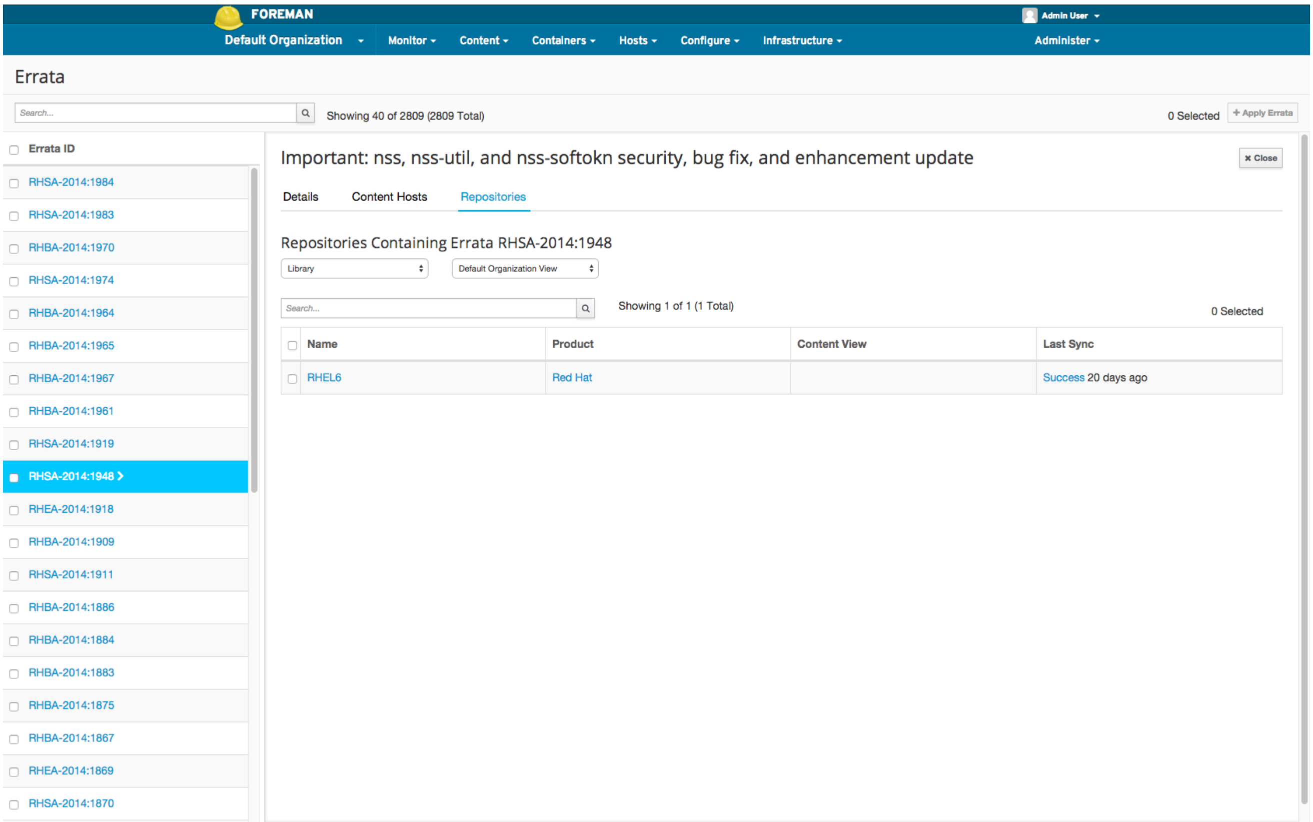
□	Name	OS	Environment	Content View
□	walden-rhel7.rdu.redhat.com	Red Hat Enterprise Linux Server 7.0	Library	CV

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.



Showing 40 of 2809 (2809 Total)

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

Repositories Containing Errata RHSA-2014:1948

Showing 1 of 1 (1 Total)

Name	Product	Content View	Last Sync
RHEL6	Red Hat		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.

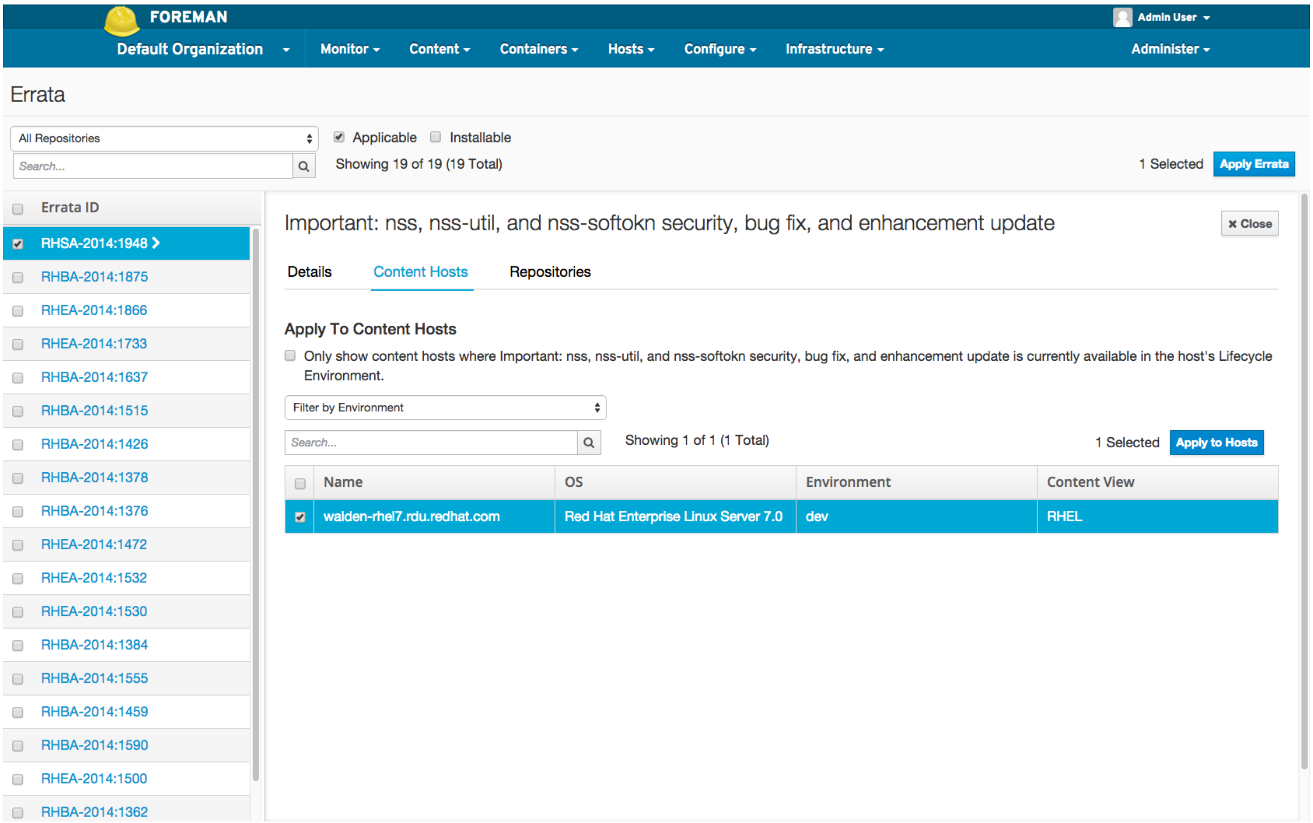
There are two ways to apply Errata:

- 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



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

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

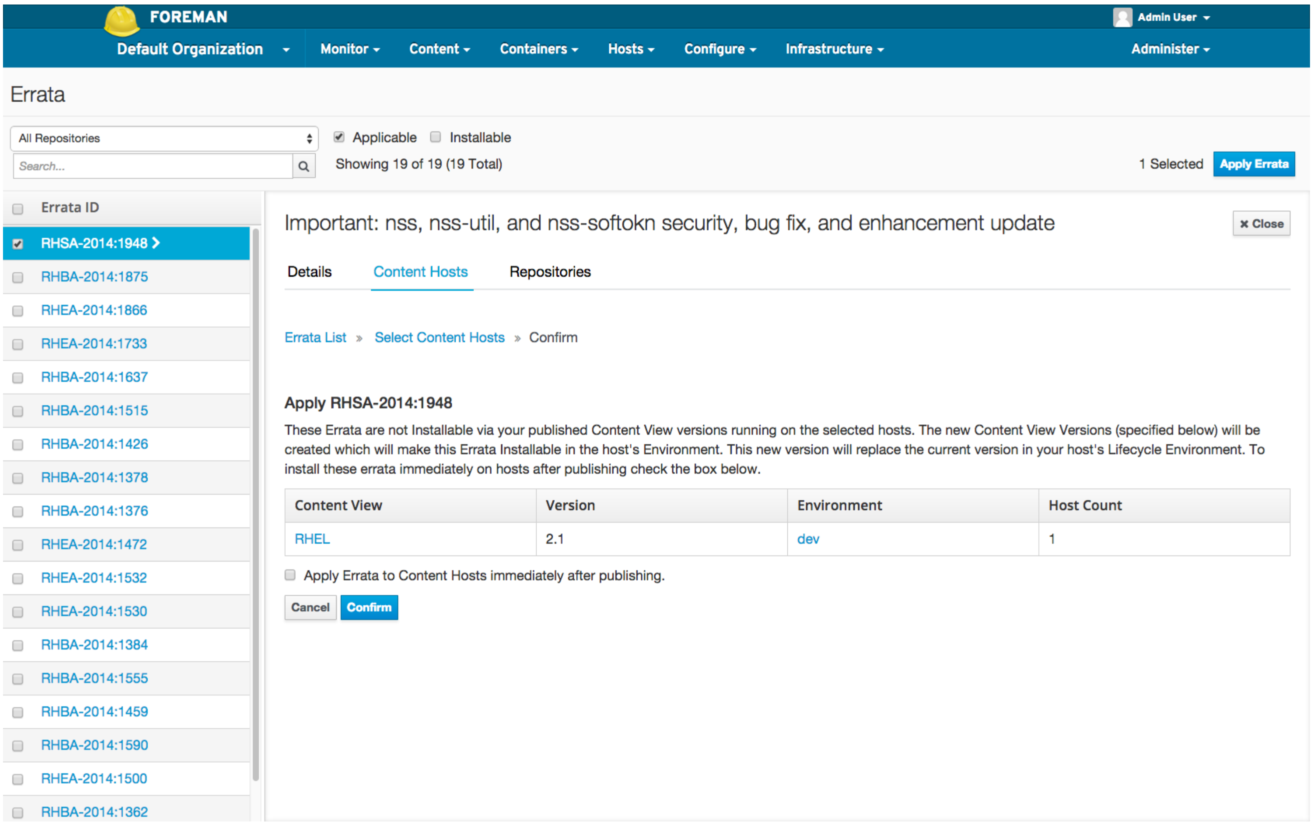
Apply To Content Hosts

□ 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.

Showing 1 of 1 (1 Total)

1 Selected

□	Name	OS	Environment	Content View
☑	walden-rhel7.rdu.redhat.com	Red Hat Enterprise Linux Server 7.0	dev	RHEL



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

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

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.

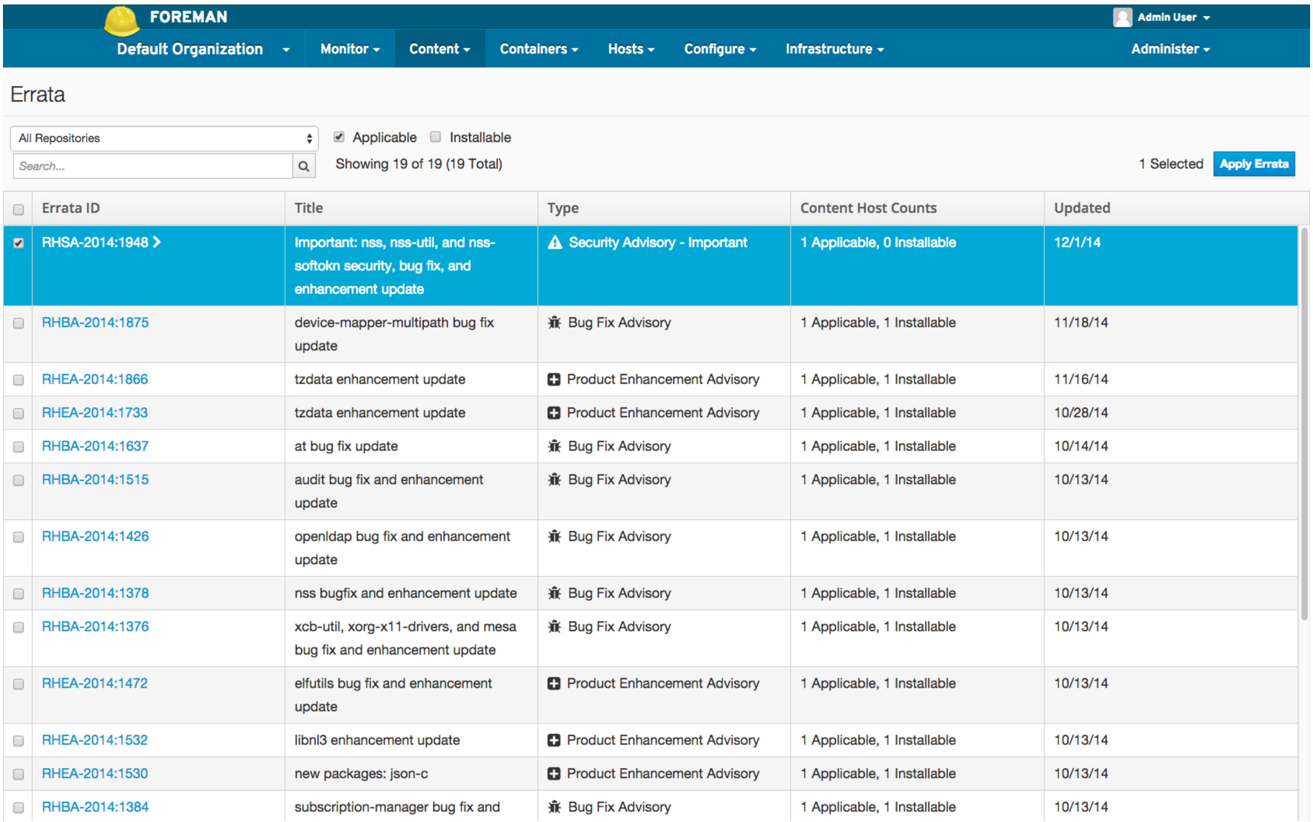
Content View	Version	Environment	Host Count
RHEL	2.1	dev	1

□ 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



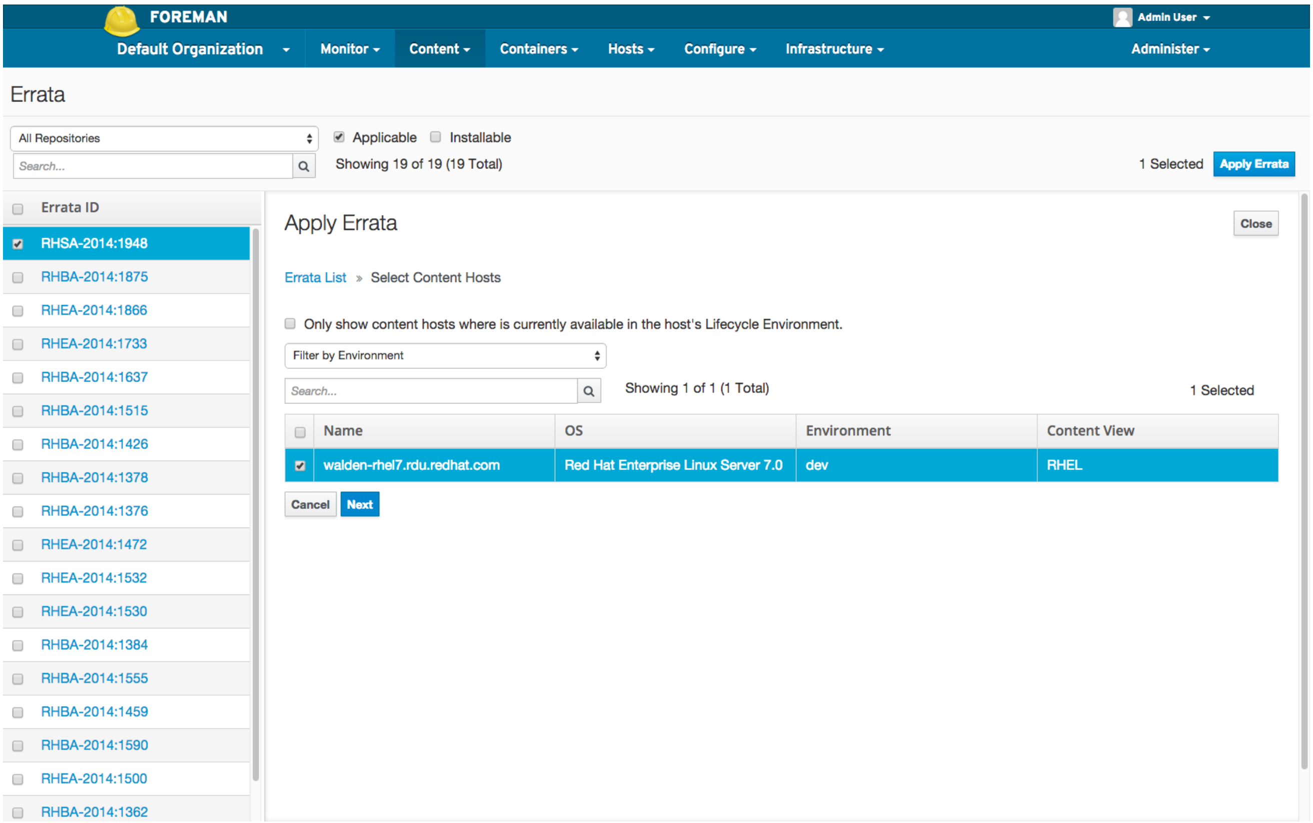
Errata

☑ Applicable □ Installable

Showing 19 of 19 (19 Total)

1 Selected

□	Errata ID	Title	Type	Content Host Counts	Updated
☑	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
□	RHBA-2014:1875	device-mapper-multipath bug fix update	Bug Fix Advisory	1 Applicable, 1 Installable	11/18/14
□	RHEA-2014:1866	tzdata enhancement update	Product Enhancement Advisory	1 Applicable, 1 Installable	11/16/14
□	RHEA-2014:1733	tzdata enhancement update	Product Enhancement Advisory	1 Applicable, 1 Installable	10/28/14
□	RHBA-2014:1637	at bug fix update	Bug Fix Advisory	1 Applicable, 1 Installable	10/14/14
□	RHBA-2014:1515	audit bug fix and enhancement update	Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14
□	RHBA-2014:1426	openldap bug fix and enhancement update	Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14
□	RHBA-2014:1378	nss bugfix and enhancement update	Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14
□	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
□	RHEA-2014:1472	elfutils bug fix and enhancement update	Product Enhancement Advisory	1 Applicable, 1 Installable	10/13/14
□	RHEA-2014:1532	libnl3 enhancement update	Product Enhancement Advisory	1 Applicable, 1 Installable	10/13/14
□	RHEA-2014:1530	new packages: json-c	Product Enhancement Advisory	1 Applicable, 1 Installable	10/13/14
□	RHBA-2014:1384	subscription-manager bug fix and	Bug Fix Advisory	1 Applicable, 1 Installable	10/13/14



Errata

☑ 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

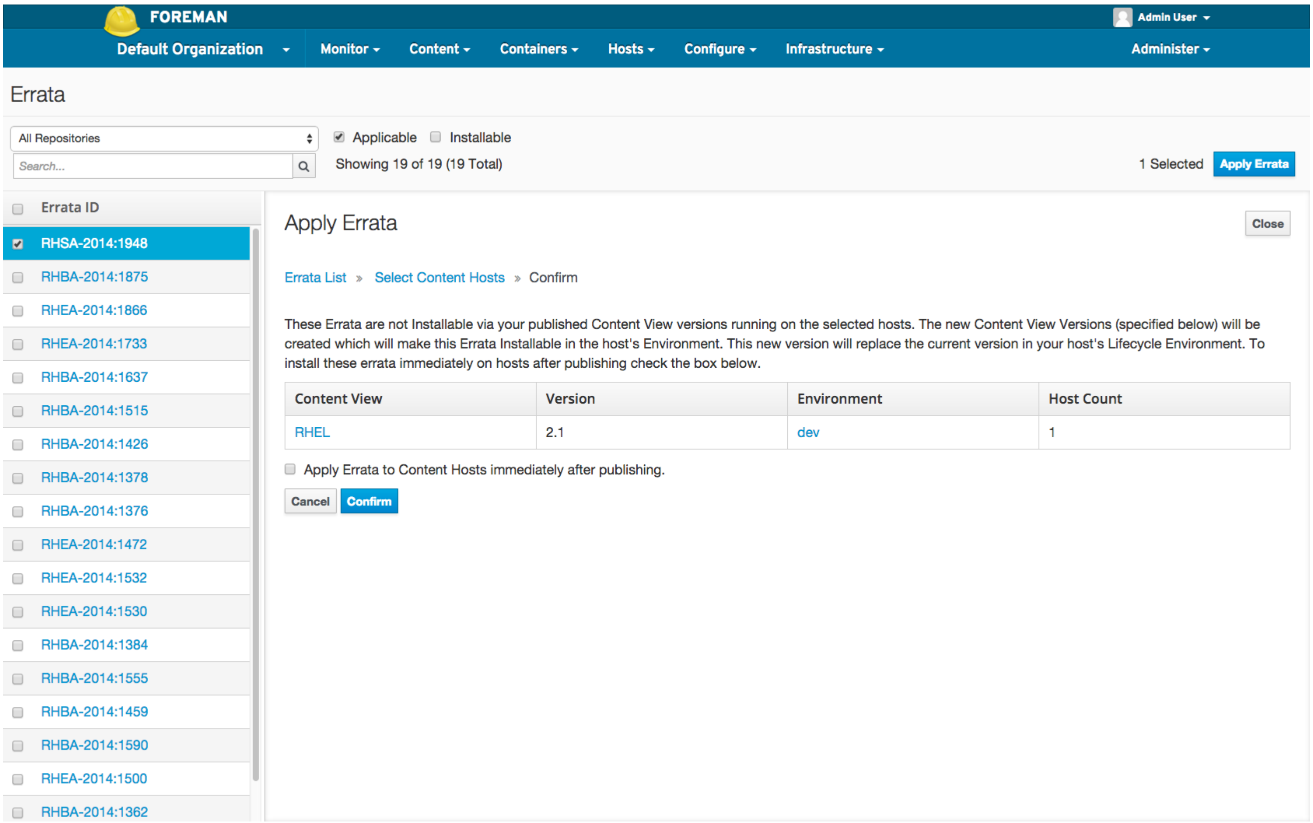
Errata List » Select Content Hosts

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

Showing 1 of 1 (1 Total)

1 Selected

□	Name	OS	Environment	Content View
☑	walden-rhel7.rdu.redhat.com	Red Hat Enterprise Linux Server 7.0	dev	RHEL



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.

Katello 3.15 Documentation

3.15 ▾

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.

Facter

A program that provides information (facts) about the system on which it is run (eg: total memory, operating system version, architecture, etc.) Facter 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.

Katello 3.15 Documentation

3.15

Content Credentials

Content Credentials can be associated with Katello Products and Repositories. Two types of Content Credentials are supported, each having a distinct purpose:

GPG Keys

GPG Keys are used by Content Hosts in order to verify the signature of packages that have been retrieved from a Repository and ensure they haven't been corrupted. For more information on GPG Keys see The GNU Privacy Guard.

SSL Certificates

SSL Certificates are used by the Katello server to sync content from upstream repositories that require client SSL authentication.

Using Content Credentials

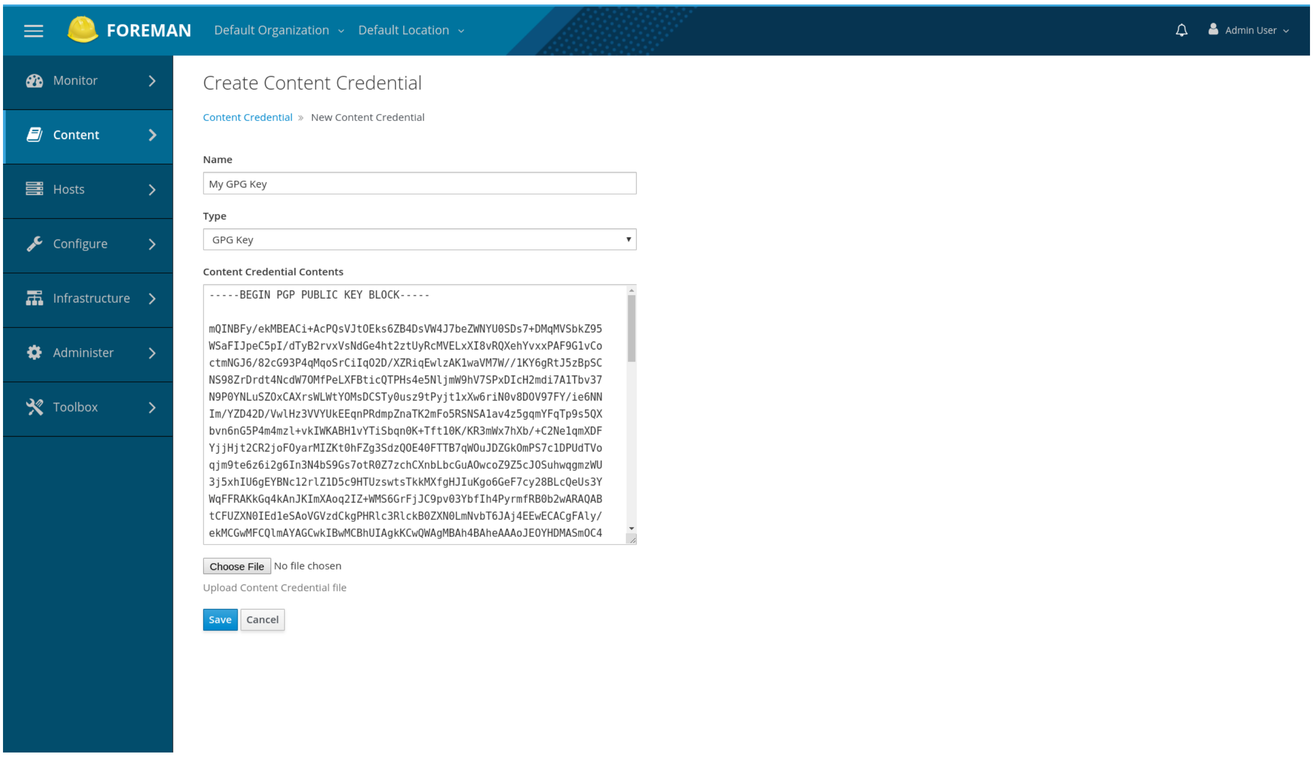
- Create a Content Credential
- Associate Content Credential with a Product
- View Associated Products
- View Associated Repositories

Create a Content Credential

To create a new Content Credential:

- navigate to Content > Content Credentials
- click **Create Content Credential**
- Enter a Name and select a Type
- You may either upload your Credential or paste its content into the text area.

When creating SSL Certificates for Products or Repositories that require them, a separate credential must be created for the CA, cert, and key respectively.



Create Content Credential

Content Credential » New Content Credential

Name

My GPG Key

Type

GPG Key

Content Credential Contents

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFy/ekMBEACi+AcPQsVJtOEks6ZB4DsVW4J7beZWNYU0SDs7+DMqMVSbkZ95
WSaFIJpeC5pI/dTyB2rvxVsNdGe4ht2ztUyRcMVELxXI8vRQXehYvxxPAF9G1vCo
ctmNGJ6/82cG93P4qMqoSrCiIqO2D/XZRiqEwlzAK1waVM7W//1KY6gRtJ5zBpSC
NS98ZrDrdt4NcdW7OMfPeLXFBticQTPHs4e5NljmW9hV7SPxDIcH2mdi7A1Tbv37
N9P0YNLuSZOxCAXrsWLWtYOMsDCSTy0usz9tPyjt1xXw6riN0v8DOV97FY/ie6NN
Im/YZD42D/VwlHz3VVYUkEEqnPRdmpZnaTK2mFo5RSNSA1av4z5gqmYFqTp9s5QX
bvn6nG5P4m4mzl+vkIWKABH1vYTiSbqn0K+Tft10K/KR3mWx7hXb/+C2Ne1qmXDF
YjjHjt2CR2joFOyarMIZKt0hFZg3SdzQOE40FTTB7qWOuJDZGkOmPS7c1DPUdTVo
qjm9te6z6i2g6In3N4bS9Gs7otR0Z7zchCXnbLbcGuAOwcoZ9Z5cJOSuhwqgmzWU
3j5xhIU6gEYBNc12rlZ1D5c9HTUzswtsTkkMXfgHJIuKgo6GeF7cy28BLcQeUs3Y
WqFFRAKkGq4kAnJKImXAoq2IZ+WMS6GrFjJC9pv03YbfIh4PyrmfRB0b2wARAQAB
tCFUZXN0IEd1eSAoVGVzdCkgPHRlc3RlckB0ZXN0LmNvbT6JAj4EEwECACgFAly/
ekMCGwMFCQlmAYAGCwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEOYHDMASmOC4
```

Choose File No file chosen

Upload Content Credential file

Save Cancel

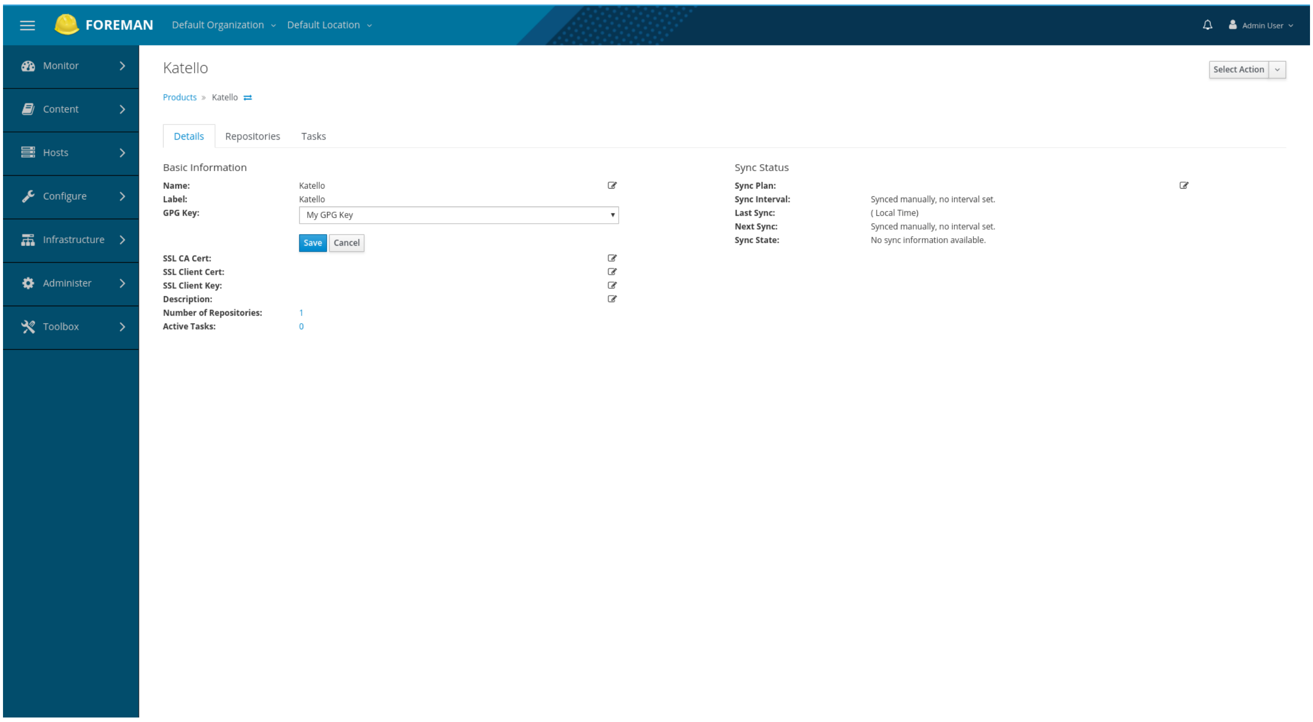
Associate Content Credential with a Product

To add a Content Credential to a Product:

Note that adding a Credential to a Product adds it to all current and future repositories unless a repository already has a Credential assigned. The Credential currently assigned to a Repository can be modified or overridden from the detail view of the repository.

- navigate to Content > Products
- select the desired Product from the list
- click **Details**
- click the edit icon on the GPG Key, SSL CA Cert, SSL Client Cert, or SSL Client Key field
- select the desired Content Credential

The steps for adding a Credential to a Repository are the same but performed from the repository’s detail view.



Katello

Products » Katello

Basic Information

Name: Katello
Label: Katello
GPG Key: My GPG Key
SSL CA Cert:
SSL Client Cert:
SSL Client Key:
Description:
Number of Repositories: 1
Active Tasks: 0

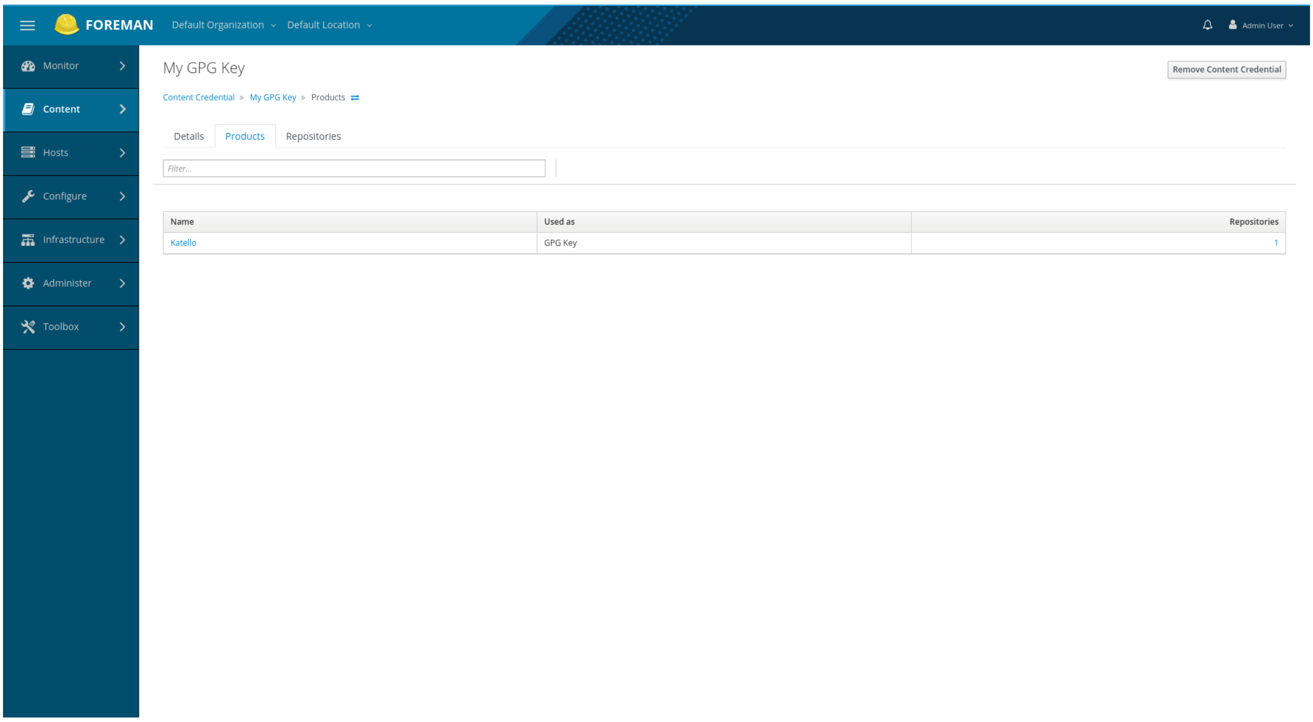
Sync Status

Sync Plan:
Sync Interval: Synced manually, no interval set.
Last Sync: (Local Time)
Next Sync: Synced manually, no interval set.
Sync State: No sync information available.

View Associated Products

To view all Products that have been assigned a Content Credential:

- navigate to Content > Content Credentials
- select the desired Credential from the list
- click **Products**



My GPG Key

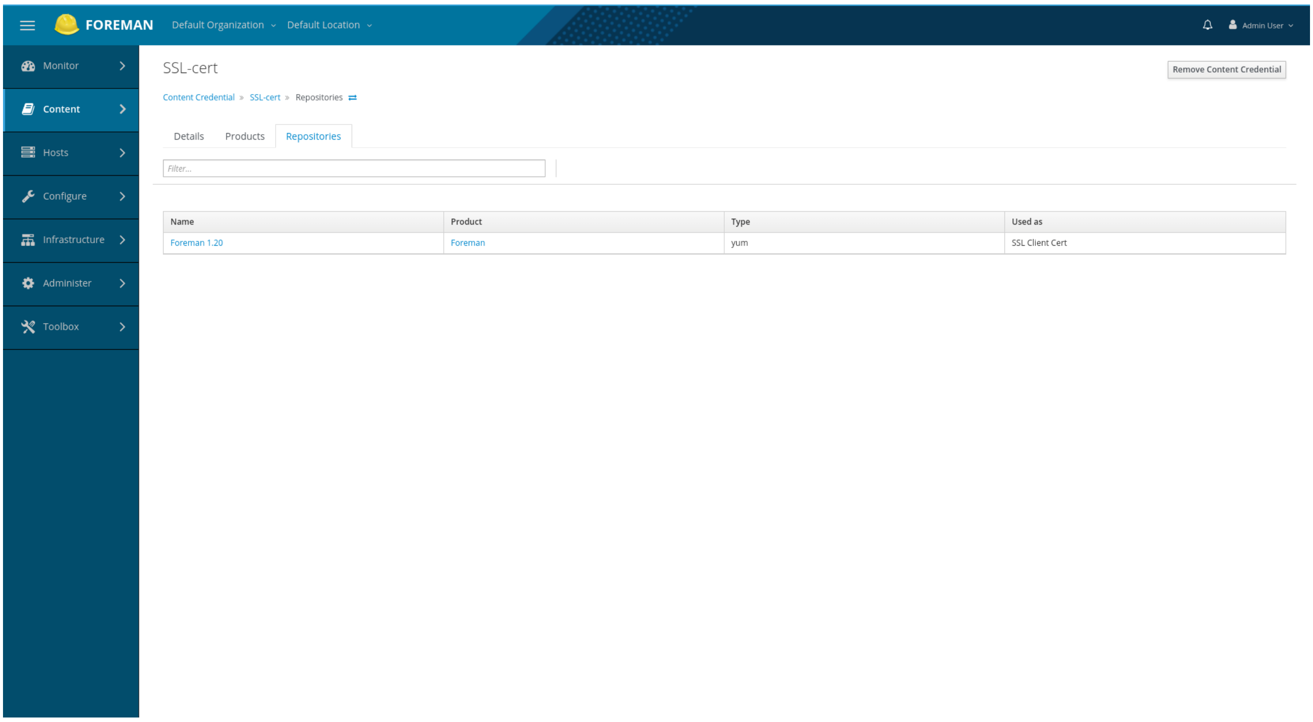
Content Credential » My GPG Key » Products

Name	Used as	Repositories
Katello	GPG Key	1

View Associated Repositories

To view all Repositories that have been assigned a Content Credential:

- navigate to Content > Content Credentials
- select the desired Credential from the list
- click **Repositories**



SSL-cert

Content Credential » SSL-cert » Repositories

Name	Product	Type	Used as
Foreman 1.20	Foreman	yum	SSL Client Cert

Katello 3.15 Documentation

3.15

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

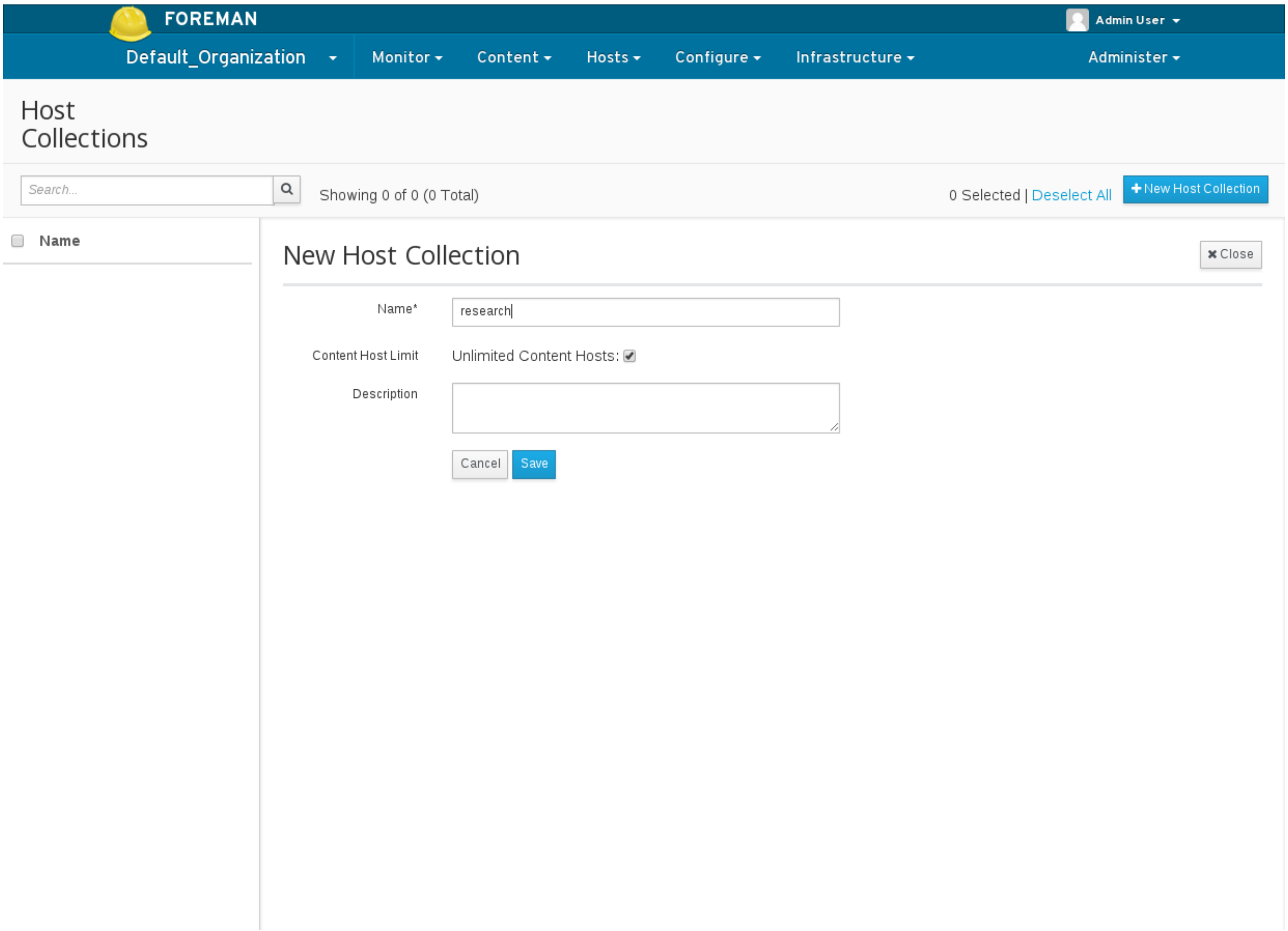
Create a Host Collection

To create a new collection,

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

Note the following option:

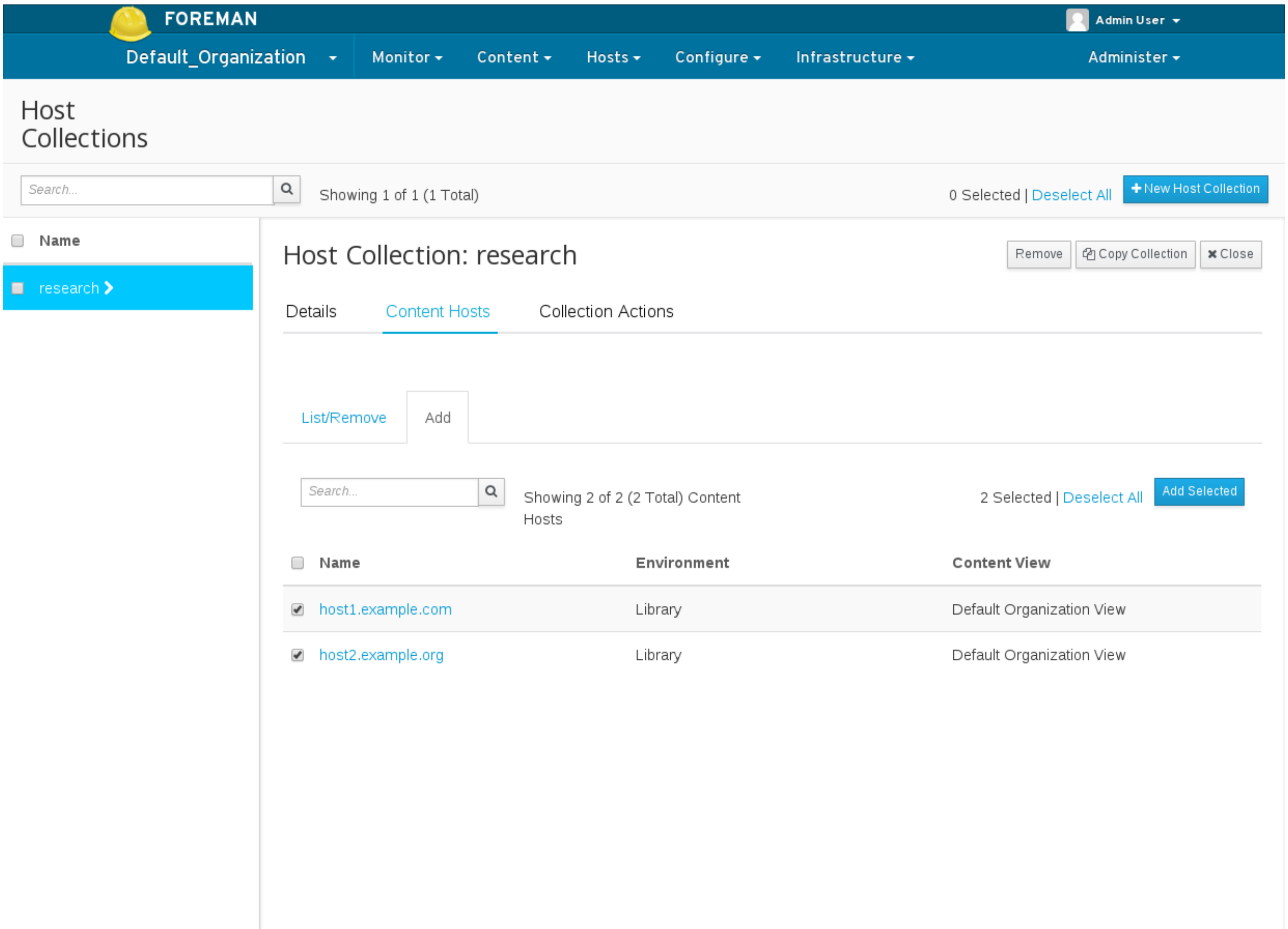
- *Content Host Limit*: This option will control how many Content Hosts are allowed to be added to the collection.



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**



Host Collections

Showing 1 of 1 (1 Total)

0 Selected | Deselect All

□	Name
□	research

Host Collection: research

Showing 2 of 2 (2 Total) Content Hosts

2 Selected | Deselect All

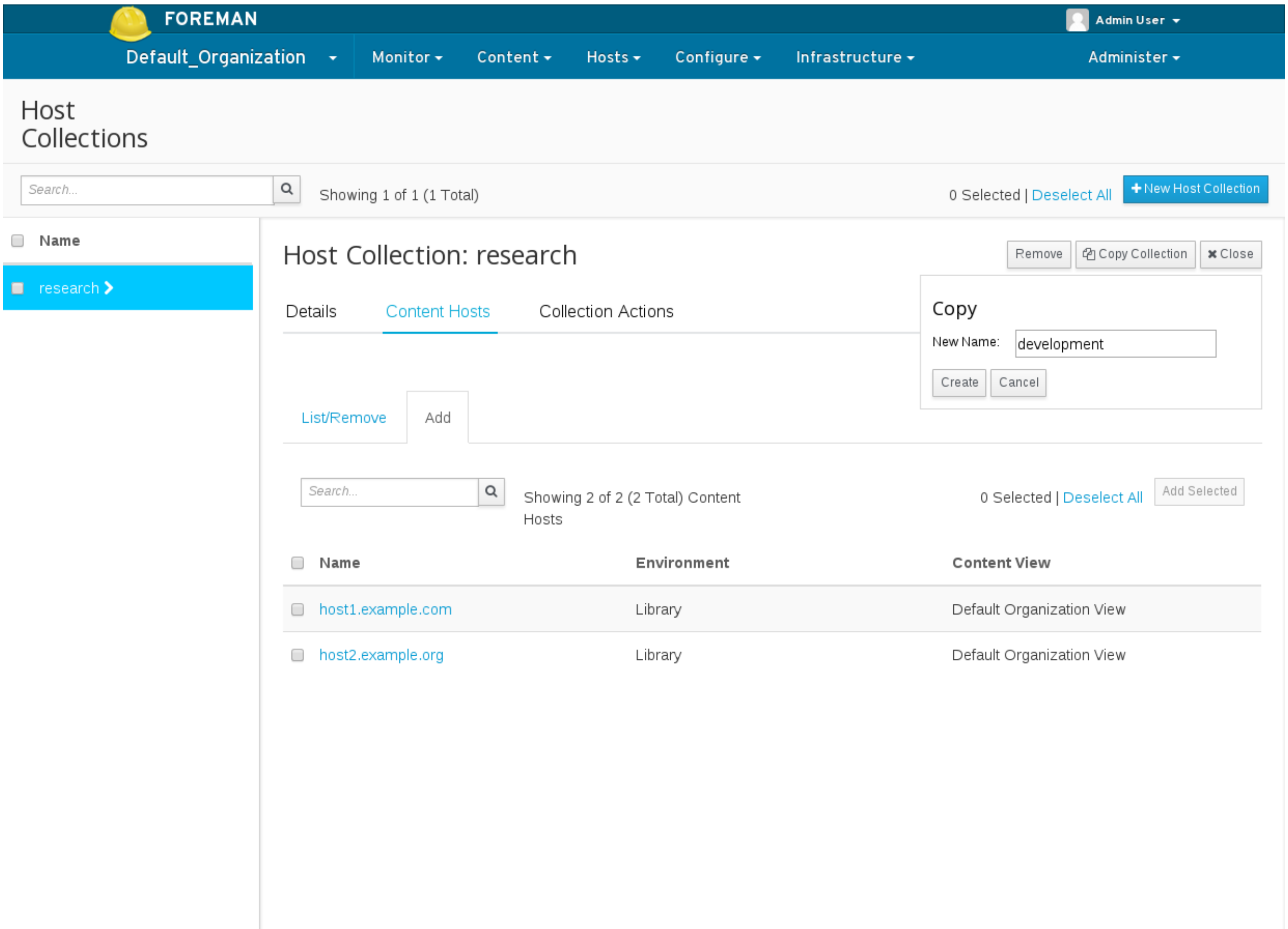
□	Name	Environment	Content View
☑	host1.example.com	Library	Default Organization View
☑	host2.example.org	Library	Default Organization View

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**



Host Collections

Showing 1 of 1 (1 Total)

0 Selected | Deselect All

Name
research

Host Collection: research

Copy

New Name: development

Showing 2 of 2 (2 Total) Content Hosts

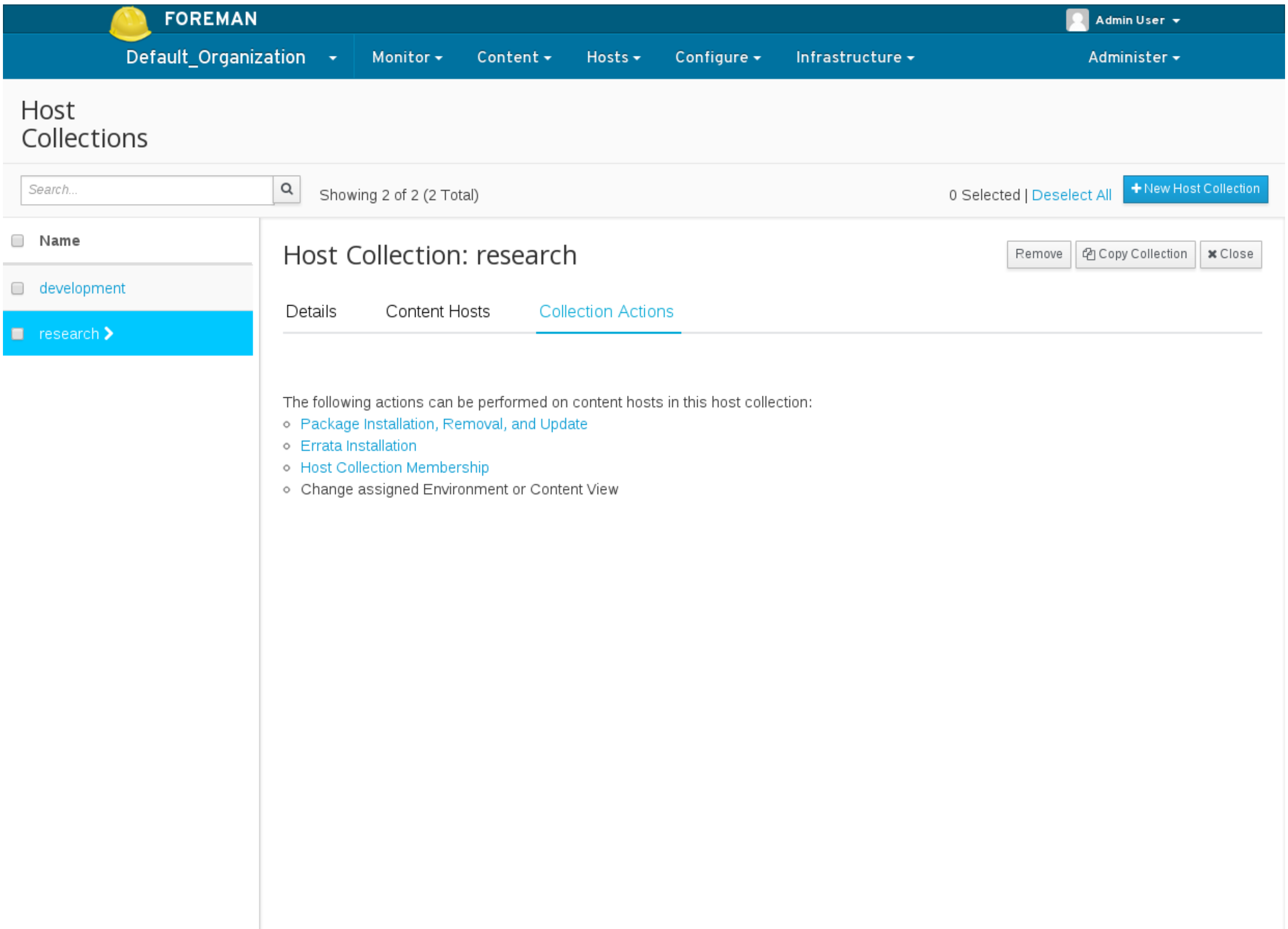
0 Selected | Deselect All

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



Host Collections

Showing 2 of 2 (2 Total)

0 Selected | Deselect All

Host Collection: research

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

- Package Installation, Removal, and Update
- Errata Installation
- Host Collection Membership
- 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

Katello 3.15 Documentation

3.15

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.

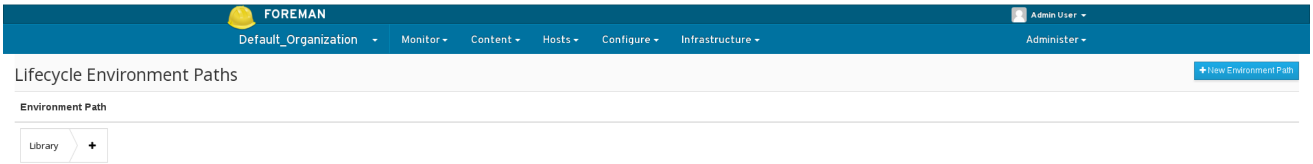
General Workflow

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

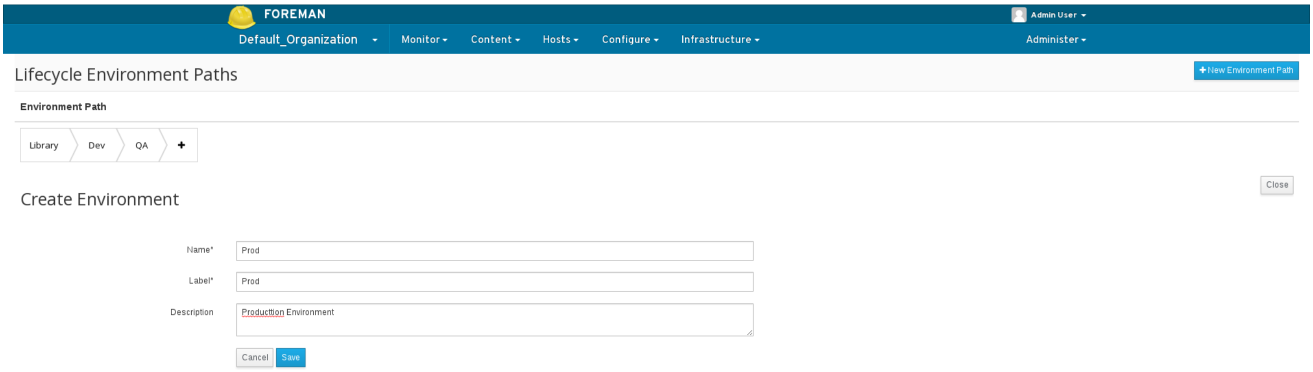
From the web UI, navigate to:

Content -> Lifecycle Environments



Creating a lifecycle environment

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



Create Environment

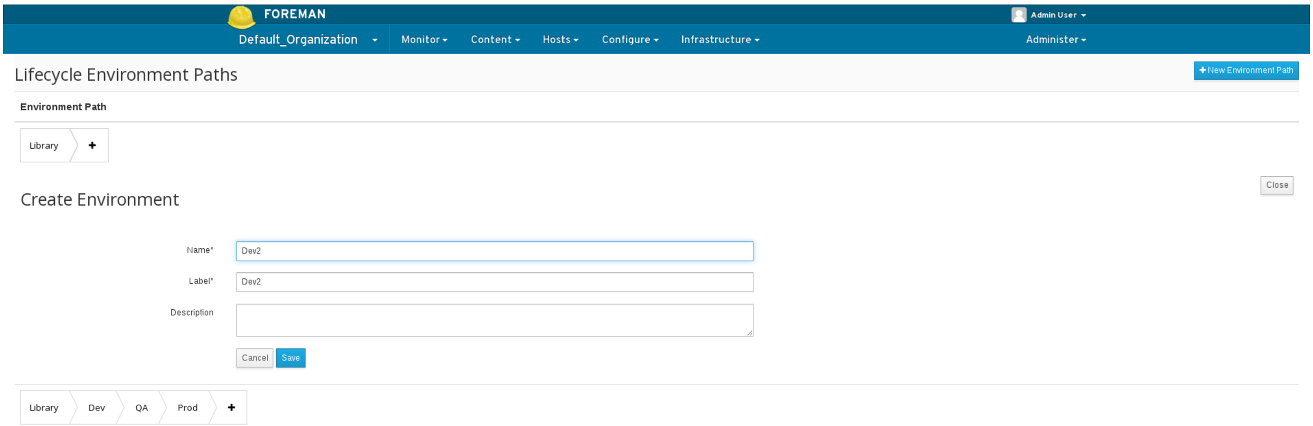
Name* Prod

Label* Prod

Description Producttion Environment

Creating a lifecycle environment path

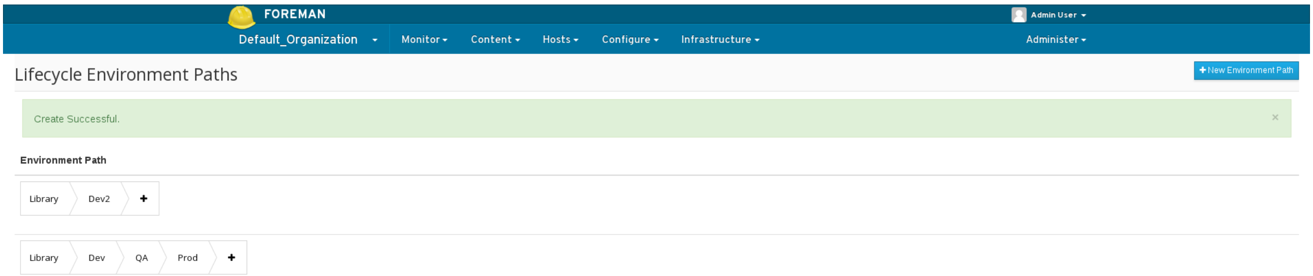
Click on the New Environment Path



Name* Dev2

Label* Dev2

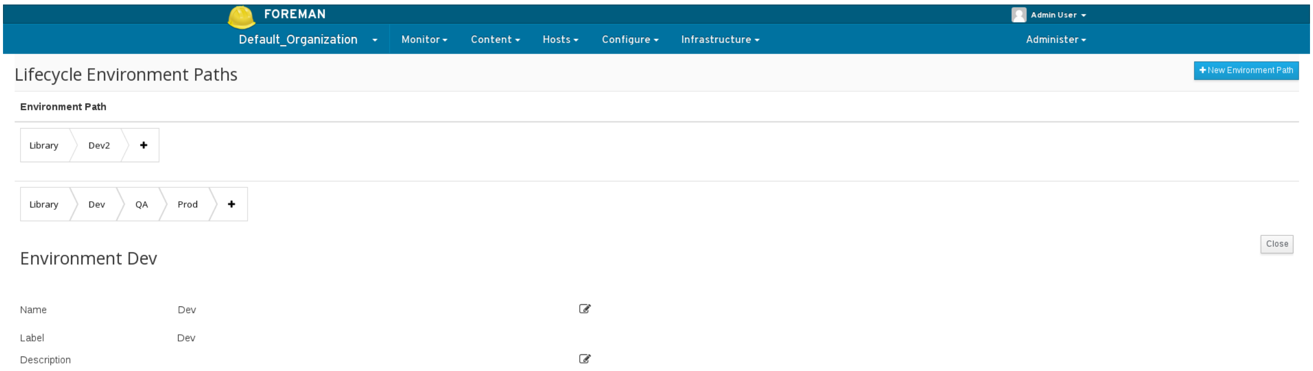
Description



Create Successful.

View/Updating environment name

Click on the name of the environment.



Lifecycle Environment Paths

Environment Path

Library › Dev2 › +

Library › Dev › QA › Prod › +

Environment Dev

Name	Dev
Label	Dev
Description	

Katello 3.15 Documentation

3.15 ▾

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.

If provisioning hosts using a synced CentOS 8 repository, the AppStream repository is needed for Kickstart to work. To set this up, create an “AppStream” repository within the same lifecycle environment and content view as the CentOS 8 repository and sync it. Katello will automatically add this AppStream repo to the Kickstart file. After creating the host using CentOS 8 content, double check that the generated Kickstart file mentions the AppStream repo. The host provisioning should proceed automatically as expected.

Note: Currently, CentOS 8 Anaconda requires that the AppStream repository is specifically named “AppStream”. To get around this, edit the Kickstart default provisioning template (or create a new one) to ensure this block:

```
<% @additional_media.each do |medium| -%>
repo --name <%= medium[:name] %> --baseurl <%= medium[:url] %> <%= medium[:install] ? ' --install' : '' %><%= proxy_string
%>
<% end -%>
```

will produce a line with `repo --name AppStream --baseurl <AppStream repo url>` .

For example, consider replacing that code block with:

```
<% @additional_media.each do |medium| -%>
<% if (medium[:url].include?("AppStream") and @host.operatingsystem.name == '< REPLACE ME >' and os_major >= 8) -%>
repo --name AppStream --baseurl <%= medium[:url] %>
<% else -%>
repo --name <%= medium[:name] %> --baseurl <%= medium[:url] %> <%= medium[:install] ? ' --install' : '' %><%= proxy_string
%>
<% end -%>
<% end %>
```

Related CentOS 8 issue

Katello 3.15 Documentation

3.15

Managing Puppet Content

Importing the Puppet Forge

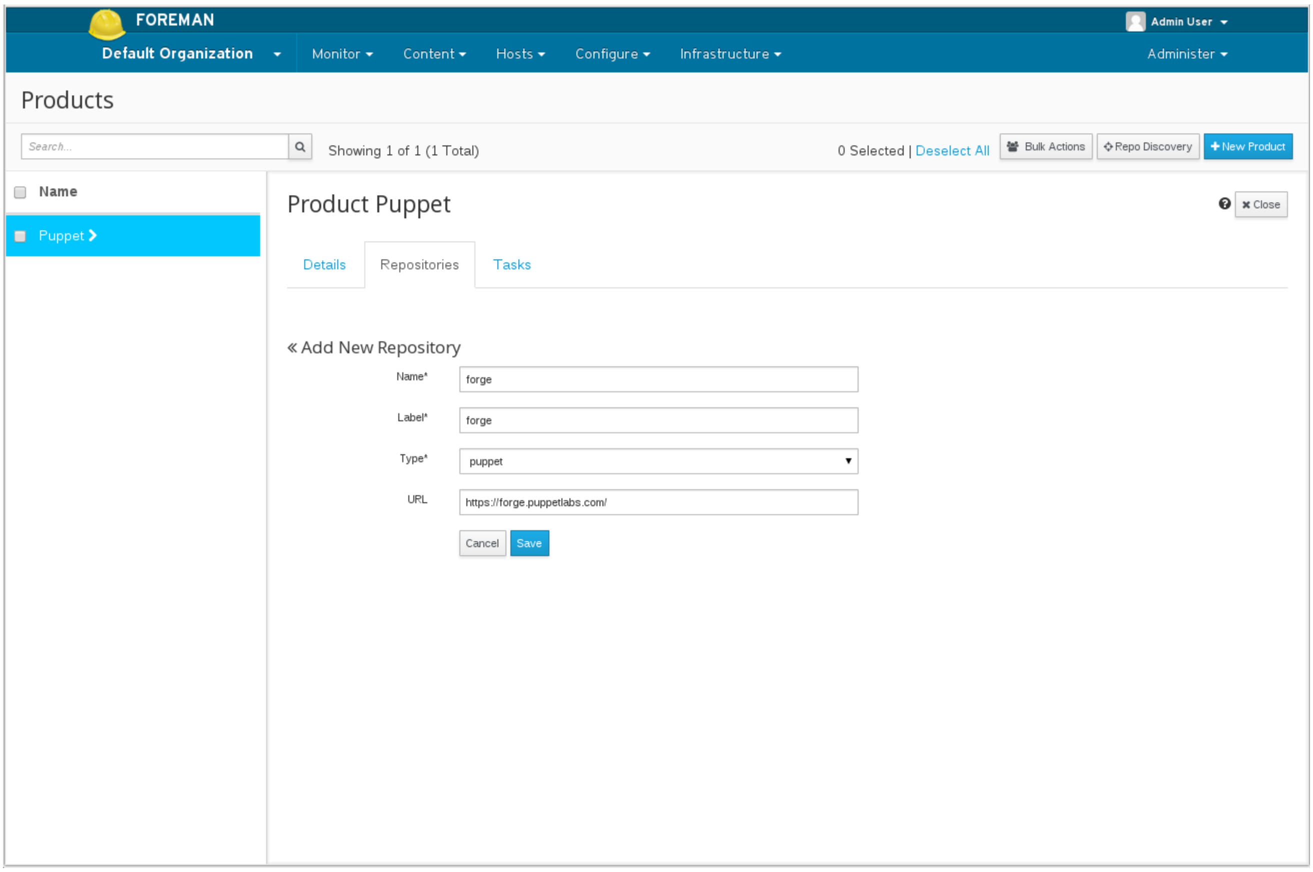
The Puppet Forge is a collection of puppet modules written by the community which can be used to manage hosts in Katello. These modules can be used in content views as described in the content views guide in order to configure the running hosts.

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 repostitory as shown:



Name* forge

Label* forge

Type* puppet

URL https://forge.puppetlabs.com/

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. If you have SELinux enabled, in order to sync from the file system, you’ll need to apply a label to the files in order for the system to access them. Two options are httpd_sys_r_content_t or pulp_tmp_t. Note: if you choose httpd_sys_r_content_t then the webserver can also read the files so that may or may not be good. One way to apply these labels would be to use the chcon command.

Next, 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.

Katello 3.15 Documentation

3.15

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

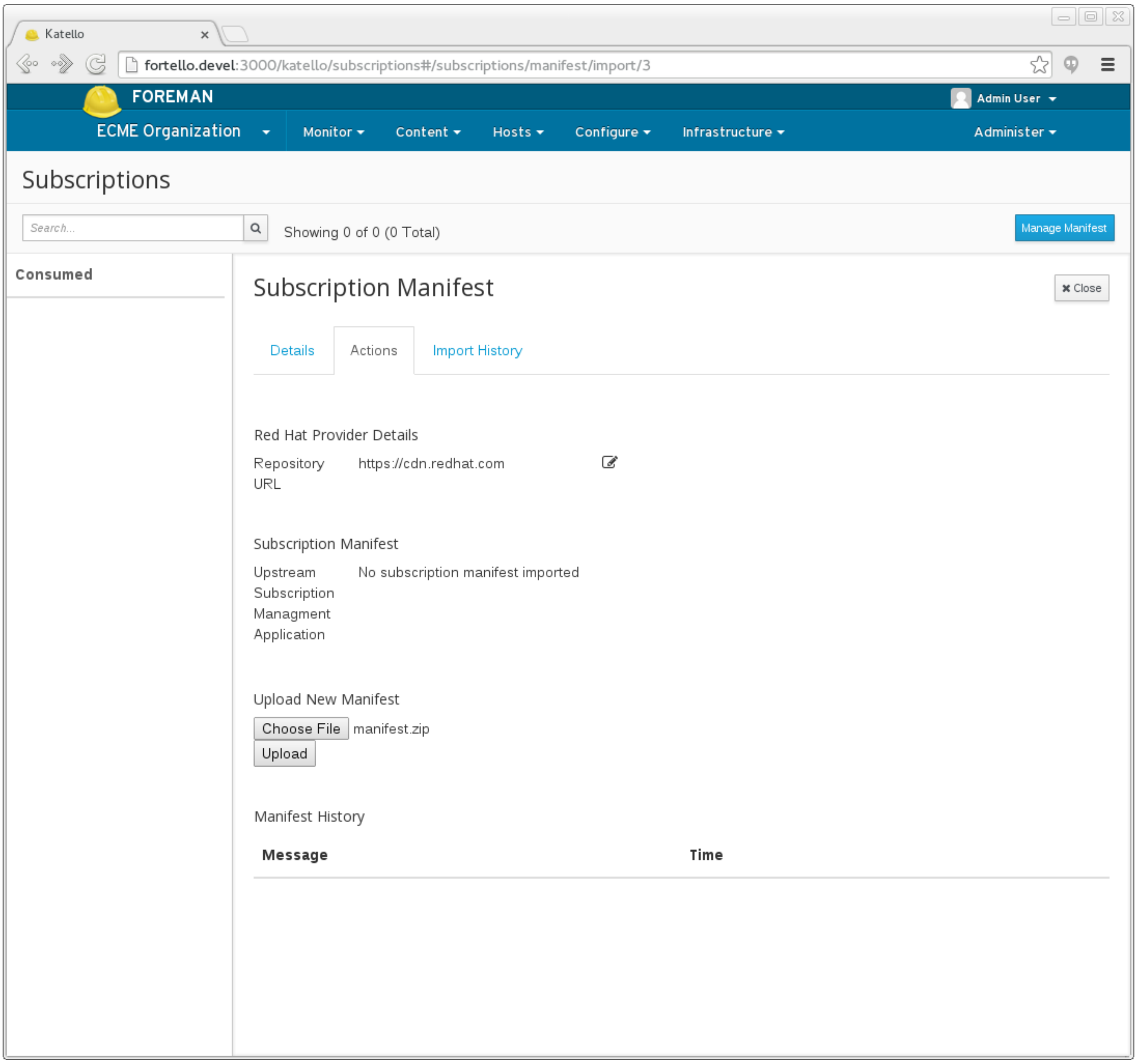
For details on how to create a subscription manifest, click here

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**



Subscriptions

Showing 0 of 0 (0 Total)

Subscription Manifest

Red Hat Provider Details

Repository URL https://cdn.redhat.com

Subscription Manifest

Upstream Subscription Managment Application No subscription manifest imported

Upload New Manifest

manifest.zip

Manifest History

Enable Red Hat Repositories

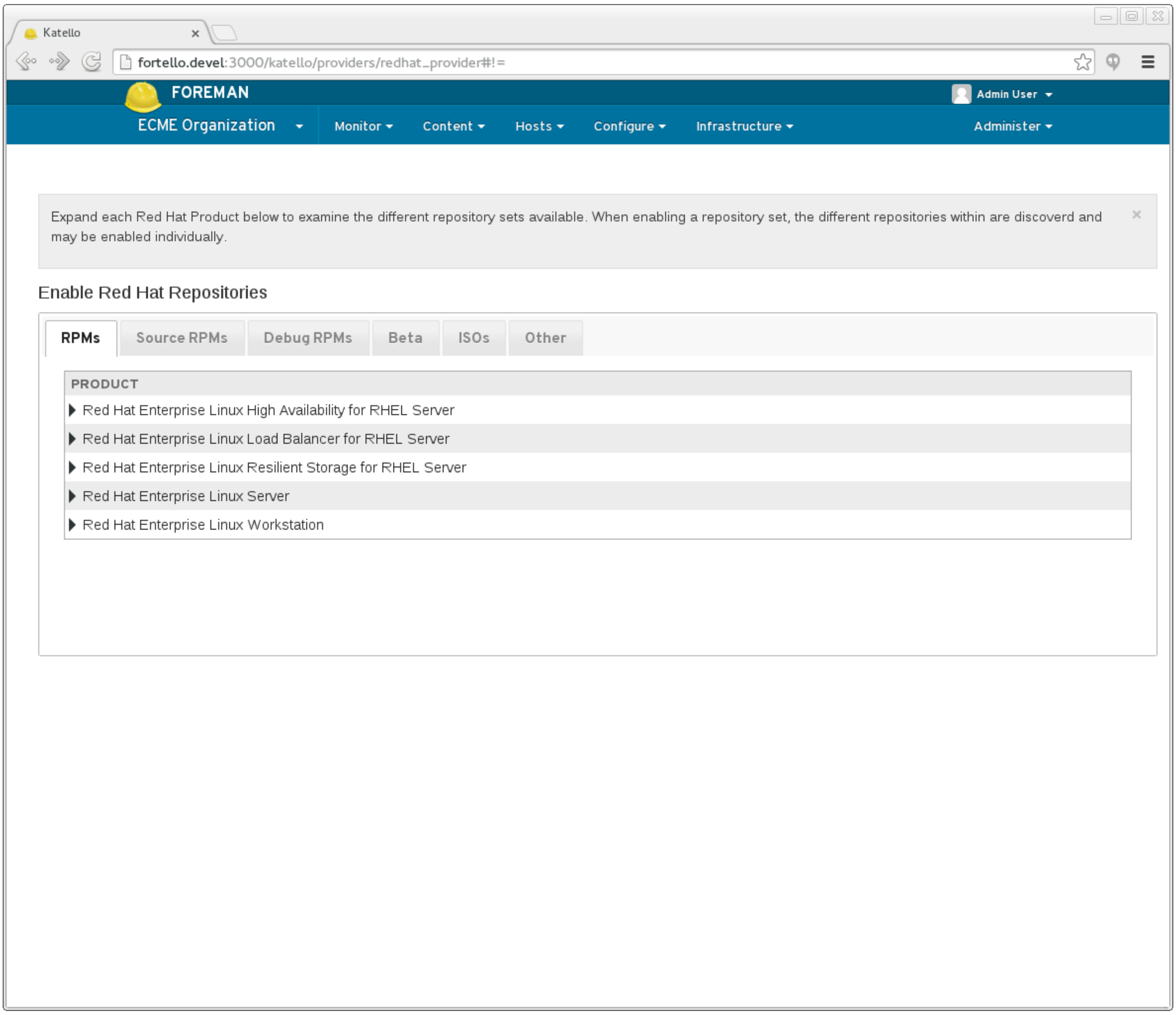
Once a subscription manifest is imported, access is available to potentially hundreds of Red Hat Repositories (e.g. Red Hat Enterprise Linux Server, Red Hat Enterprise Virtualization...etc). This process allows you to select only those that you are interested in for your enterprise.

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 Repsitory 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:

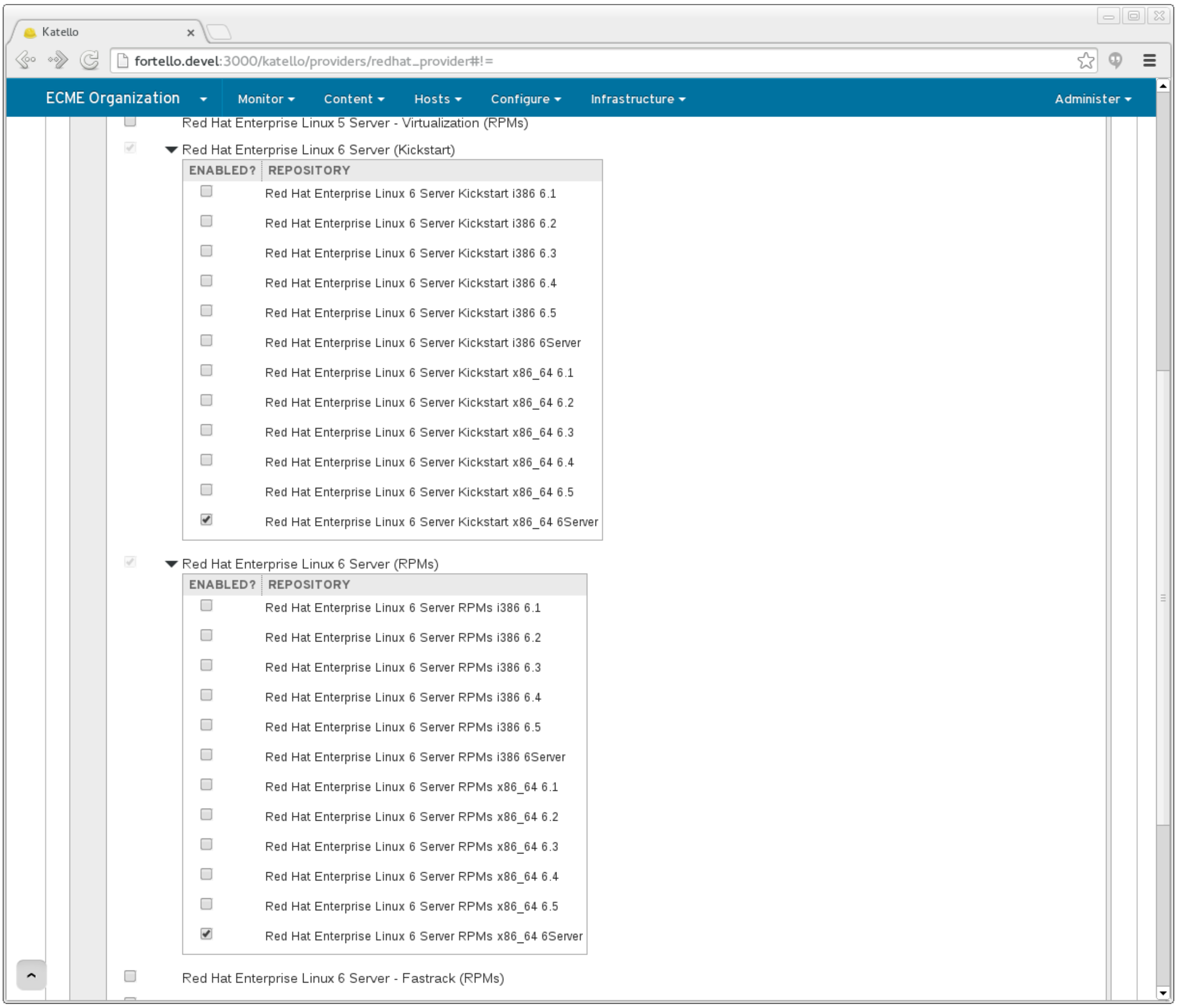
- When enabling a RHEL repository, Red Hat recommends selecting the Server repo (e.g. 6Server, 5Server) versus a specific release (e.g. 6.2). When a specific release is necessary, the preferred way is to create a Content View with filters that narrow the content to the desired version (e.g. 6.2)
- If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



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

Enable Red Hat Repositories

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



☐ Red Hat Enterprise Linux 5 Server - Virtualization (RPMs)

☑ Red Hat Enterprise Linux 6 Server (Kickstart)

ENABLED?	REPOSITORY
☐	Red Hat Enterprise Linux 6 Server Kickstart i386 6.1
☐	Red Hat Enterprise Linux 6 Server Kickstart i386 6.2
☐	Red Hat Enterprise Linux 6 Server Kickstart i386 6.3
☐	Red Hat Enterprise Linux 6 Server Kickstart i386 6.4
☐	Red Hat Enterprise Linux 6 Server Kickstart i386 6.5
☐	Red Hat Enterprise Linux 6 Server Kickstart i386 6Server
☐	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.1
☐	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.2
☐	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.3
☐	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.4
☐	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6.5
☑	Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server

☑ Red Hat Enterprise Linux 6 Server (RPMs)

ENABLED?	REPOSITORY
☐	Red Hat Enterprise Linux 6 Server RPMs i386 6.1
☐	Red Hat Enterprise Linux 6 Server RPMs i386 6.2
☐	Red Hat Enterprise Linux 6 Server RPMs i386 6.3
☐	Red Hat Enterprise Linux 6 Server RPMs i386 6.4
☐	Red Hat Enterprise Linux 6 Server RPMs i386 6.5
☐	Red Hat Enterprise Linux 6 Server RPMs i386 6Server
☐	Red Hat Enterprise Linux 6 Server RPMs x86_64 6.1
☐	Red Hat Enterprise Linux 6 Server RPMs x86_64 6.2
☐	Red Hat Enterprise Linux 6 Server RPMs x86_64 6.3
☐	Red Hat Enterprise Linux 6 Server RPMs x86_64 6.4
☐	Red Hat Enterprise Linux 6 Server RPMs x86_64 6.5
☑	Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server

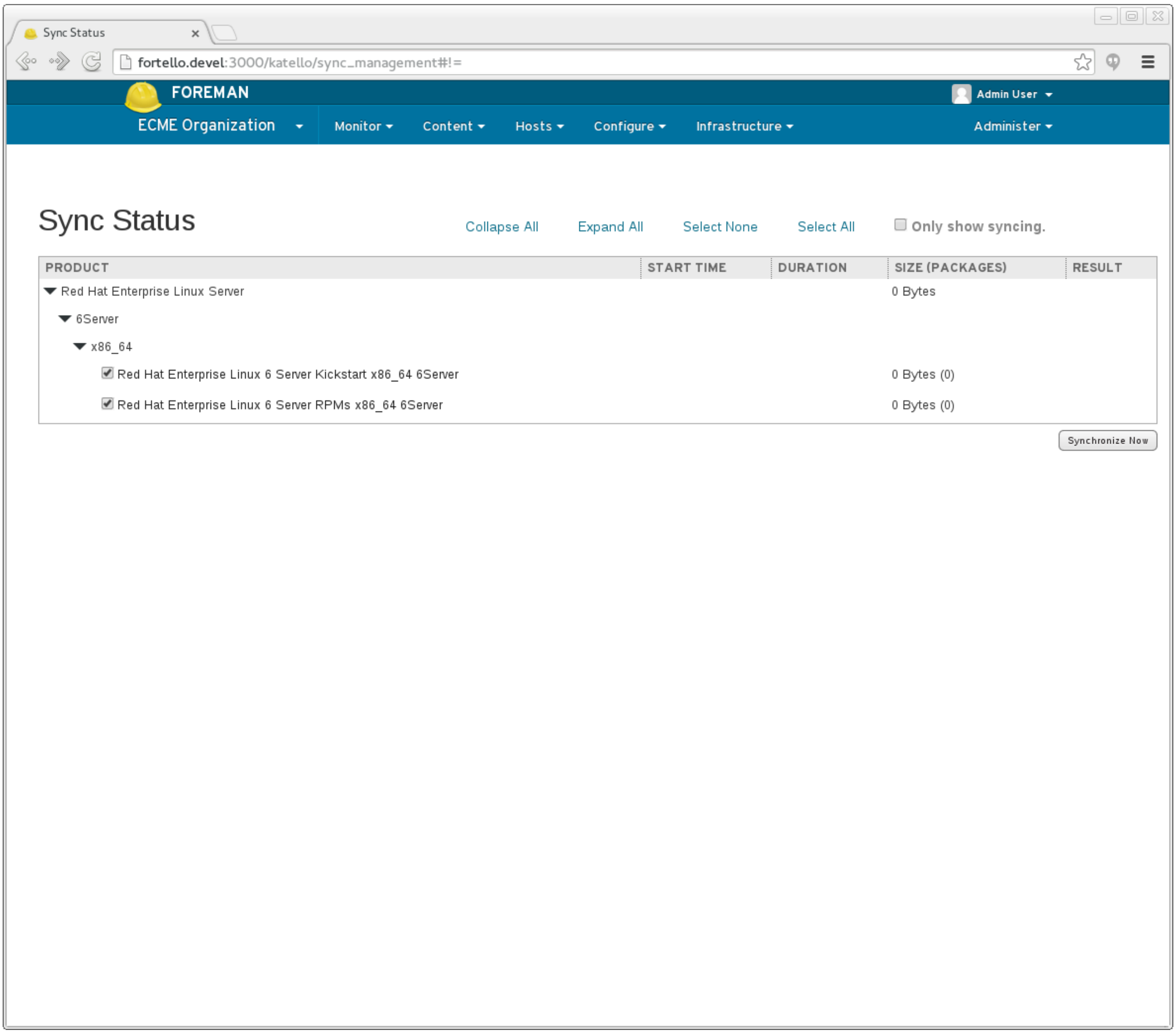
☐ Red Hat Enterprise Linux 6 Server - Fastrack (RPMs)

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**



Sync Status

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)	

Schedule Repository Synchronization

Creating a Sync Plan

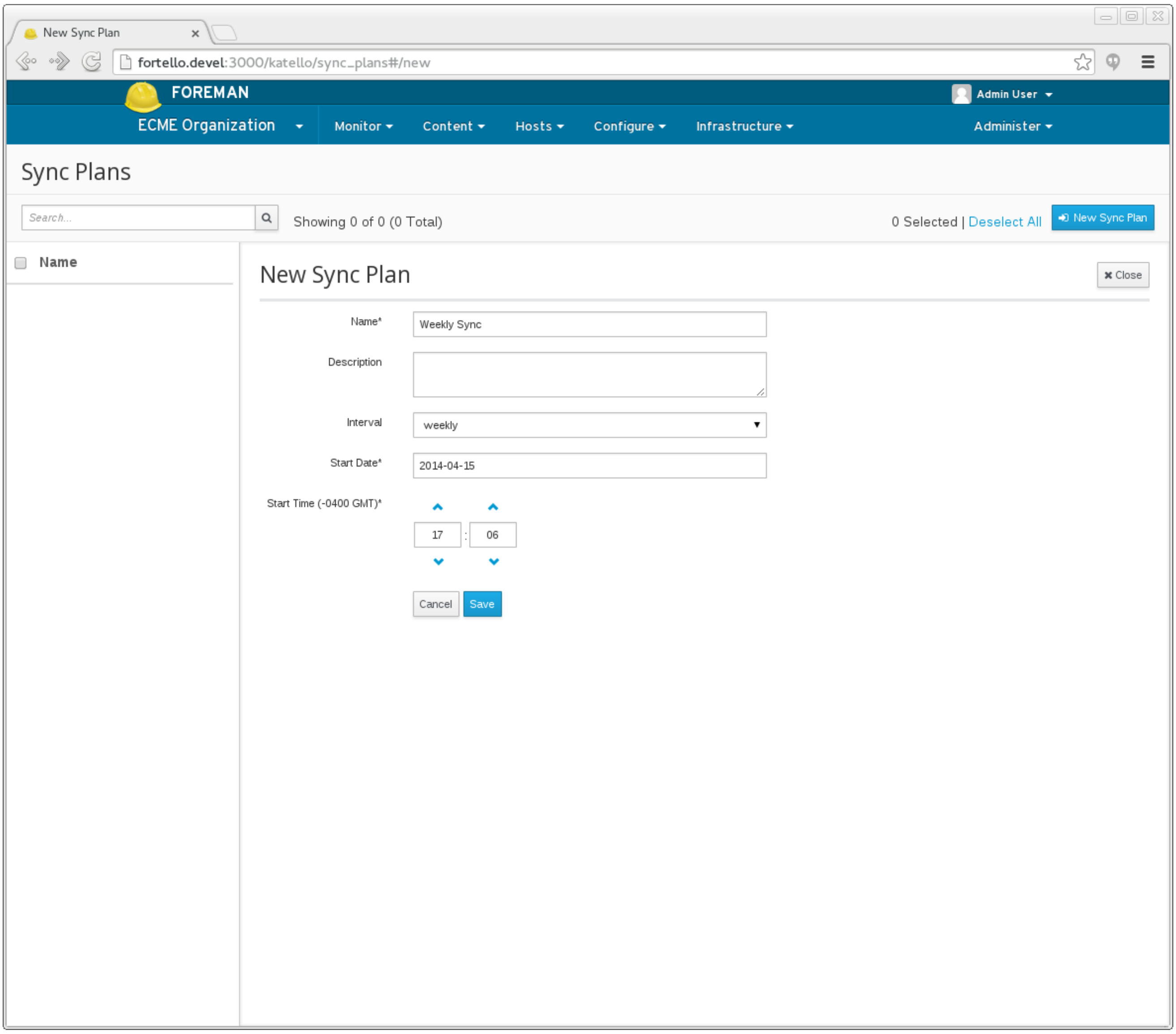
Sync plans provide the ability to schedule repository synchronization on a daily, weekly or a monthly basis. Sync plans can be applied individually or to a set of repositories.

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.



Sync Plans

Search...

Showing 0 of 0 (0 Total)

0 Selected | Deselect All

New Sync Plan

Name

New Sync Plan

Close

Name* Weekly Sync

Description

Interval weekly

Start Date* 2014-04-15

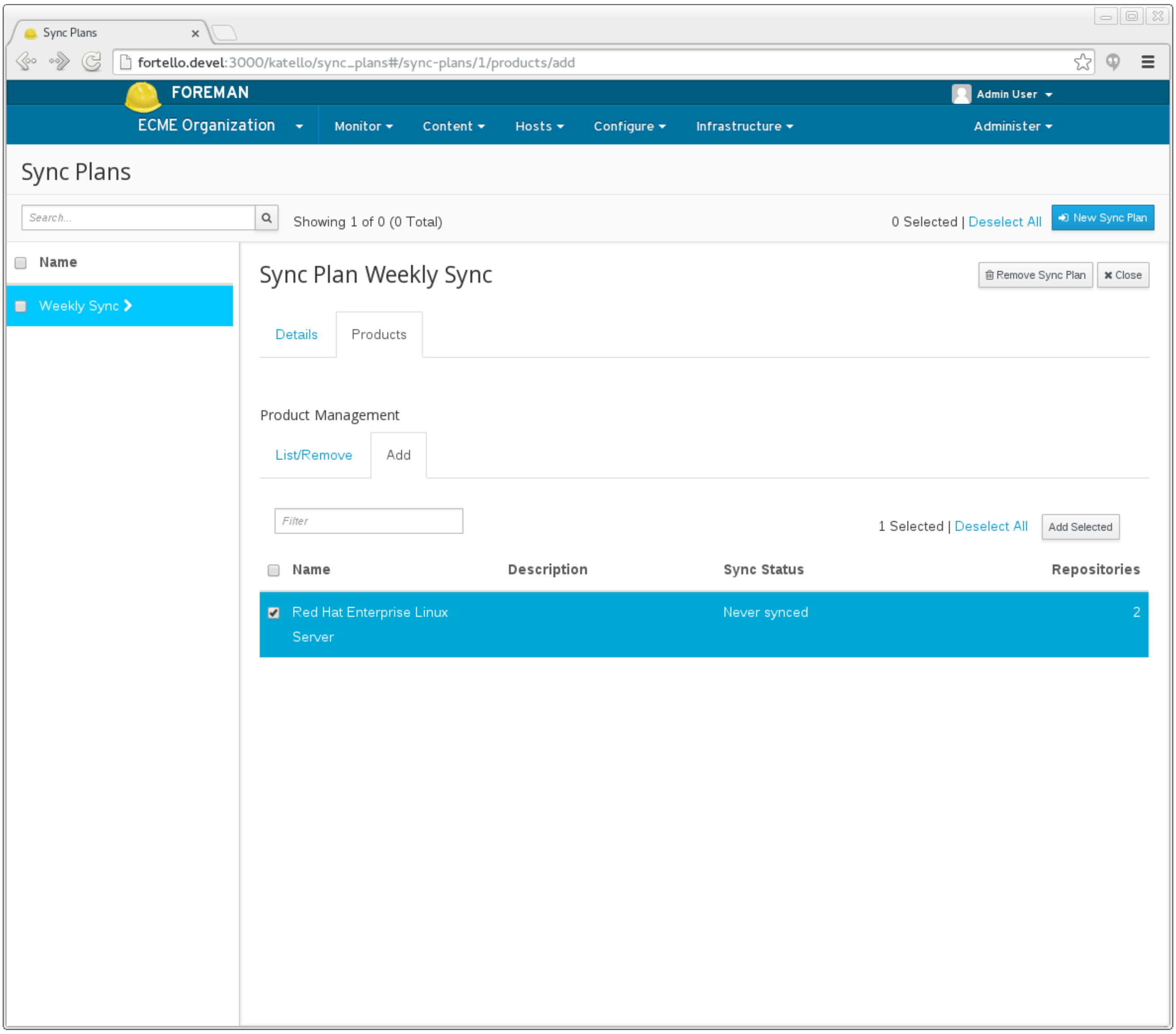
Start Time (-0400 GMT)* 17 : 06

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



Showing 1 of 0 (0 Total)

0 Selected | Deselect All

1 Selected | Deselect All

	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.

Katello 3.15 Documentation

3.15

SUSE Content

Katello can be used to manage content associated with SUSE products. This includes content such as RPMs, errata.

There are two different ways to use Katello to get manage SUSE products / repositories:

1. Use foreman_scc_manager
2. Use the smt tool

Definitions

- Repository - Collection of content (either rpm).
- 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.

Manage SUSE Content using the foreman_scc_manager

To manage SUSE Content witht the foreman_scc_manager you need to have an SUSE SCC account

General Workflow

- Install the foreman_scc_manager
- Set up the connection to your SCC account.
- Synchronize the list of available SUSE products.
- Select the products and sub-products which you want to add to Katello (Content > Products). This creates the products and all associated repositories.
- Synchronize the created repositories in Content > Products

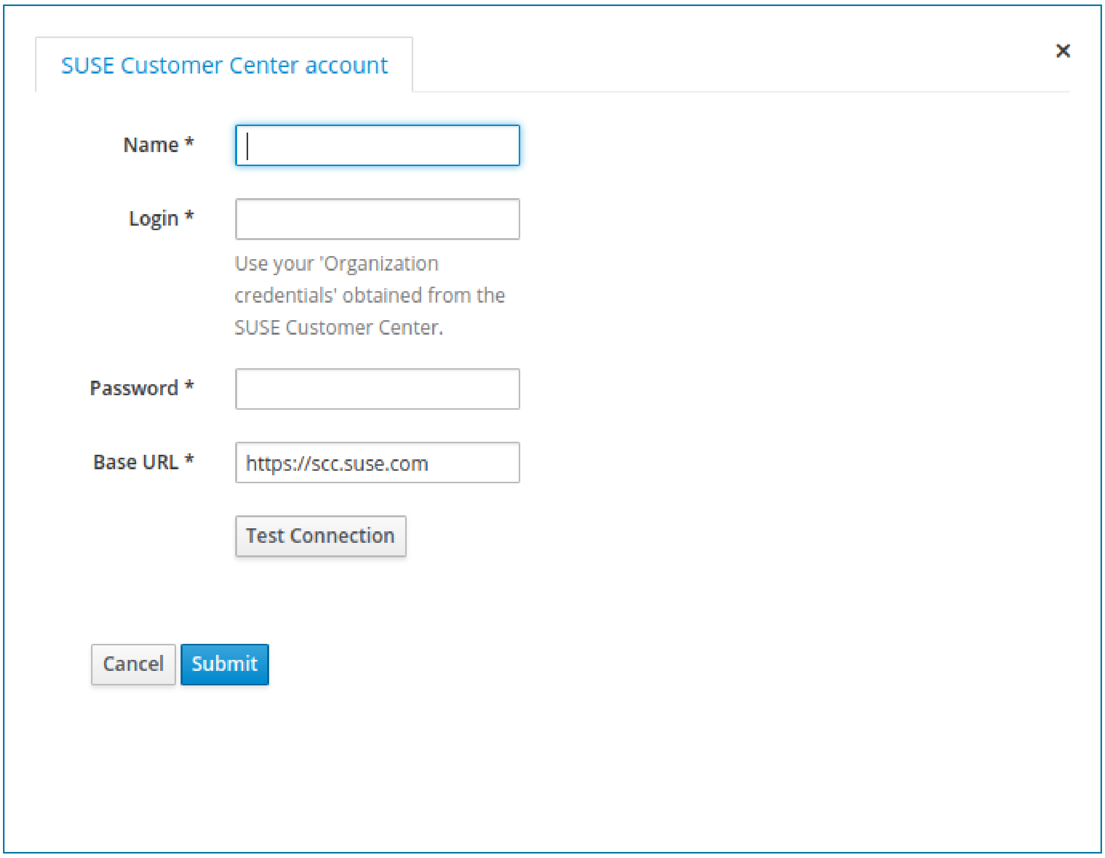
Installation

The installation of the foreman_scc_manager can be started with the following command. This will restart your Foreman!

```
yum install tfm-rubygem-foreman_scc_manager
```

Usage

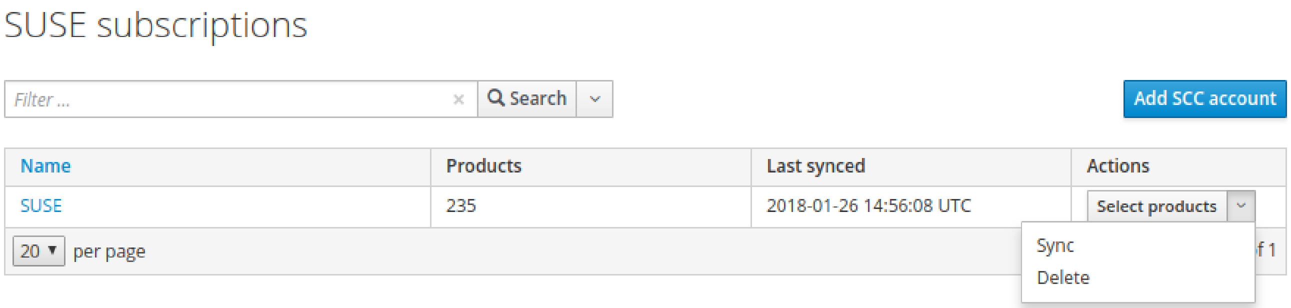
First of all, you need to add your SUSE SCC account. Use the button “Add SCC account” on the page “Content > SUSE Subscriptions” and configure your account. Verify that your account is configured correctly and the SUSE SCC portal is reachable with “Test Connection”.



Use your 'Organization credentials' obtained from the SUSE Customer Center.

Base URL * https://scc.suse.com

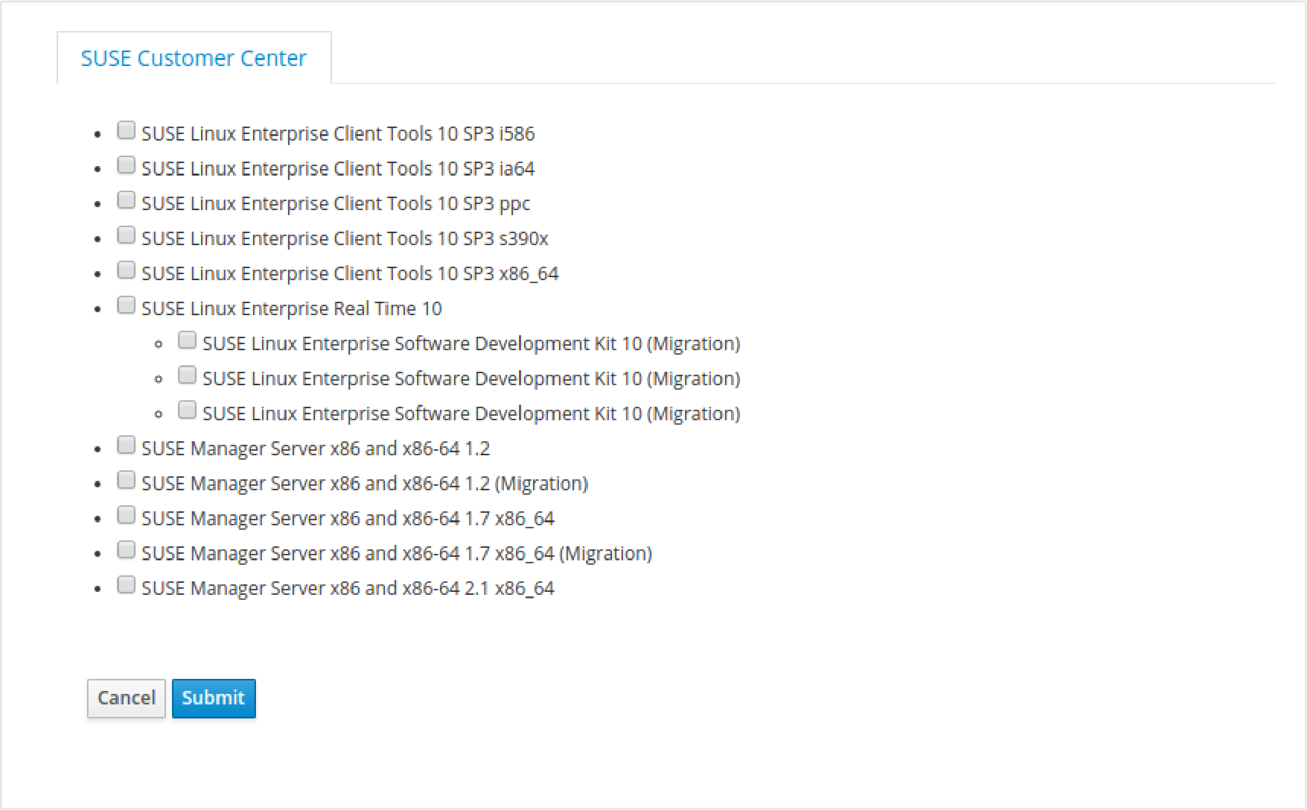
The next step is to select the action “Sync” in the “Actions” drop down list. This will start the process to synchronize all available products for your SCC account.



SUSE subscriptions

Name	Products	Last synced	Actions
SUSE	235	2018-01-26 14:56:08 UTC	Select products

Select the products which you want to include to Katello by the selecting the “Select products” in the “Actions” drop down list. This action will take some time as it will create a product and all repositories of each selected SUSE product.



SUSE Customer Center

- □ SUSE Linux Enterprise Client Tools 10 SP3 i586
- □ SUSE Linux Enterprise Client Tools 10 SP3 ia64
- □ SUSE Linux Enterprise Client Tools 10 SP3 ppc
- □ SUSE Linux Enterprise Client Tools 10 SP3 s390x
- □ SUSE Linux Enterprise Client Tools 10 SP3 x86_64
- □ SUSE Linux Enterprise Real Time 10
  - □ SUSE Linux Enterprise Software Development Kit 10 (Migration)
  - □ SUSE Linux Enterprise Software Development Kit 10 (Migration)
  - □ SUSE Linux Enterprise Software Development Kit 10 (Migration)
- □ SUSE Manager Server x86 and x86-64 1.2
- □ SUSE Manager Server x86 and x86-64 1.2 (Migration)
- □ SUSE Manager Server x86 and x86-64 1.7 x86_64
- □ SUSE Manager Server x86 and x86-64 1.7 x86_64 (Migration)
- □ SUSE Manager Server x86 and x86-64 2.1 x86_64

After the synchronization task (see Monitor > Tasks) is complete, you will see the list of selected products and its repositories in “Content > Products”. Please keep in mind, that you need to sync the repositories before you can actually use them. Feel free to remove repositories of certain products which you don’t need.

Manage SUSE Content via the smt tool

General Workflow

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

- Set up SUSE account and smt server
- Set up the smt mirroring and Content
- Create Products and Repositories

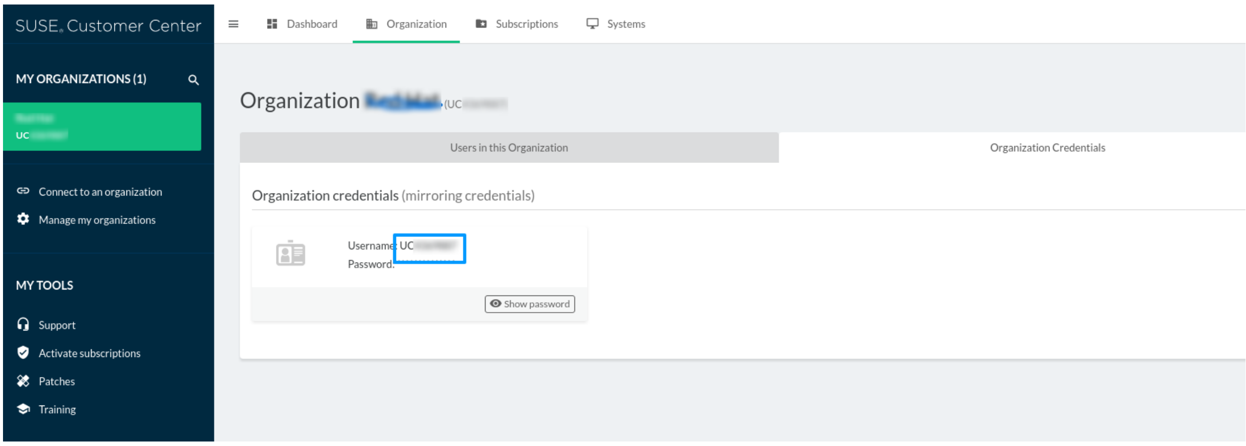
Set up SUSE account and SMT server

Assuming you have an SCC account, setup the SMT server, click here for more documentation.

- Install smt

```
# zypper install smt
# yast smt-server
```

- A TUI will pop up and you are going to have to add credentials from your account, found in scc.suse.com, as shown in screenshot below



[redacted] UC [redacted]

Organization [redacted] (UC [redacted])

Organization credentials (mirroring credentials)

Username: UC [redacted]
Password:

- Click ok and proceed with the steps provided, it will finish up and exit

Set up the SMT Content and mirroring

On a SLES 12 Box

- Sync repo data

```
# smt-sync
```

- List Repositories

```
# smt-repos
```

- Install the repositories you need to mirror via the smt-repos -e flag. For example to mirror SLES12-SP3-Pool

```
# smt-repos -e SLES12-SP3-Pool
```

- Check the enabled repos

```
# smt-repos -o
```

- Mirror the repositories

```
# smt-mirror
```

- This should install an apache2 on that host

```
# service apache2 restart
```

  You should be able to navigate to the mirror by browsing to `http://<fqdn>/repo/`

Create Products and Repositories

- Follow the steps listed here to create Products and Repositories
- Follow the steps listed on the same link as above to create a yum repository with the repository url pointing to the appropriate location. For example `http://<fqdn>/repo/SUSE/Products/SLE-SERVER/12-SP3/x86_64/product/`

Katello 3.15 Documentation

3.15

Setup Remote Databases in Foreman with Katello Plugin Installed

Foreman with Katello plugin can be installed with remote databases for both postgresql and mongo. These instructions are for a Foreman with Katello plugin server, where remote databases are currently supported. This guide will refer to the server as “Foreman”, with the assumption that the Katello plugin is installed.

High level

There are two ways to deploy Foreman with remote databases:

1. Fresh install
   - prepare Postgres server with databases for Foreman and Candlepin and dedicated users owning them
   - prepare Mongo DB with user owning the pulp_database
   - prepare box where the Foreman will be installed and make sure the databases are accessible from the box
   - run foreman-installer with right parameters pointing to the databases
2. Migration of existing Foreman installation
   - prepare Postgres server with databases for Foreman a Candlepin and dedicated users owning them
   - prepare Mongo DB with user owning the pulp_databse
   - make sure the databases are accessible from the box where Foreman is installed
   - shut down the services except the dbs you want to move (mongod, postgresql)
   - dump the DBs
   - restore the DBs on remote servers
   - run foreman-installer with right parameters pointing to the databases. It re-configures the databases and start all the services with new DB locations

Important note:

- Pulpcore installs a separate PostgreSQL database. The installer lacks support for external databases. This support is introduced in Katello 3.16.

- Foreman ([since 2.0])(https://theforeman.org/manuals/2.0/index.html#Headlinefeatures) and Pulpcore require PostgresSQL 10+. Databases should be upgraded to avoid conflicts. It is recommended to update to version 12.

In either scenario, both of the databases don't have to be remote. You can opt to use only a remote mongo database or only a remote postgresql database. Both postgresql and mongo databases can be on the same host, but this isn't recommended due to the amount of resources mongo can use.

Prepare remote Postgres

GOAL: To use remote Postgres database with Foreman we have to:

- be able to access the databases from foreman box
- the database user we use to connect to the database needs to own the database, i.e. it can create, alter and delete the tables, indexes and constraints. Note it is not required to be able to create the database itself.

Install Postgres

Warning: This is just minimal testing setup which is not suitable for production, please adjust the settings to your environment as needed.

Assume our postgres server has hostname `postgres.example.com` .

First, we install postgresql.

```
yum install -y postgresql-server postgresql-contrib
postgresql-setup initdb
systemctl enable --now postgresql
```

Now we need to make Postgres listen to inbound connections, please adjust these parameters to your own networking and security requirements.

Edit `/var/lib/pgsql/data/postgresql.conf` . Uncomment `listen_address` and modify its value to look like:

```
listen_address = "*"
```

The next step we need to take is to add a proper client authentication for remote client to our postgres server. To achieve the same, edit `/var/lib/pgsql/data/pg_hba.conf` . Append the following line at the end of the file

```
host    all    all    <katello.example.com ip>/24    md5
```

Now restart the postgres service for changes to take effect

```
systemctl restart postgresql
```

Create the databases

Switch the user role to postgres and start postgres client

```
su - postgres -c psql
```

Once inside the client, we need to create two databases and dedicated roles, one for foreman and one for candlepin.

```
CREATE USER "foreman" WITH PASSWORD '<FOREMAN_PASSWORD>';
CREATE USER "candlepin" WITH PASSWORD '<CANDLEPIN_PASSWORD>';
CREATE DATABASE foreman OWNER foreman;
CREATE DATABASE candlepin OWNER candlepin;
```

Test it works

From `katello.example.com` test the DB is accessible:

```
PGPASSWORD='<FOREMAN_PASSWORD>' psql -h postgres.example.com  -p 5432 -U foreman -d foreman -c "SELECT 1 as ping"
```

```
PGPASSWORD='<CANDLEPIN_PASSWORD>' psql -h postgres.example.com  -p 5432 -U candlepin -d candlepin -c "SELECT 1 as ping"
```

If there are no errors we are done with database preparation.

Prepare remote Mongo

GOAL: To use remote Mongo database with Foreman we have to:

- be able to access the databases from foreman box
- the database user we use to connect to the database needs to own the database

Install Mongo DB

Warning: This is just minimal testing setup which is not suitable for production.

Assume our Mongo server has hostname `mongo.example.com` . Install and enable Mongo server

```
yum install -y centos-release-scl
yum install -y rh-mongodb34-syspaths
```

Enable authentication in `/etc/mongod.conf`

```
auth=true
```

Enable and start the service

```
systemctl enable --now mongod
```

Create Pulp user and database

```
mongo admin -u admin -p admin --eval "db.createUser({user:'pulp',pwd:'<PULP_PASSWORD>',roles:[{role:'dbOwner', db:'pulp_database'},{ role: 'readWrite', db: 'pulp_database'}]})"
```

Test it works

From `katello.example.com` test the mongo DB is accessible:

```
mongo --host mongo.example.com -u pulp -p <PULP_PASSWORD> --port 27017 --eval 'ping:1' pulp_database
```

If there are no errors we are done with database preparation.

Fresh install

Install katello package

We assume the box where the Foreman server will be installed has hostname `katello.example.com` .

Follow the documentation to install the `katello` package and **do not run foreman-installer.** We need to use the remote database flags with the installer. Use the following steps once the katello rpm is installed.

Prepare remote databases

Follow the instructions to prepare remote mongo and prepare remote postgres to make the remote database servers ready for installation.

Run the installer

To install and configure Foreman we just need to run

```
foreman-installer --scenario katello \
  --foreman-db-host postgres.example.com \
  --foreman-db-password <FOREMAN_PASSWORD> \
  --foreman-db-database foreman \
  --foreman-db-manage false
  --katello-candlepin-db-host postgres.example.com \
  --katello-candlepin-db-name candlepin \
  --katello-candlepin-db-password <CANDLEPIN_PASSWORD> \
  --katello-candlepin-manage-db false \
  --katello-pulp-db-username pulp \
  --katello-pulp-db-password <PULP_PASSWORD> \
  --katello-pulp-db-seeds “mongo.example.com:27017” \
  --katello-pulp-db-name pulp_database
  --katello-pulp-manage-db false
```

Note: for more related options and tips on SSL configuration see Full list of options

Migration of existing Foreman

Migrating an existing installation to remote databases can take time, so plan for some outage time (length depending on database size) while a backup is taken and the databases are migrated.

In this example, we assume that Foreman was installed and is running on `katello.example.com` .

Prepare remote databases

Follow the instructions to prepare remote mongo and prepare remote postgres to make the remote database servers ready for migration.

Stop the Foreman server

Stop the Foreman related services to minimize risk of the data changes during the migration

```
foreman-maintain service stop --exclude postgresql,rh-mongodb34-mongod
```

Dump databases

Dump the local databases

```
foreman-maintain backup online --skip-pulp-content --preserve-directory -y /tmp/migration_backup
```

Restore data in remote databases

You can restore the SQL dumps to the remote databases from the foreman system.

```
PGPASSWORD='<FOREMAN_PASSWORD>' pg_restore -h postgres.example.com -U foreman -d foreman < /tmp/migration_backup/foreman.dump
PGPASSWORD='<CANDLEPIN_PASSWORD>' pg_restore -h postgres.example.com -U candlepin -d candlepin < /tmp/migration_backup/candlepin.dump
mongorestore --host mongo.example.com --db pulp_database --username pulp --password <PULP_PASSWORD> /tmp/migration_backup/mongo_dump
```

Now the copy of the local database is also at the remote locations.

Update the configuration

To update existing configuration of Foreman we just need to run

```
foreman-installer --scenario katello \
  --foreman-db-host postgres.example.com \
  --foreman-db-password <FOREMAN_PASSWORD> \
  --foreman-db-database foreman \
  --foreman-db-manage false \
  --katello-candlepin-db-host postgres.example.com \
  --katello-candlepin-db-name candlepin \
  --katello-candlepin-db-password <CANDLEPN_PASSWORD> \
  --katello-candlepin-manage-db false \
  --katello-pulp-db-username pulp \
  --katello-pulp-db-password <PULP_PASSWORD> \
  --katello-pulp-db-seeds “mongo.example.com:27017” \
  --katello-pulp-db-name pulp_database \
  --katello-pulp-manage-db false
```

The installer start services aside from the database related services. Everything should be up and ready at this point, and you can clean up the local databases if you would like.

Full list of remote database related options in the installer

Use `foreman-installer --full-help` for all up-to-date installer options

Foreman database related:

```
--foreman-db-manage           if enabled, will install and configure the database server on this host
--foreman-db-database         Database 'production' database (e.g. foreman)
--foreman-db-host             Database 'production' host
--foreman-db-password         Database 'production' password, default is randomly generated
--foreman-db-pool             Database 'production' size of connection pool (current: 5)
--foreman-db-port             Database 'production' port
--foreman-db-root-cert        Root cert used to verify SSL connection to postgres
--foreman-db-sslmode          Database 'production' ssl mode (disable|allow|prefer|require|verify-full)
--foreman-db-username         Database 'production' user (e.g. foreman)
```

Candlepin database related:

```
--katello-candlepin-db-host   Host with Candlepin DB
--katello-candlepin-db-name   Name of the Candlepin DB
--katello-candlepin-db-password  Candlepin DB password
--katello-candlepin-db-port   Port accepting connections to Candlepin DB
--katello-candlepin-db-ssl    Boolean indicating if the connection to the database should be over
--katello-candlepin-db-ssl-verify  Boolean indicating if the SSL connection to the database should be verified
--katello-candlepin-db-user   Candlepin DB user
--katello-candlepin-manage-db  Boolean indicating whether a database should be installed, this includes db creation and user
```

Mongo database related:

```
--katello-pulp-db-ca-path     The ca_certs file contains a set of concatenated "certification authority" certificates,
--katello-pulp-db-name        Name of the database to use
--katello-pulp-db-password    The password to use for authenticating to the MongoDB server
--katello-pulp-db-replica-set  The name of replica set configured in MongoDB, if one is in use
--katello-pulp-db-seeds       Comma-separated list of hostname:port of database replica seed hosts
--katello-pulp-db-ssl         Whether to connect to the database server using SSL.
--katello-pulp-db-ssl-certfile  The certificate file used to identify the local connection against mongod.)
--katello-pulp-db-ssl-keyfile  A path to the private keyfile used to identify the local connection against mongod. If
--katello-pulp-db-unsafe-autoretry  If true, retry commands to the database if there is a connection error.
--katello-pulp-db-username    The user name to use for authenticating to the MongoDB server
--katello-pulp-db-verify-ssl  Specifies whether a certificate is required from the other side of the connection, and
--katello-pulp-db-write-concern  Write concern of 'majority' or 'all'. When 'all' is specified, 'w' is set to number of
```

The actual option names may vary between versions. Check the actual naming with foreman-installer –full-help.

SSL configuration

Here is sample installer command that sets up Postgres databases with SSL verification. The Postgres server has its own CA. The CA cert used by Candlepin needs to be stored in system trust (`/etc/pki/java/cacerts`) as there is no other way to pass it to Candlepin

```
foreman-installer -S katello \
  --foreman-admin-password changeme \
  --foreman-db-host postgres.example.com \
  --foreman-db-password foreman \
  --foreman-db-database foreman_2 \
  --foreman-db-root-cert /etc/pki/ca-trust/source/anchors/ca-chain.cert.pem \
  --foreman-db-sslmode verify-full \
  --katello-candlepin-db-host postgres.example.com \
  --katello-candlepin-db-name candlepin_2 \
  --katello-candlepin-db-password candlepin \
  --katello-candlepin-db-ssl true \
  --katello-candlepin-manage-db false
```

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.

Katello 3.15 Documentation

3.15 ▾

Content View Import/Export

Content view import/export is one of the new features in Katello 3.9. This feature is for users who want the *exact* same content view version on one Katello instance to be available on another Katello instance. The other Katello instances may or may not have Internet access.

System administrators the ability to have fine-grained control over their content view versions, and they can have the same content view on two or more Katello instances. Additional APIs now exist to allow for this, as well as new commands to the hammer CLI tool.

This new feature works differently than the old export/import feature. The old feature is still available but has been deprecated.

- In earlier versions of Katello, you could only export yum repositories. You could export an entire content view version, but this simply exported each yum repository in the version without any additional metadata.

API additions

Katello 3.9 allows users to publish content view versions with a list of packages. This overrides any filters already set on the content view. It is meant for users who want to say “give me exactly what I want in this content view, with these exact RPMs that cannot be substituted”. Errata will be pulled in based on the RPM list. For example, if you include “walrus-0.71.noarch.rpm” and there is an errata that includes that exact package, the errata will be pulled in.

Here is an example. You would POST this to `/katello/api/v2/content_views/<id>/publish` as the `repos_units` parameter:

```
[
    {
        "label": "zoo",
        "rpm_filenames": [
            "walrus-5.21-1.noarch.rpm",
            "gorilla-0.62-1.noarch.rpm"
        ]
    },
    {
        "label": "a_longer_label",
        "rpm_filenames": [
            "facter-2.4.6-3.el7sat.x86_64.rpm",
            "pulp-rpm-handlers-2.13.4.9-1.el7sat.noarch.rpm"
        ]
    }
]
```

- You can also set the `major` and `minor` versions when publishing. For example, if you called the `/publish` API with `major=55` and `minor=4` , the content view would be version `55.4` .
- Using `major` , `minor` , and `repos_units` , you can create a content view version that exactly matches the content on another Katello, with the same version number.

Note

If you call ***/publish*** with the ***repos_units*** parameter set and also have content view filters set, the***repos_units*** will override any filters. This is intentional. A message will be logged to ***/var/log/foreman/production.log*** if the filter is overriden by the ***repos_units*** parameter.

Hammer additions

- The Hammer CLI tool has two new commands: `hammer content-view version export` and `hammer content-view version import` .

Hammer export

- The `hammer content-view version export` command gathers information about a content view version, and then creates a tar file with that information. It will first create a json file with information about the content view. Here is an example:

```
{
  "name": "my-cv",
  "major": 1,
  "minor": 0,
  "repositories": [
    {
      "id": 2,
      "label": "zoo",
      "content_type": "yum",
      "backend_identifier": "1-my-cv-v1_0-da3c4462-7343-4e30-bbd8-a802aa64be63",
      "relative_path": "Default_Organization/content_views/my-cv/1.0/custom/test/zoo",
      "on_disk_path": "/var/lib/pulp/published/yum/https/repos/Default_Organization/content_views/my-cv/1.0/custom/test/zoo",
      "rpm_filenames": [
        "bear-4.1-1.noarch.rpm",
        "camel-0.1-1.noarch.rpm",
        "cat-1.0-1.noarch.rpm",
        "cheetah-1.25.3-5.noarch.rpm",
        "chimpanzee-0.21-1.noarch.rpm",
        "cockateel-3.1-1.noarch.rpm",
        "cow-2.2-3.noarch.rpm",
        "crow-0.8-1.noarch.rpm",
        "dog-4.23-1.noarch.rpm",
        "dolphin-3.10.232-1.noarch.rpm",
        "duck-0.6-1.noarch.rpm",
        "elephant-8.3-1.noarch.rpm",
        "fox-1.1-2.noarch.rpm",
        "frog-0.1-1.noarch.rpm",
        "giraffe-0.67-2.noarch.rpm",
        "gorilla-0.62-1.noarch.rpm",
        "horse-0.22-2.noarch.rpm",
        "kangaroo-0.2-1.noarch.rpm",
        "lion-0.4-1.noarch.rpm",
        "mouse-0.1.12-1.noarch.rpm"
      ],
      "errata_ids": [
        "RHEA-2012:0003",
        "RHEA-2012:0001",
        "RHEA-2012:0004",
        "RHEA-2012:0002"
      ]
    }
  ]
}
```

Important

The *errata_ids* field is informational only.**ALL** errata in the repository are exported. A process during the import will then clean up errata that are not used.

The hammer command will also create a tar file that contains all of the repositories listed. The final result of the command is a tar file that contains two files: the json, and an inner tar file with all of the repositories. This tar file can be copied to a USB key and used for the `import` command.

Note

Older versions of Katello relied on the Pulp*export_distributor* and *group_export_distributor* to create an ISO image with the yum repositories. Katello would start a server-side task, create an ISO, and then copy the ISO to */var/lib/pulp/katello-export*. This process could take many hours and hundreds of GB of disk space for temporary files. Users can now use hammer to create the tar file. This lets us avoid creating temporary copies of extremely large files.

Hammer import

The `hammer content-view version import` command uses the tar file created from the `export` command to create a content view version with the same data. It will create a content view version with the same major and minor version numbers, and the same repositories with the same packages and errata.

Before you run the `import` command for the first time, you will need to create the same products on the importing Katello that you had on the exporting Katello. This is a step you will only need to do once. You will also need to create the same content view, with same label. Again, you will only need to do this once.

The `import` command will synchronize the packages from the export tar file into Library. It will then call the `/publish` API and create a new content view version using those packages.

Important

You will need to make sure Katello and Pulp can both read the tar file. If it cannot, you may get an error. The error will be logged in ***/var/log/foreman/production.log*** for Katello, or in ***/var/log/messages*** for Pulp. Ownership of the directory and files should be*Apache* with *system_u:object_r:httpd_sys_rw_content_t:s0* as the SELinux context.

Note

The import process will import all errata from an export. It will then purge any errata that are not associated with packages. This is the same process that is used today when copying RPMs between repositories.

Import/Export Best Practices

The intent of import/export is to capture a content view version on one Katello, and then re-create it on another Katello. The feature does *not* replicate a standard operating environment (SOE) from one Katello to another. A standard operating environment includes a manifest file, content view definitions, products, repositories, activation keys, host groups, and other information.

Please use foreman-ansible-modules or Hammer scripts to define your SOE in a reproducible way. Once you have a reproducible SOE, you can then use import/export to keep your Katello updated.

Note

To ensure proper SELinux contexts on the importing tar and files, use the `/var/lib/pulp/katello-export` directory on the importing Katello. This directory already has the correct permissions and correct SELinux labels, and was created specifically as a landing place for files not created by Pulp that Pulp needs to read or write. If choosing to use a different directory please see the alert at the end of the import section for proper permissions and SELinux context settings.

Katello 3.15 Documentation

3.15

Source RPMs

Listing source RPMs with Hammer CLI

- SRPM info can be retrieved in a few ways: `hammer srpm list` and `hammer srpm info`.
- The `hammer srpm list` command lists source RPMS by default across the entire Katello server. This can be filtered down by organization, product, repository, content-view and lifecycle environment.
- The `hammer srpm info` command gathers information about a source RPM. Returning ID, Name, Version, Architecture, Epoch, Release, Filename, and Description.

Uploading SRPMs with Hammer CLI

Create a product and repository

First we want to create a product:

```
# hammer product create --name Zoo --organization-id 1

Product created.
```

Next we want to grab our product id for the repository creation step:

```
# hammer product list --organization-id 1

---|------|-------------|----------------------|--------------|-----------
ID | NAME | DESCRIPTION | ORGANIZATION         | REPOSITORIES | SYNC STATE
---|------|-------------|----------------------|--------------|-----------
1  | Zoo  |             | Default Organization | 0            |
---|------|-------------|----------------------|--------------|-----------
```

Next we will create our repo:

```
# hammer repository create --product-id 1 --organization-id 1 --content-type yum --name source_rpms --publish-via-http yes

Repository created.
```

Now we want to grab our repo id for the upload step:

```
# hammer repository list

---|-------------|---------|--------------|----
ID | NAME        | PRODUCT | CONTENT TYPE | URL
---|-------------|---------|--------------|----
1  | source_rpms | Zoo     | yum          |
---|-------------|---------|--------------|----
```

Upload a source RPM into a repository

Now that we have our repository, we will upload a source RPM using the following command:

```
hammer repository upload-content --content-type srpm --id 1 --product-id 1 --path /root/garmindev-0.3.4-9.el7.src.rpm

Successfully uploaded file 'garmindev-0.3.4-9.el7.src.rpm'
```

Retrieving the repository info will show that the SRPM was uploaded. Note the content counts.

```
# hammer repository info --id 1

ID:                 1
Name:               source_rpms
Label:              source_rpms
Organization:       Default Organization
Red Hat Repository: no
Content Type:       yum
Mirror on Sync:     yes
URL:
Publish Via HTTP:   yes
Published At:       http://centos7-katello-nightly.area51.example.com/pulp/repos/Default_Organization/Library/custom/Zoo/source_rpms/
Relative Path:      Default_Organization/Library/custom/Zoo/source_rpms
Download Policy:    immediate
Product:
    ID:   1
    Name: Zoo
GPG Key:

Sync:
    Status: Not Synced
Created:            2019/08/21 15:37:14
Updated:            2019/08/21 15:37:16
Content Counts:
    Packages:       0
    Source RPMS:    1
    Package Groups: 0
    Errata:         0
    Module Streams: 0
```

Listing source RPMS

To list source rpms across the entire Katello server run this command:

```
# hammer srpm list

---|-----------|------------------------------
ID | NAME      | FILENAME
---|-----------|------------------------------
1  | garmindev | garmindev-0.3.4-9.el7.src.rpm
---|-----------|------------------------------
```

- Filter down by organization, product, repository, content-view and lifecycle environment by passing in the appropriate flags.

Getting information on a source RPM

To grab information on a source RPM run this command:

```
# hammer srpm info --id 1

ID:          1
Name:        garmindev
Version:     0.3.4
Architecture: src
Epoch:       0
Release:     9.el7
Filename:    garmindev-0.3.4-9.el7.src.rpm
Description:  Drivers for communication with Garmin GPS devices
```

Uploading SRPMs with the Repositories API

To upload source RPMs with the content_type parameter to the `import_uploads` API endpoint

To see all of the available options to use with the new Source RPM API

Katello 3.15 Documentation

3.15

Certificates

Checking for Validity

During installation any certificates for Katello are checked for validity. The same can be performed manually with `katello-certs-check` . Doing so can be useful when looking into SSL related issues or configuring custom certificates.

```
katello-certs-check -c ~/path/to/server.crt\
            -k ~/path/to/server.key\
            -b ~/path/to/cacert.crt
```

If you would like to configure Katello with a set of invalid certs, the validation check can be skipped by passing --certs-skip-check to the installer.

Custom Server Certificates

New Katello Installations

```
foreman-installer --scenario katello\
                  --certs-server-cert ~/path/to/server.crt\
                  --certs-server-key ~/path/to/server.key\
                  --certs-server-ca-cert ~/path/to/cacert.crt
```

--certs-server-ca-cert is the CA used for issuing the server certs. This CA gets distributed to content hosts and Smart Proxies.

For Smart Proxies the following options are passed to foreman-proxy-certs-generate :

```
foreman-proxy-certs-generate --foreman-proxy-fqdn "$FOREMAN_PROXY"\
                             --certs-tar ~/$FOREMAN_PROXY-certs.tar\
                             --server-cert ~/path/to/server.crt\
                             --server-key ~/path/to/server.key\
                             --server-ca-cert ~/cacert.crt
```

The rest of the procedure is identical to the default CA setup.

Existing Katello Installations

The first run of foreman-installer --scenario katello uses the default CA for both server and client certificates. To enforce the custom certificates to be deployed, one needs to set --certs-update-server to update the server certificate. --certs-update-server-ca should be given when updating the server CA in order for katello-ca-consumer-latest.noarch.rpm to be regenerated.

```
foreman-installer --scenario katello\
                  --certs-server-cert ~/path/to/server.crt\
                  --certs-server-key ~/path/to/server.key\
                  --certs-server-ca-cert ~/path/to/cacert.crt\
                  --certs-update-server --certs-update-server-ca
```

After the server CA changes the new version of the katello-ca-consumer RPM needs to be installed on content hosts:

```
rpm -Uvh http://katello.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

Any custom CA on the server needs to be used on the server certificates of any Smart Proxies as well. The certificates for Smart Proxies are generated by foreman-proxy-certs-generate .

```
foreman-proxy-certs-generate --foreman-proxy-fqdn "$FOREMAN_PROXY_CONTENT"\
                             --certs-tar ~/$FOREMAN_PROXY_CONTENT-certs.tar\
                             --server-cert ~/path/to/server.crt\
                             --server-key ~/path/to/server.key\
                             --server-ca-cert ~/cacert.crt\
                             --certs-update-server
```

After generation the utility will provide the necessary details on how to copy the new certificates to and run the installer on the Smart Proxy.

Updating Certificates

On the Katello server

To regenerate the server certificates when using the default CA or enforce deploying new certificates for the custom server CA the installer may be run in this way:

```
foreman-installer --scenario katello --certs-update-server
```

To regenerate all the certificates used in the Katello server use the --certs-update-all flag. This will generate and deploy the certificates as well as restart corresponding services.

On a Smart Proxy

For updating the certificates on a Smart Proxy pass the same options (--certs-update-server or --certs-update-all) to foreman-proxy-certs-generate . A tarball is generated containing the new certs and output will be shown indicating how to transfer it to the Smart Proxy and run the installer.

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15

HTTP Proxies

HTTP Proxy Support

Katello enables external HTTP proxies (provided by utilities such as squid) for repository operations such as synchronization.

HTTP proxies can be created and then assigned to a product though bulk selection as well as for each individual repository. Additionally, Katello provides HTTP proxy policies for products or repositories. Policies include:

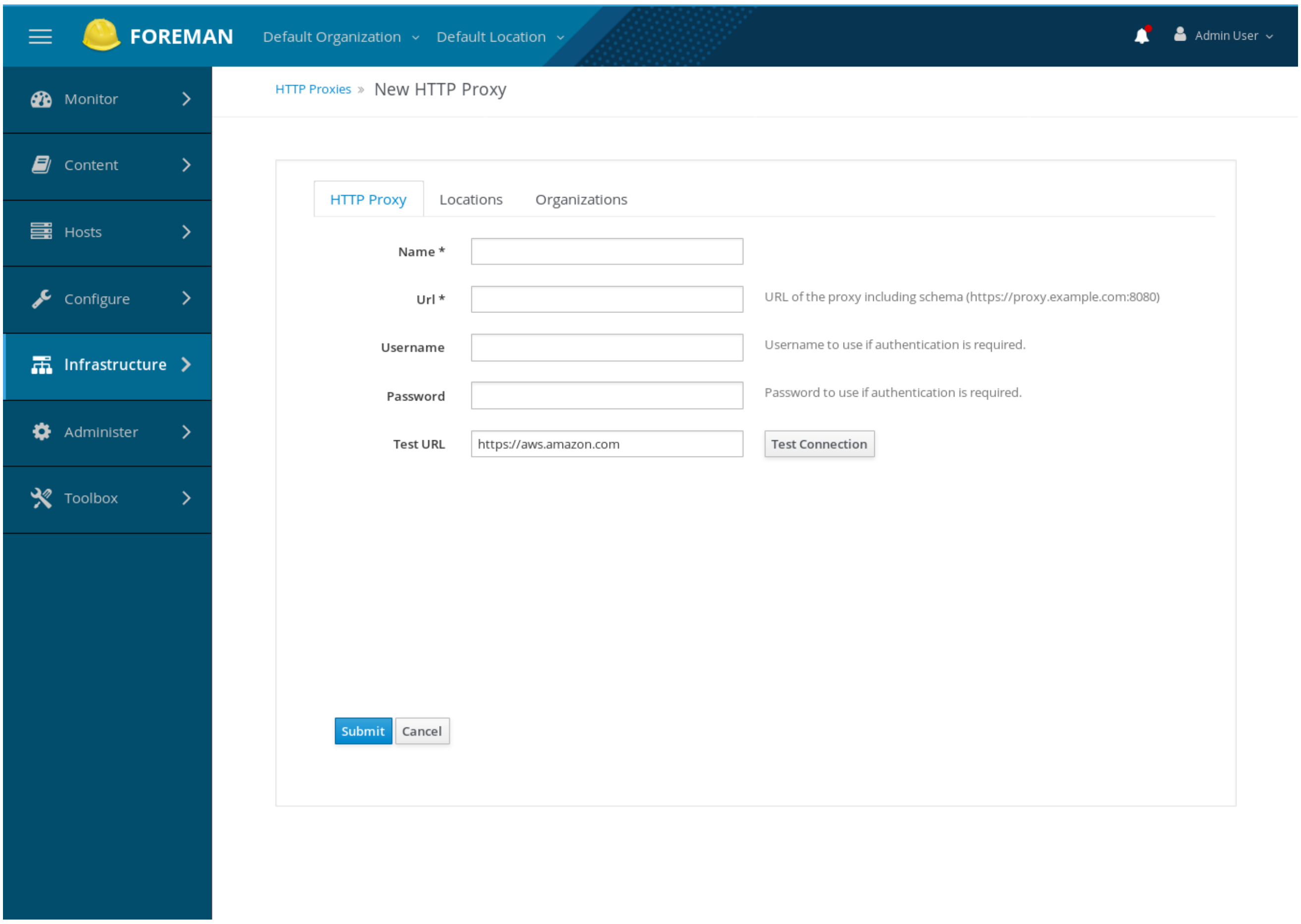
- Using the global HTTP proxy (the default)
- Using a specified HTTP proxy other than the global HTTP proxy

- Not using an HTTP proxy

Creation

Creating an HTTP Proxy through the web UI

- navigate to: Infrastructure > HTTP Proxies
- click **New HTTP Proxy**



Name *

Url * — URL of the proxy including schema (https://proxy.example.com:8080)

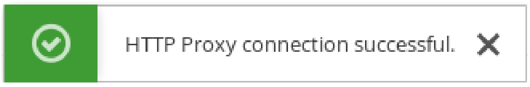
Username — Username to use if authentication is required.

Password — Password to use if authentication is required.

Test URL: https://aws.amazon.com

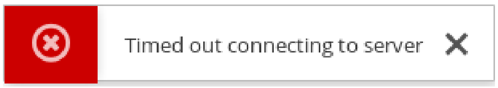
- *Name*: This required option is used to identify the HTTP proxy.
- *Url*: This required option is the URL of the proxy. Note that the scheme should be included. For example: “http://proxy.example.org:8888”
- *Username*: This option is used for proxy authentication, if required.
- *Password*: This option is used for proxy authentication, if required.

The provided field for **Test Connection** may be used to verify the proxy fields are set correctly. The field accepts a URL that a GET request will be sent to via the proxy configured in the form. If successful you will see a user notification such as:



HTTP Proxy connection successful.

If there is a problem with the proxy configuration, you will see an error notification similar to:



Timed out connecting to server

Removal

To remove a HTTP Proxy:

- navigate to: Infrastructure > HTTP Proxies
- click **Delete** in the row of the proxy you want to remove

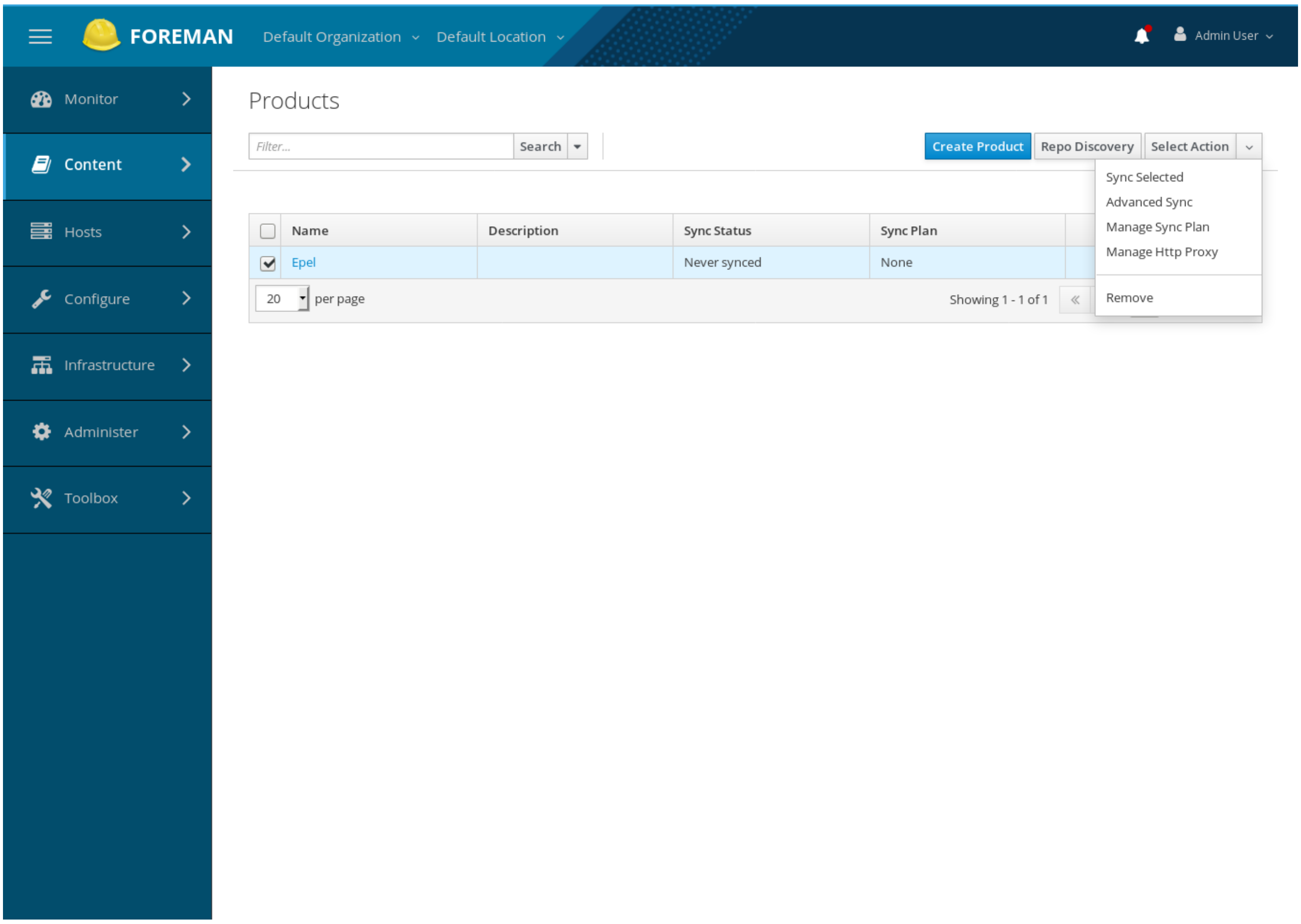
Bulk Applying HTTP Proxy Policies and HTTP Proxy Selection

HTTP proxy policies and HTTP proxy selection can be selected for one or more products. The policies and proxy selection propogates to all repositories contain in the products.

Selecting the Global Default Proxy

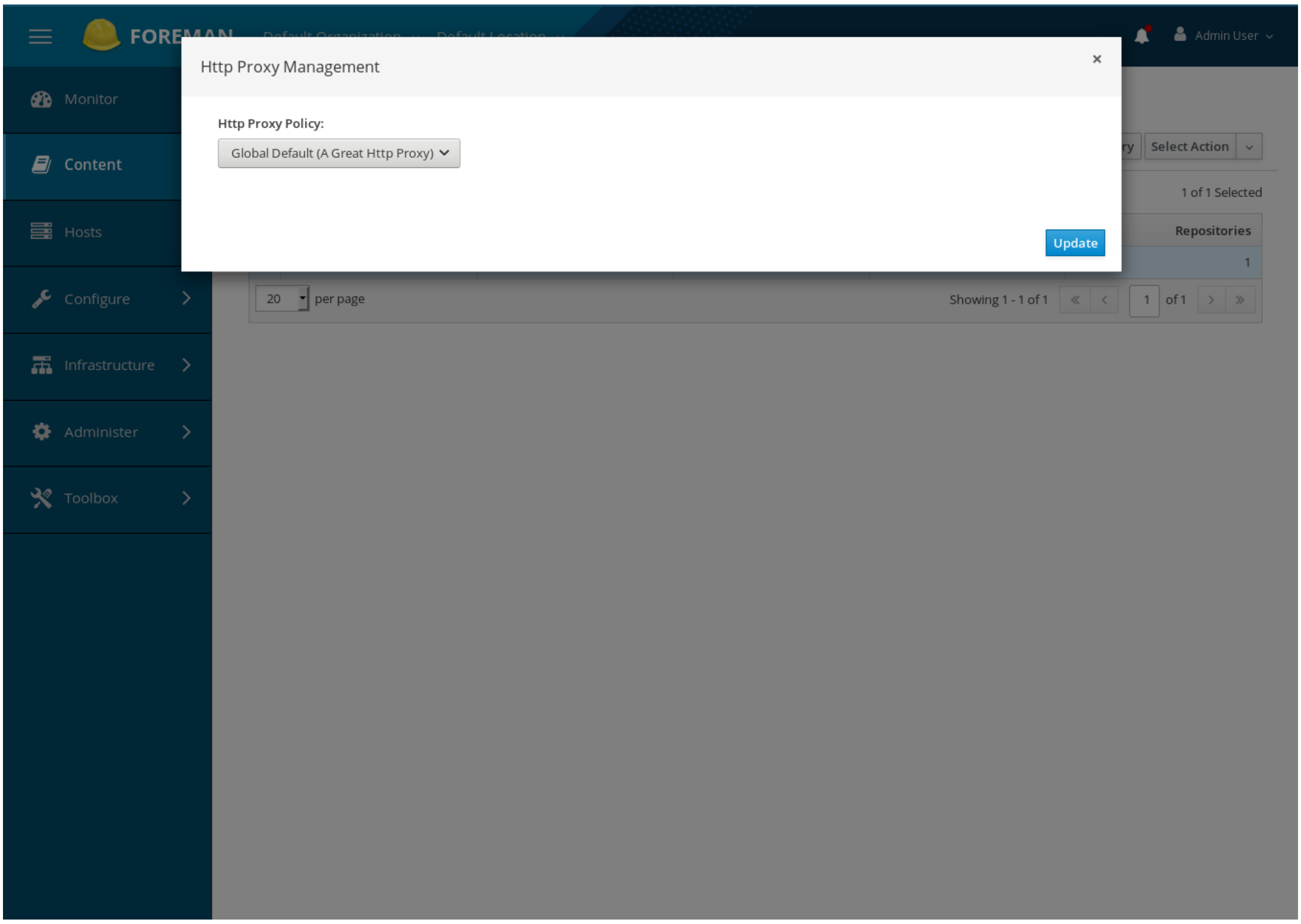
To apply the global default proxy policy to one or more products:

- navigate to: Content > Products
- Select the row checkbox for each product you want to assign the HTTP policy to
- Click the **Select Action** dropdown
- Select **Manage Http Proxy**
- Accept the default selection **Global Default (xxxx)**. Note that the text within the parenthesis is the name of the HTTP Proxy set as the global default.
- Click **Update**



☐	Name	Description	Sync Status	Sync Plan
☑	Epel		Never synced	None

20 per page   Showing 1 - 1 of 1



Http Proxy Management

Http Proxy Policy: Global Default (A Great Http Proxy)

Selecting the No HTTP Proxy policy

To choose a policy where no HTTP proxy is used for one or more products:

- navigate to: Content > Products
- Select one or more products by clicking the row checkbox for each product you want to assign the HTTP policy for
- Click the **Select Action** dropdown
- Select **Manage Http Proxy**
- Select the “No HTTP Proxy” selection.
- Click **Update**

Selecting a Specific HTTP Proxy

To choose a specific HTTP proxy for one or more products:

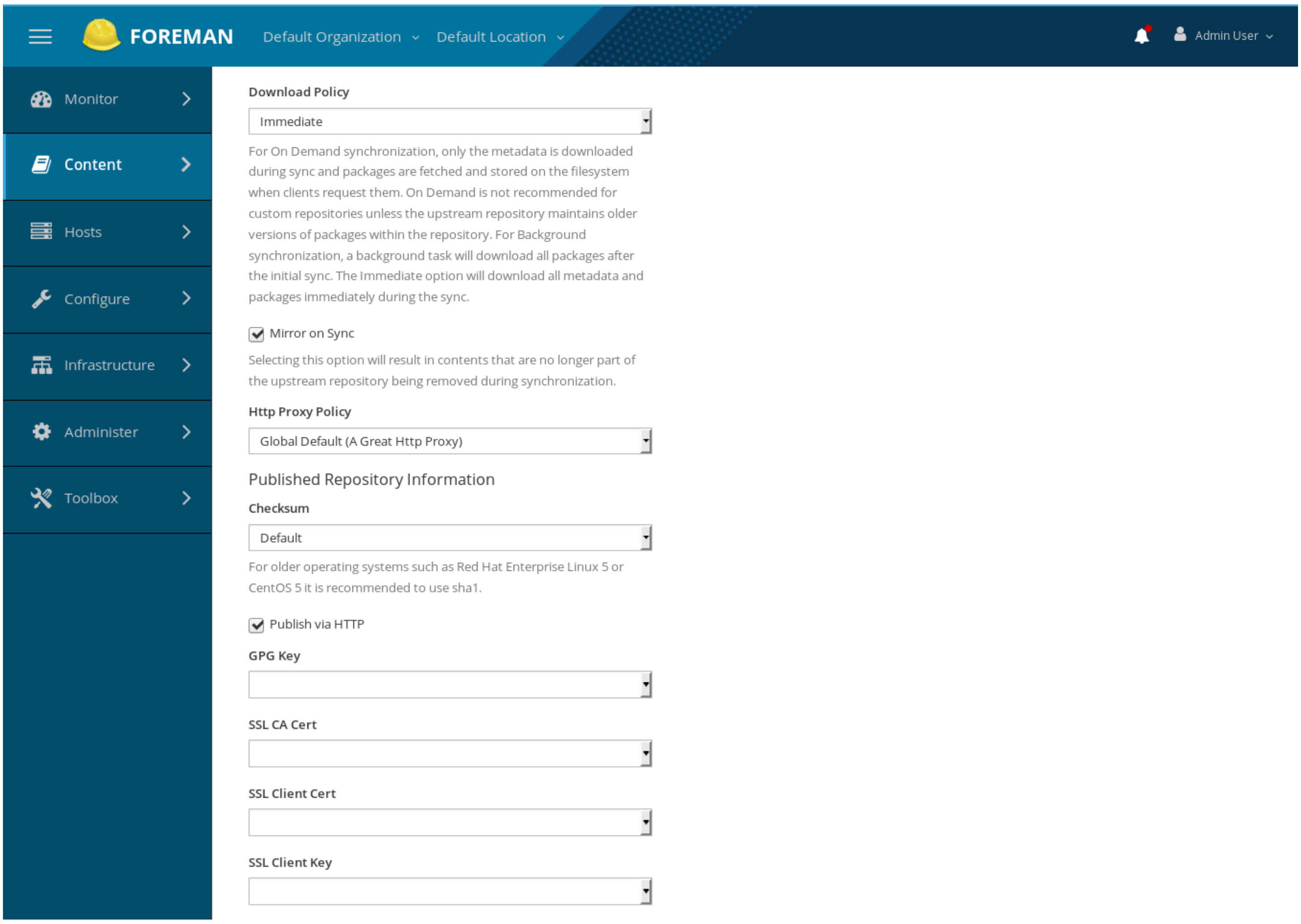
- navigate to: Content > Products
- Select one or more products by clicking the row checkbox for each product you want to assign the HTTP policy for
- Click the **Select Action** dropdown
- Select **Manage Http Proxy**
- Select the “Use specific HTTP proxy” policy selection.
- A new drop down will appear with a list of all curently defined HTTP proxies. Select the proxy you want.
- Click **Update**

Selecting a HTTP Proxy policy and HTTP Proxy for a new Repository

For more information about creating a new repository see Creating a Repository. There are two fields on the new repostory form for selecting the HTTP proxy policy and, if needed, a specific HTTP proxy.

Assigning the global default HTTP Proxy

By default, a new repository will be created with the “Global Default” policy. You can see this is the default selection in the **Http Proxy Policy** selection.



Download Policy

Immediate

For On Demand synchronization, only the metadata is downloaded during sync and packages are fetched and stored on the filesystem when clients request them. On Demand is not recommended for custom repositories unless the upstream repository maintains older versions of packages within the repository. For Background synchronization, a background task will download all packages after the initial sync. The Immediate option will download all metadata and packages immediately during the sync.

☑ Mirror on Sync

Selecting this option will result in contents that are no longer part of the upstream repository being removed during synchronization.

Http Proxy Policy

Global Default (A Great Http Proxy)

Published Repository Information

Checksum

Default

For older operating systems such as Red Hat Enterprise Linux 5 or CentOS 5 it is recommended to use sha1.

☑ Publish via HTTP

GPG Key

SSL CA Cert

SSL Client Cert

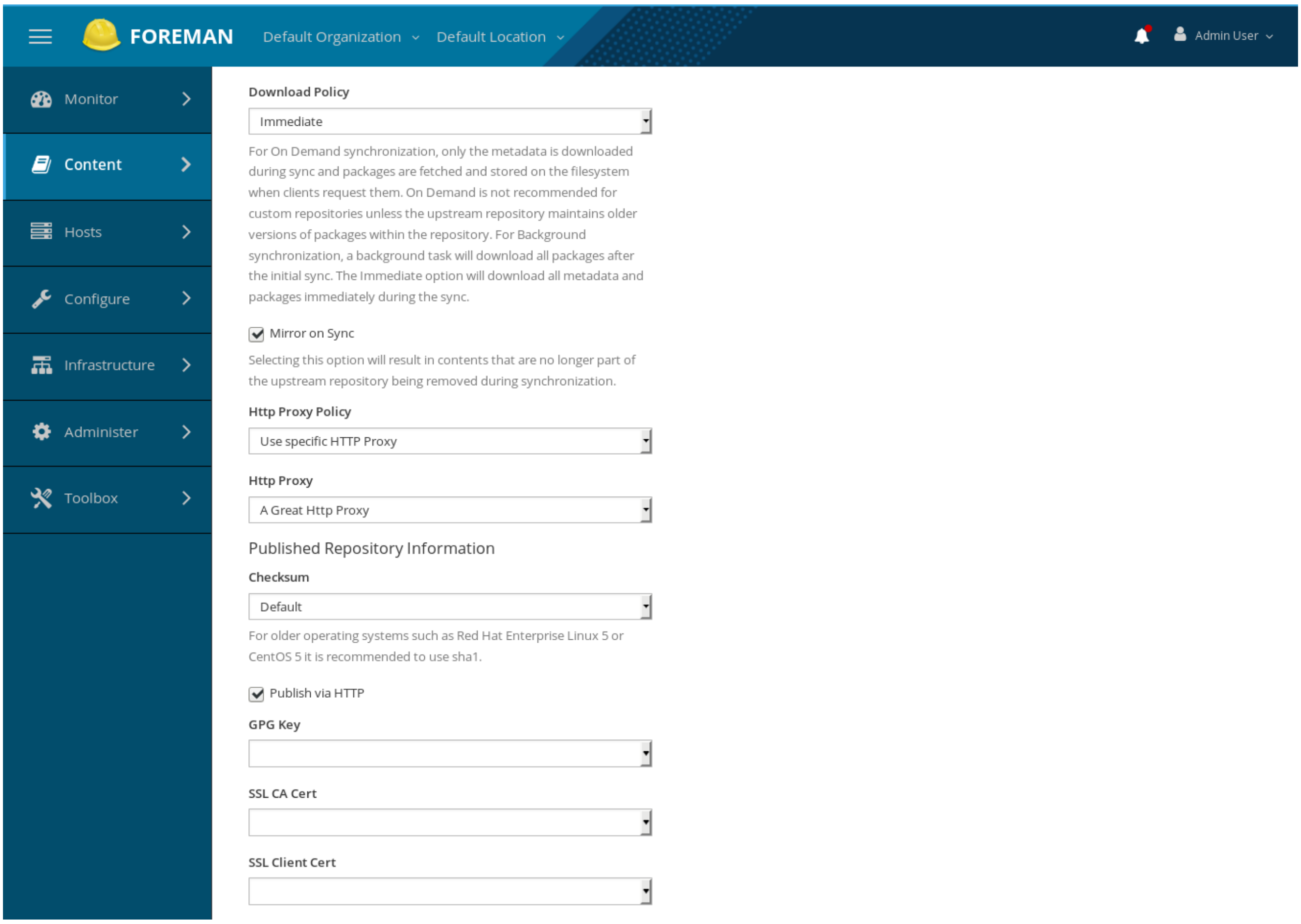
SSL Client Key

Assigning the No HTTP Proxy policy

If you don’t want any http proxy to be used, click the **Http Proxy Policy** menu and select “No HTTP Proxy”.

Assigning a specific HTTP Proxy

If you want to assign a specific HTTP proxy to be used for the new repository, click the **Http Proxy Policy**. A new menu will appear, presenting a list of all currently defined HTTP proxies. Select the wanted proxy and the new repository will use that proxy.



Download Policy

Immediate

For On Demand synchronization, only the metadata is downloaded during sync and packages are fetched and stored on the filesystem when clients request them. On Demand is not recommended for custom repositories unless the upstream repository maintains older versions of packages within the repository. For Background synchronization, a background task will download all packages after the initial sync. The Immediate option will download all metadata and packages immediately during the sync.

☑ Mirror on Sync

Selecting this option will result in contents that are no longer part of the upstream repository being removed during synchronization.

Http Proxy Policy

Use specific HTTP Proxy

Http Proxy

A Great Http Proxy

Published Repository Information

Checksum

Default

For older operating systems such as Red Hat Enterprise Linux 5 or CentOS 5 it is recommended to use sha1.

☑ Publish via HTTP

GPG Key

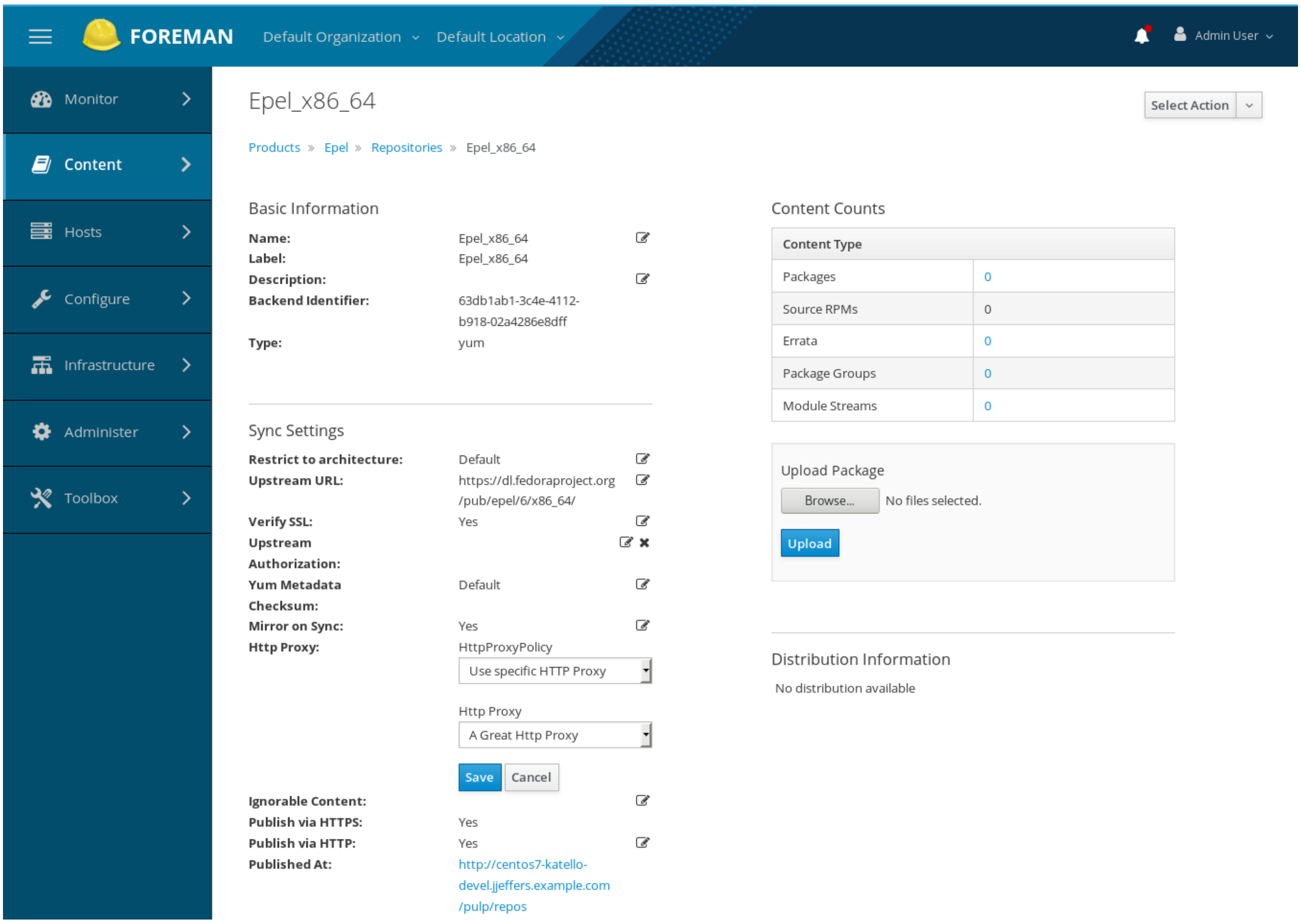
SSL CA Cert

SSL Client Cert

Assigning a HTTP Proxy Policy and HTTP Proxy for an existing Repository

To change the HTTP proxy policy or the HTTP proxy used by an existing repository:

- navigate to: Content -> Products -> (the repository you want to modify)
- under **Sync Settings** cick the form edit icon for the **Http Proxy** field. Note that the default display will show the current HTTP proxy policy and the name of the proxy the policy enables
- Menus for both the HTTP Proxy policy and, if you choose the “Use specific HTTP Proxy” policy, the HTTP Proxy appear
- To save any changes, click the **Save** button
- To ignore any changes, click the **Cancel** button



Epel_x86_64

Products » Epel » Repositories » Epel_x86_64

Basic Information

Name:	Epel_x86_64
Label:	Epel_x86_64
Description:	
Backend Identifier:	63db1ab1-3c4e-4112-b918-02a4286e8dff
Type:	yum

Sync Settings

Restrict to architecture:	Default
Upstream URL:	https://dl.fedoraproject.org/pub/epel/6/x86_64/
Verify SSL:	Yes
Upstream Authorization:	
Yum Metadata Checksum:	Default
Mirror on Sync:	Yes
Http Proxy:	HttpProxyPolicy Use specific HTTP Proxy Http Proxy A Great Http Proxy Save Cancel
Ignorable Content:	
Publish via HTTPS:	Yes
Publish via HTTP:	Yes
Published At:	http://centos7-katello-devel.jjeffers.example.com/pulp/repos

Content Counts

Content Type	
Packages	0
Source RPMs	0
Errata	0
Package Groups	0
Module Streams	0

Upload Package

Browse... No files selected.

Upload

Distribution Information

No distribution available

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

3.15

Katello Troubleshooting

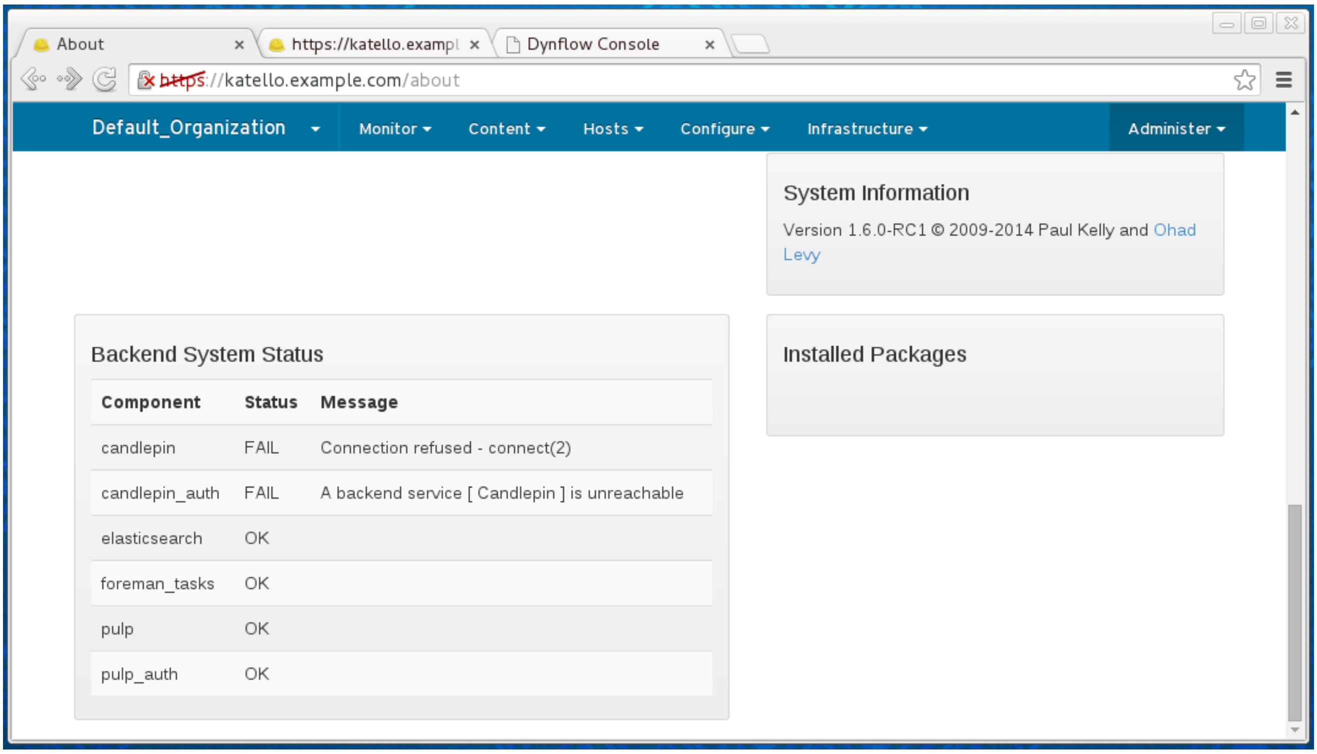
For general support information, see here.

Table of Contents

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:



System Information

Version 1.6.0-RC1 © 2009-2014 Paul Kelly and Ohad Levy

Backend System Status

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	

Installed Packages

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

`foreman-maintain` tool can be used to restart Katello related services. See `foreman-maintain --help` for more details.

Tasks

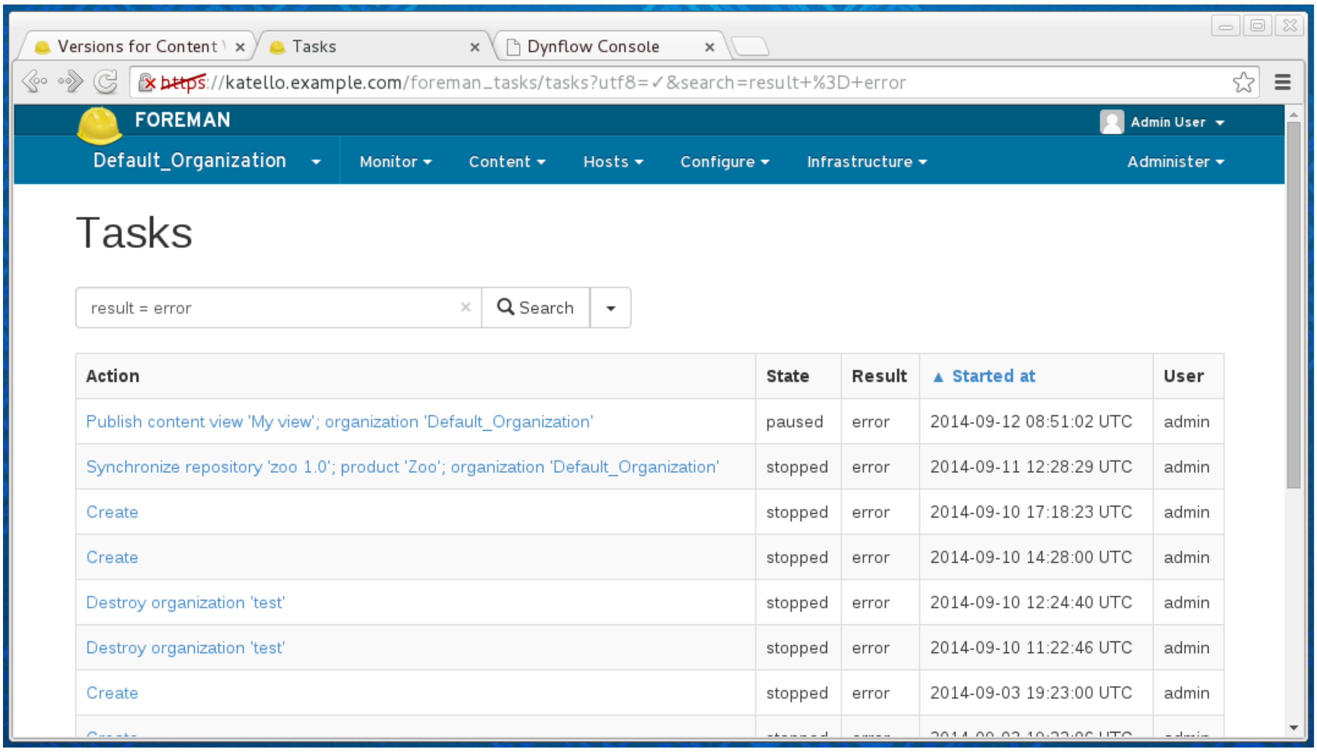
Katello uses Foreman Tasks for orchestration between the underlying services (local database, Pulp, Candlepin…). The tasks are modeled as Dynflow processes. When something goes wrong (and there might be many reasons for this happening), Dynflow gives us the tools to recover from these errors to get to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



https://katello.example.com/foreman_tasks/tasks?utf8=✓&search=result+%3D+error

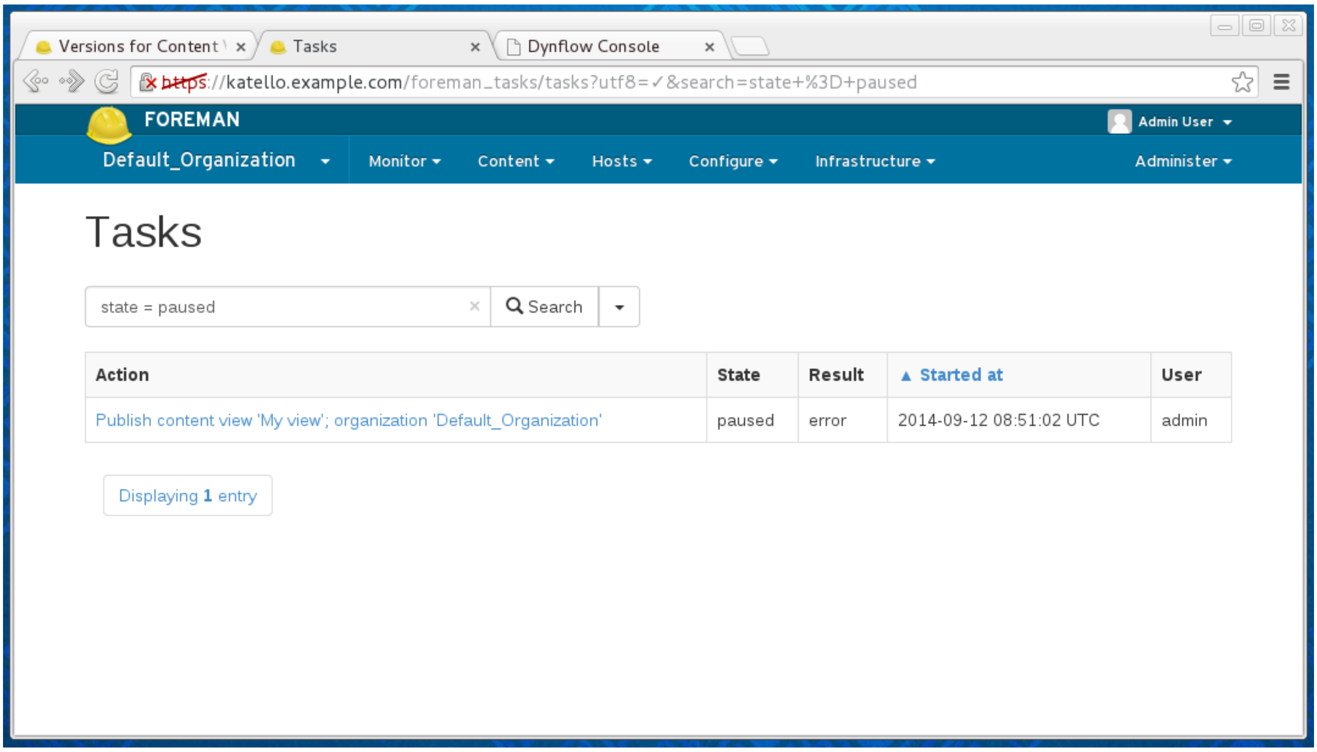
Tasks

result = error

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

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` :



https://katello.example.com/foreman_tasks/tasks?utf8=✓&search=state+%3D+paused

Tasks

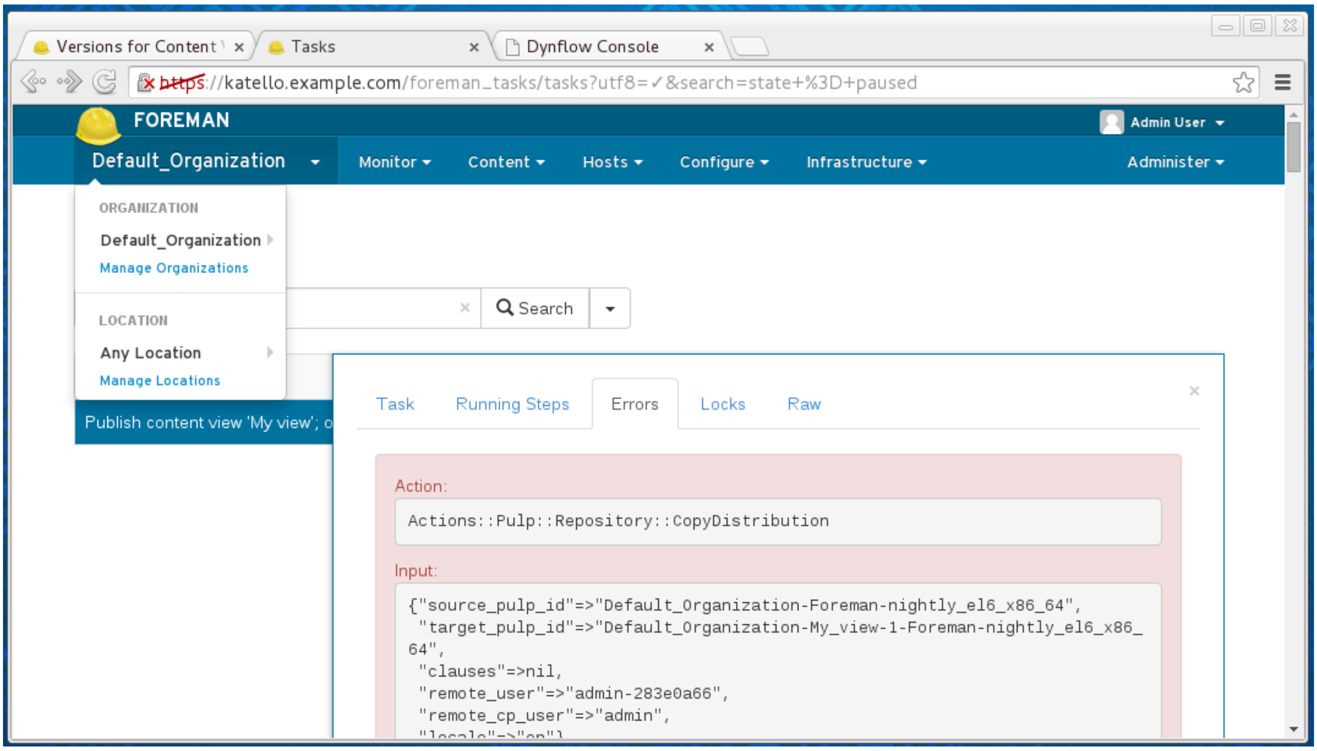
state = paused

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

Displaying **1** entry

Dealing with paused task

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



Action:

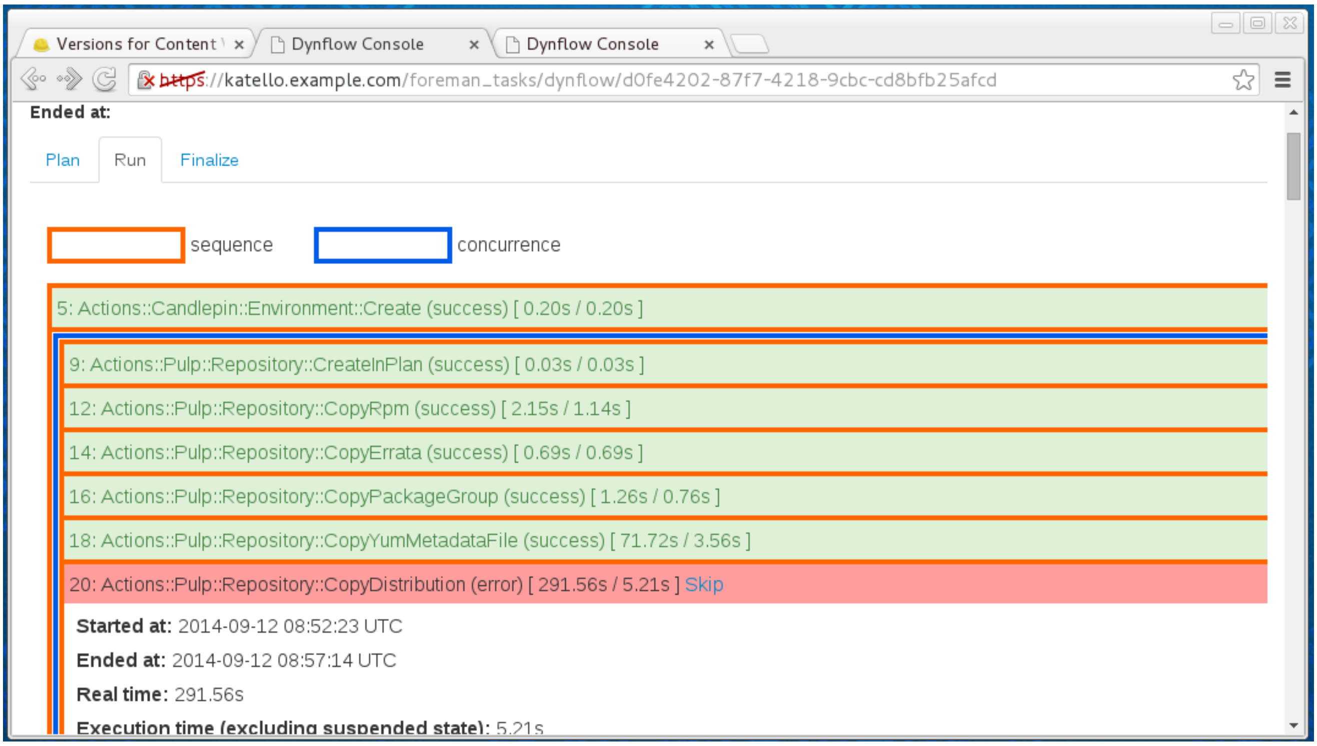
```
Actions::Pulp::Repository::CopyDistribution
```

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",
```

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):



Ended at:

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 (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

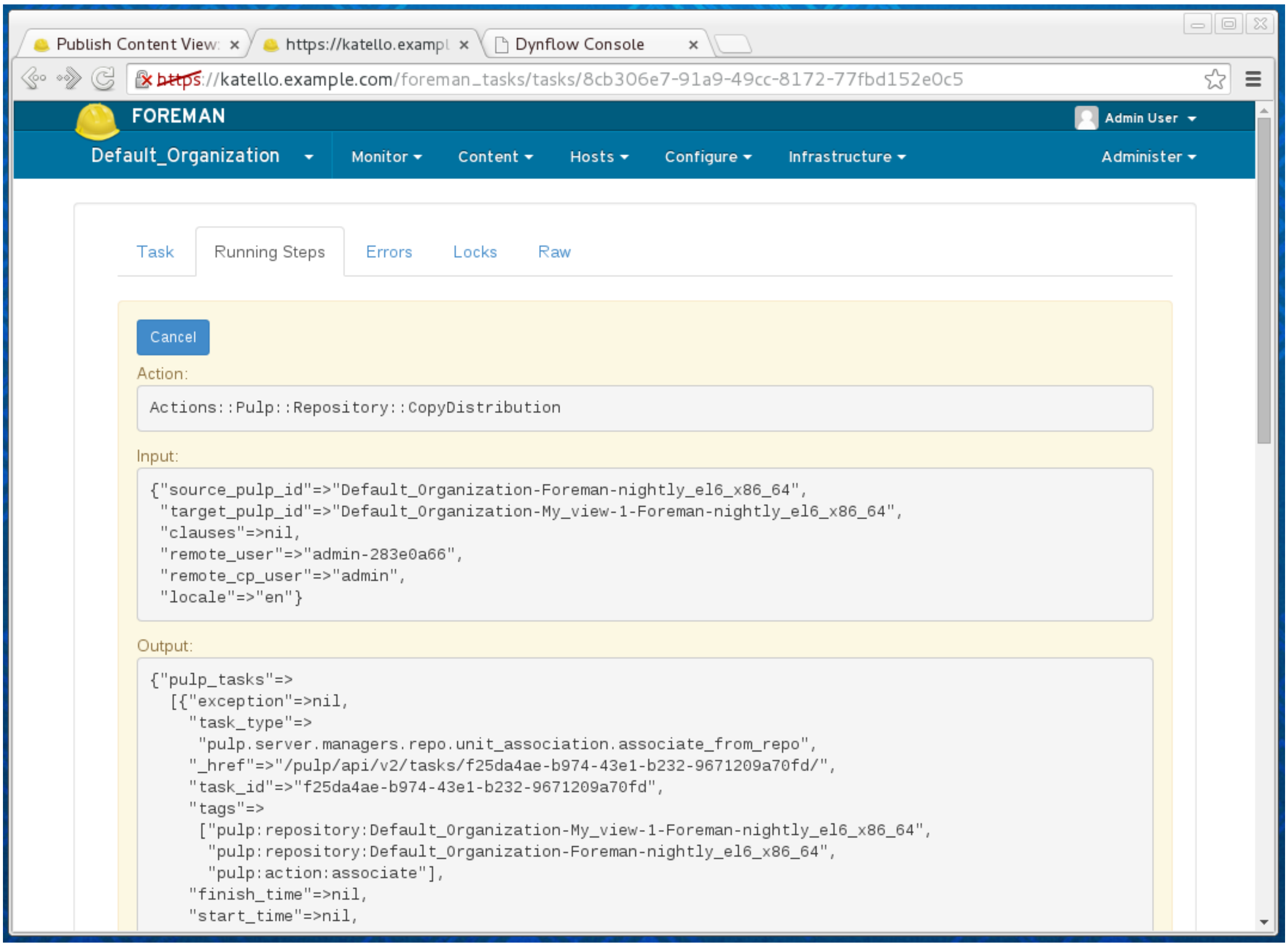
Caution: Dynflow console is considered a low-level tool and should be used very carefully, ideally discussing other options before using its features

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 came 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 withing 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:

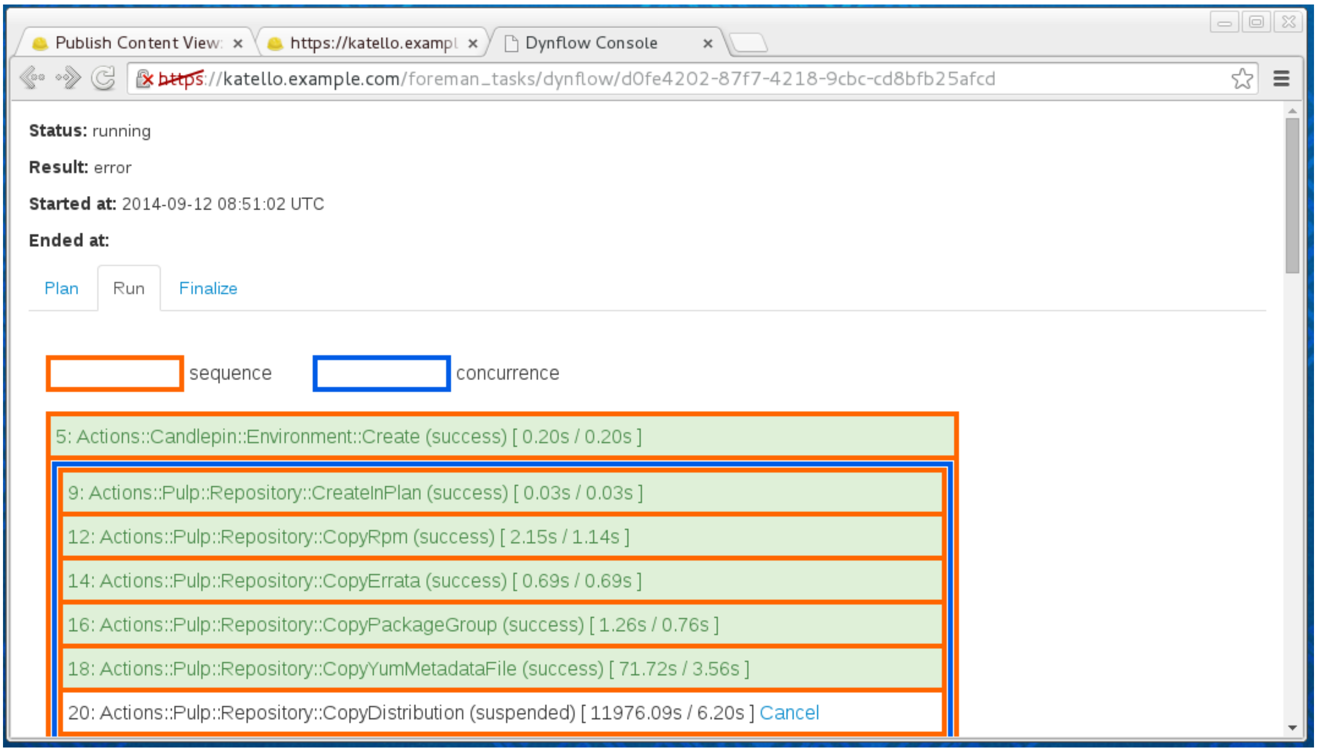
```
{"source_pulp_id"=>"Default_Organization-Foreman-nightly_el6_x86_64",
 "target_pulp_id"=>"Default_Organization-My_view-1-Foreman-nightly_el6_x86_64",
 "clauses"=>nil,
 "remote_user"=>"admin-283e0a66",
 "remote_cp_user"=>"admin",
 "locale"=>"en"}
```

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:

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]

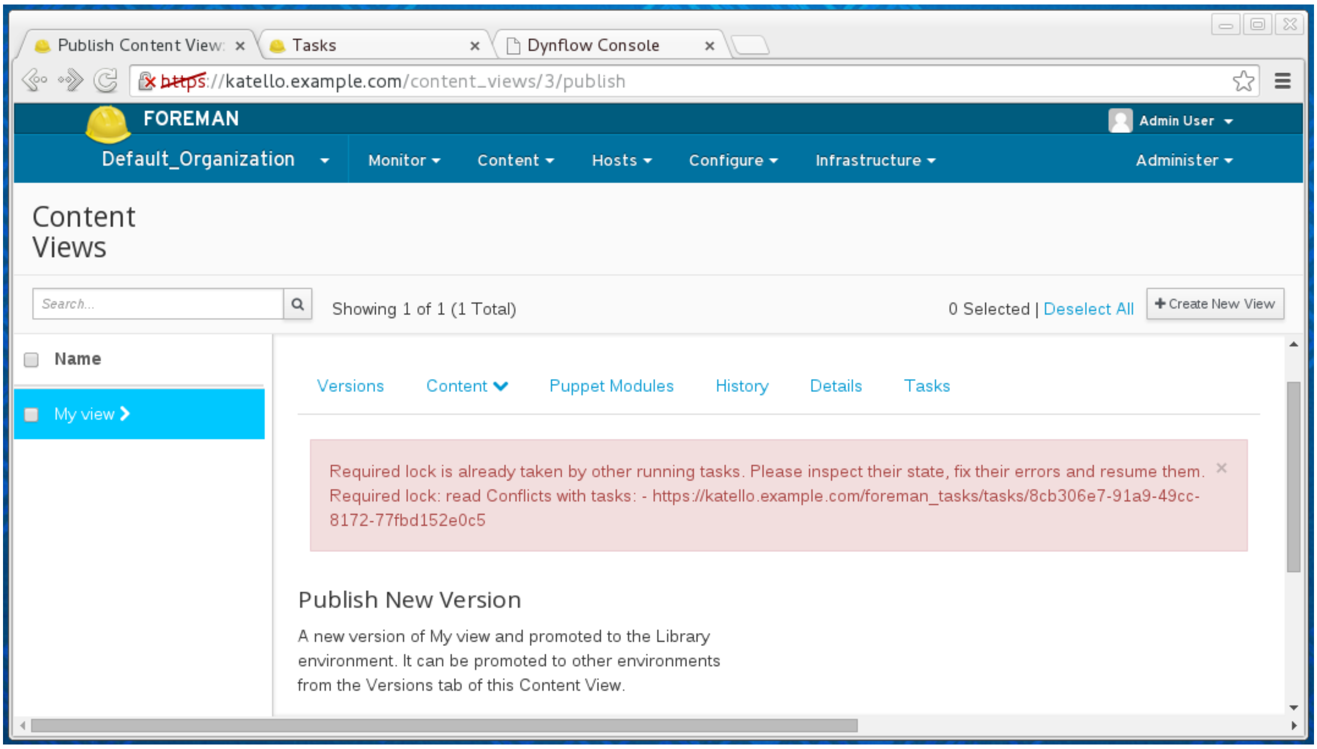
20: Actions::Pulp::Repository::CopyDistribution (suspended) [11976.09s / 6.20s] Cancel

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 following 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.` :



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-8172-77fbd152e0c5

Publish New Version

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).

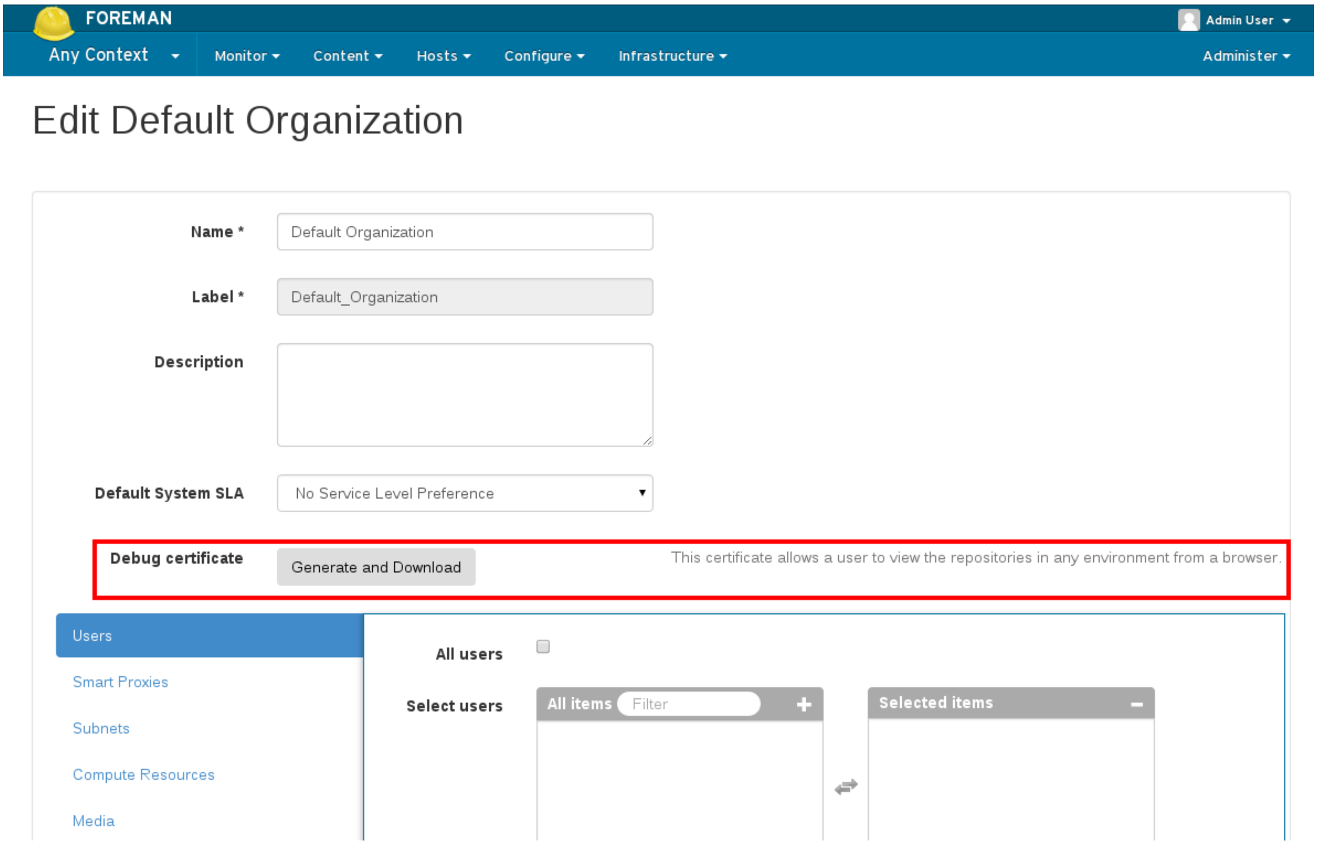
In rare cases, it might be hard to get into the stopped state. There is a possibility to unlock the resource in the `running` / `paused` task. This will switch the task into `stopped` state, freeing the resources for other tasks. **Caution: unlocking allows running other tasks to run on potentially inconsistent 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 * Default Organization

Label * Default_Organization

Description

Default System SLA No Service Level Preference

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

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/repomd.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/repomd.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 authoriity 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=<Non
e>,
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.

Katello 3.15 Documentation

3.15 ▾

Troubleshooting Content Hosts

Registration Issues

Starting with Katello 3.12, content hosts are required to have a unique UUID reported by subscription-manager when registering. This UUID comes from the BIOS or virtualization layer and is unique in most cases. If a registering host reports a UUID that matches an existing registration an error will be received on the client:

```
The DMI UUID of this host (ABF18D4C-2708-11B2-A85C-AA9890D228CC) matches other registered hosts: host-a.example.com, ho
st-b.example.com
```

Action should be taken in order to ensure that the reported UUID is unique. However, Katello provides two mechanisms to provide a temporary solution so that the content host can be registered and receive updates while the underlying cause is resolved.

Client-side DMI UUID override

The DMI UUID can be overridden on the content host itself by setting a custom fact value which will be uploaded by subscription-manager.

First, generate a new UUID:

```
[root@myhost ~]$ uuidgen
eae7fe44-8256-4072-813f-0e1d691f093c
```

Take the generated UUID and place it into a JSON structure and save it to a file where subscription-manager can find it:

```
echo '{"dmi.system.uuid": "eae7fe44-8256-4072-813f-0e1d691f093c"}' > /etc/rhsm/facts/uuid_override.facts
```

subscription-manager will now report this value to the Katello server

```
subscription-manager facts | grep dmi.system.uuid
dmi.system.uuid: eae7fe44-8256-4072-813f-0e1d691f093c
```

Registration should be successful at this point.

Server-side BIOS UUID override

If overriding the BIOS UUID from the client side (preferred) is not an option, Katello can be configured to automatically generate a suitable UUID at registration-time. In order to do so, the problematic UUID must be noted from the registration error received by the client. It can also be found in the server production.log

With the DMI UUID handy, navigate to Administer -> Settings -> Content in the UI. Find the setting named ‘Host Duplicate DMI UUIDs’ and edit its value. The value must be in array format, ex: [‘ABF18D4C-2708-11B2-A85C-AA9890D228CC’] Multiple values can be separated by commas.

After saving, any content hosts with the UUID(s) applied in the setting will have a new DMI UUID created when registering that will persist when the host checks in. However, this value exists on the server only and is not pushed to the client.

Foreman v2

Foreman API v2 is currently the default API version.

Resources

Activation keys

Resource	Description
GET /katello/api/activation_keys	List activation keys
GET /katello/api/environments/:environment_id/activation_keys	
GET /katello/api/organizations/:organization_id/activation_keys	
POST /katello/api/activation_keys	Create an activation key
PUT /katello/api/activation_keys/:id	Update an activation key
DELETE /katello/api/activation_keys/:id	Destroy an activation key
GET /katello/api/activation_keys/:id	Show an activation key
POST /katello/api/activation_keys/:id/copy	Copy an activation key
GET /katello/api/activation_keys/:id/host_collections/available	List host collections the activation key does not belong to
GET /katello/api/activation_keys/:id/releases	Show release versions available for an activation key
GET /katello/api/activation_keys/:id/product_content	Show content available for an activation key
POST /katello/api/activation_keys/:id/host_collections	
PUT /katello/api/activation_keys/:id/host_collections	
PUT /katello/api/activation_keys/:id/add_subscriptions	Attach a subscription
PUT /katello/api/activation_keys/:id/remove_subscriptions	Unattach a subscription
PUT /katello/api/activation_keys/:id/content_override	Override content for activation_key

Ansible Collections

Resource	Description
GET /katello/api/ansible_collections	List ansible_collections

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

Thanks to our sponsors for hosting this website and our project services.

Katello 3.15 Documentation

Repository Sync (default settings)

Request # 1:

POST /pulp/api/v2/repositories/scenario_test/actions/sync/

Backend Service: pulp

Description:

Request body

```
{
  "override_config": {
    "num_threads": 4,
    "validate": true
  }
}
```

Response body

```
{
  "spawned_tasks": [
    {
      "_href": "/pulp/api/v2/tasks/8b9c7591-4d1b-48b7-b37b-bce4fdc9c2fd/"
,
      "task_id": "8b9c7591-4d1b-48b7-b37b-bce4fdc9c2fd"
    }
  ],
  "result": null,
  "error": null
}
```

Request # 2: Poll Sync task

GET /pulp/api/v2/tasks/8b9c7591-4d1b-48b7-b37b-bce4fdc9c2fd/

Backend Service: pulp

Description:

Total Requests for this URL: 32

Request body

None

Response body

```
{
  "exception": null,
  "task_type": "pulp.server.managers.repo.sync.sync",
  "_href": "/pulp/api/v2/tasks/8b9c7591-4d1b-48b7-b37b-bce4fdc9c2fd/",
  "task_id": "8b9c7591-4d1b-48b7-b37b-bce4fdc9c2fd",
  "tags": [
    "pulp:repository:scenario_test",
    "pulp:action:sync"
  ],
  "finish_time": "2017-03-30T21:16:05Z",
  "_ns": "task_status",
  "start_time": "2017-03-30T21:16:05Z",
  "traceback": null,
  "spawned_tasks": [
    {
      "_href": "/pulp/api/v2/tasks/0837f608-4696-449a-811f-70ddabe59025/"
,
      "task_id": "0837f608-4696-449a-811f-70ddabe59025"
    }
  ],
  "progress_report": {
    "yum_importer": {
      "content": {
        "items_total": 0,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": {
          "rpm_total": 0,
          "rpm_done": 0,
          "drpm_total": 0,
          "drpm_done": 0
        },
        "size_total": 0,
        "size_left": 0,
        "items_left": 0
      },
      "comps": {
        "state": "FINISHED"
      },
      "purge_duplicates": {
        "state": "FINISHED"
      },
      "distribution": {
        "items_total": 3,
        "state": "FINISHED",
        "error_details": [

        ],
        "items_left": 0
      },
      "errata": {
        "state": "FINISHED"
      },
      "metadata": {
        "state": "FINISHED"
      }
    }
  },
  "queue": "reserved_resource_worker-1@dev.example.com.dq",
  "state": "finished",
  "worker_name": "reserved_resource_worker-1@dev.example.com",
  "result": {
    "result": "success",
    "importer_id": "yum_importer",
    "exception": null,
    "repo_id": "scenario_test",
    "traceback": null,
    "started": "2017-03-30T21:16:05Z",
    "_ns": "repo_sync_results",
    "completed": "2017-03-30T21:16:05Z",
    "importer_type_id": "yum_importer",
    "error_message": null,
    "summary": {
      "content": {
        "state": "FINISHED"
      },
      "comps": {
        "state": "FINISHED"
      },
      "purge_duplicates": {
        "state": "FINISHED"
      },
      "distribution": {
        "state": "FINISHED"
      },
      "errata": {
        "state": "FINISHED"
      },
      "metadata": {
        "state": "FINISHED"
      }
    },
    "added_count": 15,
    "removed_count": 0,
    "updated_count": 0,
    "id": "58dd7595418a8a0648b9bc52",
    "details": {
      "content": {
        "size_total": 0,
        "items_left": 0,
        "items_total": 0,
        "state": "FINISHED",
        "size_left": 0,
        "details": {
          "rpm_total": 0,
          "rpm_done": 0,
          "drpm_total": 0,
          "drpm_done": 0
        },
        "error_details": [

        ]
      },
      "comps": {
        "state": "FINISHED"
      },
      "purge_duplicates": {
        "state": "FINISHED"
      },
      "distribution": {
        "items_total": 3,
        "state": "FINISHED",
        "error_details": [

        ],
        "items_left": 0
      },
      "errata": {
        "state": "FINISHED"
      },
      "metadata": {
        "state": "FINISHED"
      }
    }
  },
  "error": null,
  "_id": {
    "$oid": "58dd7594e6919db96421ea13"
  },
  "id": "58dd7594e6919db96421ea13"
}
```

Request # 3: Poll Publish task

GET /pulp/api/v2/tasks/0837f608-4696-449a-811f-70ddabe59025/

Backend Service: pulp

Description:

Total Requests for this URL: 12

Request body

None

Response body

```
{
  "exception": null,
  "task_type": "pulp.server.managers.repo.publish.publish",
  "_href": "/pulp/api/v2/tasks/0837f608-4696-449a-811f-70ddabe59025/",
  "task_id": "0837f608-4696-449a-811f-70ddabe59025",
  "tags": [
    "pulp:repository:scenario_test",
    "pulp:action:publish"
  ],
  "finish_time": "2017-03-30T21:16:05Z",
  "_ns": "task_status",
  "start_time": "2017-03-30T21:16:05Z",
  "traceback": null,
  "spawned_tasks": [

  ],
  "progress_report": {
    "scenario_test": [
      {
        "num_success": 1,
        "description": "Copying files",
        "step_type": "save_tar",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "8fa5d08e-3473-4127-80f6-223bee0a0409",
        "num_processed": 1
      },
      {
        "num_success": 1,
        "description": "Initializing repo metadata",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "32cee4d1-6e51-419a-9833-5f2c88e3efd9",
        "num_processed": 1
      },
      {
        "num_success": 1,
        "description": "Publishing Distribution files",
        "step_type": "distribution",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "d62a1815-acd5-439c-b511-dcbd69edade4",
        "num_processed": 1
      },
      {
        "num_success": 8,
        "description": "Publishing RPMs",
        "step_type": "rpms",
        "items_total": 8,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "197b0894-07fd-470b-8bb4-5a55b9713d18",
        "num_processed": 8
    },
    {
      "num_success": 0,
      "description": "Publishing Delta RPMs",
      "step_type": "drpms",
      "items_total": 1,
      "state": "SKIPPED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "54795382-de70-4de3-a8ce-5f726b8f9cb2",
      "num_processed": 0
    },
    {
      "num_success": 3,
      "description": "Publishing Errata",
      "step_type": "errata",
      "items_total": 3,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "1996449c-9471-4988-b3ee-b7ceae0c6b41",
      "num_processed": 3
    },
    {
      "num_success": 3,
      "description": "Publishing Comps file",
      "step_type": "comps",
      "items_total": 3,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "f12472f5-4012-4112-afe0-49f57348e569",
      "num_processed": 3
    },
    {
      "num_success": 0,
      "description": "Publishing Metadata.",
      "step_type": "metadata",
      "items_total": 0,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "31efb1ef-a413-48be-99f2-7f4788a1cebd",
      "num_processed": 0
    },
    {
      "num_success": 1,
      "description": "Closing repo metadata",
      "step_type": "close_repo_metadata",
      "items_total": 1,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "22b9313d-ffae-49e3-8027-c86ef68f34de",
      "num_processed": 1
    },
    {
      "num_success": 0,
      "description": "Generating sqlite files",
      "step_type": "generate sqlite",
      "items_total": 1,
      "state": "SKIPPED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "fbc75485-e400-4cfa-bfa7-693edd8a832c",
      "num_processed": 0
    },
    {
      "num_success": 0,
      "description": "Generating HTML files",
      "step_type": "repoview",
      "items_total": 1,
      "state": "SKIPPED",
      "error_details": [
        ],
        "details": "",
        "num_failures": 0,
        "step_id": "bf1215fe-c617-4481-894a-1ff1c7c7043b",
        "num_processed": 0
      },
      {
        "num_success": 1,
        "description": "Publishing files to web",
        "step_type": "publish_directory",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "605ae2e4-c443-4a5b-ba9f-f313d23c4822",
        "num_processed": 1
      },
      {
        "num_success": 1,
        "description": "Writing Listings File",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "9fa215e9-b465-4f33-9ddd-e05c5a9e314d",
        "num_processed": 1
      }
    ]
  },
  "queue": "reserved_resource_worker-1@dev.example.com.dq",
  "state": "finished",
  "worker_name": "reserved_resource_worker-1@dev.example.com",
  "result": {
    "result": "success",
    "exception": null,
    "repo_id": "scenario_test",
    "started": "2017-03-30T21:16:05Z",
    "_ns": "repo_publish_results",
    "completed": "2017-03-30T21:16:05Z",
    "traceback": null,
    "distributor_type_id": "yum_distributor",
    "summary": {
      "generate sqlite": "SKIPPED",
      "initialize_repo_metadata": "FINISHED",
      "rpms": "FINISHED",
      "repoview": "SKIPPED",
      "close_repo_metadata": "FINISHED",
      "drpms": "SKIPPED",
      "comps": "FINISHED",
      "distribution": "FINISHED",
      "save_tar": "FINISHED",
      "publish_directory": "FINISHED",
      "errata": "FINISHED",
      "metadata": "FINISHED"
    },
    "error_message": null,
    "distributor_id": "scenario_test",
    "id": "58dd7595418a8a0648b9bc53",
    "details": [
      {
        "num_success": 1,
        "description": "Copying files",
        "step_type": "save_tar",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "8fa5d08e-3473-4127-80f6-223bee0a0409",
        "num_processed": 1
      },
      {
        "num_success": 1,
        "description": "Initializing repo metadata",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "32cee4d1-6e51-419a-9833-5f2c88e3efd9",
        "num_processed": 1
      },
    },
    {
      "num_success": 1,
      "description": "Publishing Distribution files",
      "step_type": "distribution",
      "items_total": 1,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "d62a1815-acd5-439c-b511-dcbd69edade4",
      "num_processed": 1
    },
    {
      "num_success": 8,
      "description": "Publishing RPMs",
      "step_type": "rpms",
      "items_total": 8,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "197b0894-07fd-470b-8bb4-5a55b9713d18",
      "num_processed": 8
    },
    {
      "num_success": 0,
      "description": "Publishing Delta RPMs",
      "step_type": "drpms",
      "items_total": 1,
      "state": "SKIPPED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "54795382-de70-4de3-a8ce-5f726b8f9cb2",
      "num_processed": 0
    },
    {
      "num_success": 3,
      "description": "Publishing Errata",
      "step_type": "errata",
      "items_total": 3,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "1996449c-9471-4988-b3ee-b7ceae0c6b41",
      "num_processed": 3
    },
    {
      "num_success": 3,
      "description": "Publishing Comps file",
      "step_type": "comps",
      "items_total": 3,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "f12472f5-4012-4112-afe0-49f57348e569",
      "num_processed": 3
    },
    {
      "num_success": 0,
      "description": "Publishing Metadata.",
      "step_type": "metadata",
      "items_total": 0,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "31efb1ef-a413-48be-99f2-7f4788a1cebd",
      "num_processed": 0
    },
    {
      "num_success": 1,
      "description": "Closing repo metadata",
      "step_type": "close_repo_metadata",
      "items_total": 1,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
        "details": "",
        "num_failures": 0,
        "step_id": "22b9313d-ffae-49e3-8027-c86ef68f34de",
        "num_processed": 1
      },
      {
        "num_success": 0,
        "description": "Generating sqlite files",
        "step_type": "generate sqlite",
        "items_total": 1,
        "state": "SKIPPED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "fbc75485-e400-4cfa-bfa7-693edd8a832c",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Generating HTML files",
        "step_type": "repoview",
        "items_total": 1,
        "state": "SKIPPED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "bf1215fe-c617-4481-894a-1ff1c7c7043b",
        "num_processed": 0
      },
      {
        "num_success": 1,
        "description": "Publishing files to web",
        "step_type": "publish_directory",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "605ae2e4-c443-4a5b-ba9f-f313d23c4822",
        "num_processed": 1
      },
      {
        "num_success": 1,
        "description": "Writing Listings File",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "9fa215e9-b465-4f33-9ddd-e05c5a9e314d",
        "num_processed": 1
      }
    ]
  },
  "error": null,
  "_id": {
    "$oid": "58dd7595e6919db96421ea23"
  },
  "id": "58dd7595e6919db96421ea23"
}
```

Request # 4: Distribution Search

POST /pulp/api/v2/repositories/scenario_test/search/units/

Backend Service: pulp

Description: Search for distribution information with all fields

Request body

```
{
  "criteria": {
    "type_ids": [
      "distribution"
    ]
  }
}
```

Response body

```
[
  {
    "metadata": {
      "files": [
        {
          "relativepath": "images/test2.img",
          "checksumtype": "sha256",
          "checksum": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
        },
        {
          "relativepath": "empty.iso",
          "checksumtype": "sha256",
          "checksum": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
        },
        {
          "relativepath": "images/test1.img",
          "checksumtype": "sha256",
          "checksum": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
        }
      ],
      "_storage_path": "/var/lib/pulp/content/units/distribution/9b/831256a124718bf39166b564d8e689954ff0a8f0f479ba24cfa26350109b
c5",
      "family": "Test Family",
      "downloaded": true,
      "timestamp": 1323112153.09,
      "_last_updated": 1490908565,
      "_content_type_id": "distribution",
      "variant": "TestVariant",
      "id": "ks-Test Family-TestVariant-16-x86_64",
      "version": "16",
      "version_sort_index": "02-16",
      "pulp_user_metadata": {
      },
      "packagedir": "",
      "_id": "be1ea9b9-518c-4c05-ab97-3d6102482158",
      "arch": "x86_64",
      "_ns": "units_distribution"
    },
    "updated": "2017-03-30T21:16:05Z",
    "repo_id": "scenario_test",
    "created": "2017-03-30T21:16:05Z",
    "unit_type_id": "distribution",
    "unit_id": "be1ea9b9-518c-4c05-ab97-3d6102482158",
    "_id": {
      "$oid": "58dd7595e6919db96421ea1c"
    }
  }
]
```

Request # 5: Fetch rpm unit ids for this repo

POST /pulp/api/v2/repositories/scenario_test/search/units/

Backend Service: pulp

Description:

Request body

```
{
  "criteria": {
    "type_ids": [
      "rpm"
    ],
    "fields": {
      "unit": [

      ],
      "association":
[
        "unit_id"
      ]
    }
  }
}
```

Response body

```
[
  {
    "metadata": {
      "_id": "085babbb-3a43-4b9c-bdfb-915fd78c7bec",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea1a"
    },
    "unit_id": "085babbb-3a43-4b9c-bdfb-915fd78c7bec",
    "unit_type_id": "rpm"
  },
  {
    "metadata": {
      "_id": "2b00d383-f2c7-462c-bfe3-25a008caa0fe",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea18"
    },
    "unit_id": "2b00d383-f2c7-462c-bfe3-25a008caa0fe",
    "unit_type_id": "rpm"
  },
  {
    "metadata": {
      "_id": "5ee1ec3a-7f81-47e6-bbba-11b443251f29",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea15"
    },
    "unit_id": "5ee1ec3a-7f81-47e6-bbba-11b443251f29",
    "unit_type_id": "rpm"
  },
  {
    "metadata": {
      "_id": "6b8e6197-c155-4c8c-a931-aa5b85bd218c",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea17"
    },
    "unit_id": "6b8e6197-c155-4c8c-a931-aa5b85bd218c",
    "unit_type_id": "rpm"
  },
  {
    "metadata": {
      "_id": "7ff4f862-7ae7-4b20-b072-0d154a9cc527",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea16"
    },
    "unit_id": "7ff4f862-7ae7-4b20-b072-0d154a9cc527",
    "unit_type_id": "rpm"
  },
  {
    "metadata": {
      "_id": "9f6b8fda-dd97-453e-8841-093d05c8eac3",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea19"
    },
    "unit_id": "9f6b8fda-dd97-453e-8841-093d05c8eac3",
    "unit_type_id": "rpm"
  },
  {
    "metadata": {
      "_id": "d35640a0-8613-44bd-9e99-141af2843087",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea14"
    },
    "unit_id": "d35640a0-8613-44bd-9e99-141af2843087",
    "unit_type_id": "rpm"
  },
  {
    "metadata": {
      "_id": "f521f967-a646-436e-ba83-1ac9ef8f7cd8",
      "_content_type_id": "rpm"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea1b"
    },
    "unit_id": "f521f967-a646-436e-ba83-1ac9ef8f7cd8",
    "unit_type_id": "rpm"
  }
]
```

Request # 6: Fetch rpm units for this repository

POST /pulp/api/v2/content/units/rpm/search/

Backend Service: pulp

Description: Search for all rpms with the previously fetched ids. We do this because historically we've wanted the list of all repo ids for each unit, which is only available via this endpoint.

Request body

```
{
  "criteria": {
    "limit": 8,
    "skip": 0,
    "fields": [
      "name",
      "version",
      "release",
      "arch",
      "epoch",
      "summary",
      "sourcerpm",
      "checksum",
      "filename",
      "_id"
    ],
    "filters": {
      "_id": {
        "$in": [
          "085babbb-3a43-4b9c-bdfb-915fd78c7bec",
          "2b00d383-f2c7-462c-bfe3-25a008caa0fe",
          "5ee1ec3a-7f81-47e6-bbba-11b443251f29",
          "6b8e6197-c155-4c8c-a931-aa5b85bd218c"
,

          "7ff4f862-7ae7-4b20-b072-0d154a9cc527",
          "9f6b8fda-dd97-453e-8841-093dd05c8eac3",
          "d35640a0-8613-44bd-9e99-141af2843087",
          "f521f967-a646-436e-ba83-1ac9ef8f7cd8"
        ]
      }
    }
  },
  "include_repos": true
}
```

Response body

```
[
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "walrus-0.3-0.8.src.rpm",
    "name": "walrus",
    "checksum": "6e8d6dc057e3e2c9819f0dc7e6c7b7f86bf2e8571bba414adec7fb621a461dfd",
    "summary": "A dummy package of walrus",
    "filename": "walrus-0.3-0.8.noarch.rpm",
    "epoch": "0",
    "version": "0.3",
    "release": "0.8",
    "_id": "085babbb-3a43-4b9c-bdfb-915fd78c7bec",
    "arch": "noarch",
    "children": {
    },
    "_href": "/pulp/api/v2/content/units/rpm/085babbb-3a43-4b9c-bdfb-915fd78c7bec/"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "penguin-0.3-0.8.src.rpm",
    "name": "penguin",
    "checksum": "3fcb2c927de9e13bf68469032a28b139d3e5ad2e58564fc210fd6e48635be694",
    "summary": "A dummy package of penguin",
    "filename": "penguin-0.3-0.8.noarch.rpm",
    "epoch": "0",
    "version": "0.3",
    "release": "0.8",
    "_id": "2b00d383-f2c7-462c-bfe3-25a008caa0fe",
    "arch": "noarch",
    "children": {
    },
    "_href": "/pulp/api/v2/content/units/rpm/2b00d383-f2c7-462c-bfe3-25a008caa0fe/"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "elephant-0.3-0.8.src.rpm",
    "name": "elephant",
    "checksum": "3e1c70cd1b421328acaf6397cb3d16145306bb95f65d1b095fc31372a0a701f3",
    "summary": "A dummy package of elephant",
    "filename": "elephant-0.3-0.8.noarch.rpm",
    "epoch": "0",
    "version": "0.3",
    "release": "0.8",
    "_id": "5ee1ec3a-7f81-47e6-bbba-11b443251f29",
    "arch": "noarch",
    "children": {
    },
    "_href": "/pulp/api/v2/content/units/rpm/5ee1ec3a-7f81-47e6-bbba-11b443251f29/"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "monkey-0.3-0.8.src.rpm",
    "name": "monkey",
    "checksum": "0e8fa50d0128fbabc7ccc5632e3fa25d39b0280169f6166cb8e2c84de8501db1",
    "summary": "A dummy package of monkey",
    "filename": "monkey-0.3-0.8.noarch.rpm",
    "epoch": "0",
    "version": "0.3",
    "release": "0.8",
    "_id": "6b8e6197-c155-4c8c-a931-aa5b85bd218c",
    "arch": "noarch",
    "children": {
    },
    "_href": "/pulp/api/v2/content/units/rpm/6b8e6197-c155-4c8c-a931-aa5b85bd218c/"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "lion-0.3-0.8.src.rpm",
    "name": "lion",
    "checksum": "12400dc95c23a4c160725a908716cd3fcdd7a8981585437ab64cd62efa3e4ae4",
    "summary": "A dummy package of lion",
    "filename": "lion-0.3-0.8.noarch.rpm",
    "epoch": "0",
    "version": "0.3",
    "release": "0.8",
    "_id": "7ff4f862-7ae7-4b20-b072-0d154a9cc527",
    "arch": "noarch",
    "children": {
    },
    "_href": "/pulp/api/v2/content/units/rpm/7ff4f862-7ae7-4b20-b072-0d154a9cc527/"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "cheetah-0.3-0.8.src.rpm",
    "name": "cheetah",
    "checksum": "422d0baa0cd9d7713ae796e886a23e17f578f924f74880debdbb7d65fb368dae",
    "summary": "A dummy package of cheetah",
    "filename": "cheetah-0.3-0.8.noarch.rpm",
    "epoch": "0",
    "version": "0.3",
    "release": "0.8",
    "_id": "9f6b8fda-dd97-453e-8841-093d05c8eac3",
    "arch": "noarch",
    "children": {
    },
    "_href": "/pulp/api/v2/content/units/rpm/9f6b8fda-dd97-453e-8841-093d05c8eac3/"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "giraffe-0.3-0.8.src.rpm",
    "name": "giraffe",
    "checksum": "f25d67d1d9da04f12e57ca323247b43891ac46533e355b82de6d1922009f9f14",
    "summary": "A dummy package of giraffe",
    "filename": "giraffe-0.3-0.8.noarch.rpm",
    "epoch": "0",
    "version": "0.3",
    "release": "0.8",
    "_id": "d35640a0-8613-44bd-9e99-141af2843087",
    "arch": "noarch",
    "children": {
    },
    "_href": "/pulp/api/v2/content/units/rpm/d35640a0-8613-44bd-9e99-141af2843087/"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "sourcerpm": "squirrel-0.3-0.8.src.rpm",
    "name": "squirrel",
    "checksum": "251768bdd15f13d78487c27638aa6aecd01551e253756093cde1c0ae878a17d2",
    "summary": "A dummy package of squirrel"
```

```
      "summary": "A dummy package of squirrel",
      "filename": "squirrel-0.3-0.8.noarch.rpm",
      "epoch": "0",
      "version": "0.3",
      "release": "0.8",
      "_id": "f521f967-a646-436e-ba83-1ac9ef8f7cd8",
      "arch": "noarch",
      "children": {
      },
      "_href": "/pulp/api/v2/content/units/rpm/f521f967-a646-436e-ba83-1ac9ef8f7cd8/"
    }
  ]
```

Request # 7: Fetch Errata ids for repository

POST /pulp/api/v2/repositories/scenario_test/search/units/

Backend Service: pulp

Description:

Request body

```
{
  "criteria": {
    "type_ids": [
      "erratum"
    ],
    "fields": {
      "unit": [

      ],
      "association":
[
        "unit_id"
      ]
    }
  }
}
```

Response body

```
[
  {
    "metadata": {
      "_id": "4b12197f-28b0-4d5c-bf8d-057dc0b378f1",
      "_content_type_id": "erratum"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea1e"
    },
    "unit_id": "4b12197f-28b0-4d5c-bf8d-057dc0b378f1",
    "unit_type_id": "erratum"
  },
  {
    "metadata": {
      "_id": "5bc4a860-2872-461a-8061-b30626274615",
      "_content_type_id": "erratum"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea1f"
    },
    "unit_id": "5bc4a860-2872-461a-8061-b30626274615",
    "unit_type_id": "erratum"
  },
  {
    "metadata": {
      "_id": "c12277ae-b619-40cc-afbc-75c92e78ca13",
      "_content_type_id": "erratum"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea1d"
    },
    "unit_id": "c12277ae-b619-40cc-afbc-75c92e78ca13",
    "unit_type_id": "erratum"
  }
]
```

Request # 8: Fetch errata units.

POST /pulp/api/v2/content/units/erratum/search/

Backend Service: pulp

Description: Using the previously fetched ids, we fetch all errata in the repo.

Request body

```
{
  "criteria": {
    "limit": 3,
    "skip": 0,
    "filters": {
      "_id": {
        "$in": [
          "4b12197f-28b0-4d5c-bf8d-057dc0b378f1",
          "5bc4a860-2872-461a-8061-b30626274615"
,
          "c12277ae-b619-40cc-afbc-75c92e78ca13"
        ]
      }
    }
  },
  "include_repos": true
}
```

Response body

```
[
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "_href": "/pulp/api/v2/content/units/erratum/4b12197f-28b0-4d5c-bf8d-057dc0b378f1/",
    "issued": "2010-11-10 00:00:00",
    "references": [
      {
        "href": "https://rhn.redhat.com/errata/RHSA-2010-0858.html",
        "type": "self",
        "id": null,
        "title": "RHSA-2010:0858"
      },
      {
        "href": "https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=627882",
        "type": "bugzilla",
        "id": "627882",
        "title": "CVE-2010-0405 bzip2: integer overflow flaw in BZ2_decompress"
      },
      {
        "href": "https://www.redhat.com/security/data/cve/CVE-2010-0405.html",
        "type": "cve",
        "id": "CVE-2010-0405",
        "title": "CVE-2010-0405"
      },
      {
        "href": "http://www.redhat.com/security/updates/classification/#important",
        "type": "other",
        "id": null,
        "title": null
      }
    ],
    "pulp_user_metadata": {
    },
    "_content_type_id": "erratum",
    "id": "RHSA-2010:0858",
    "from": "security@redhat.com",
    "severity": "Important",
    "title": "Important: bzip2 security update",
    "children": {
    },
    "version": "3",
    "reboot_suggested": false,
    "type": "security",
    "pkglist": [
      {
        "_pulp_repo_id": "scenario_test",
        "packages": [
          {
            "src": "bzip2-1.0.5-7.el6_0.src.rpm",
            "name": "bzip2-devel",
            "sum": [
              "sha256",
              "ea67c664da1ff96a6dc94d33009b73d8fab31b59824183fb45e9ba2ebf82d583"
            ],
            "filename": "bzip2-devel-1.0.5-7.el6_0.i686.rpm",
            "epoch": "0",
            "version": "1.0.5",
            "release": "7.el6_0",
            "arch": "i686"
          },
          {
            "src": "bzip2-1.0.5-7.el6_0.src.rpm",
            "name": "bzip2-libs",
            "sum": [
              "sha256",
              "c9f064a6862573fb9f2a6aff7c3621f1940b492df2edfc2ebbdc0b8305f51147"
            ]
            ],
            "filename": "bzip2-libs-1.0.5-7.el6_0.i686.rpm",
            "epoch": "0",
            "version": "1.0.5",
            "release": "7.el6_0",
            "arch": "i686"
          },
          {
            "src": "bzip2-1.0.5-7.el6_0.src.rpm",
            "name": "bzip2",
            "sum": [
              "sha256",
              "b8a3f72bc2b0d89ba737099ac98bf8d2af4bea02d31884c02db97f7f66c3d5c2"
            ],
            "filename": "bzip2-1.0.5-7.el6_0.x86_64.rpm",
            "epoch": "0",
            "version": "1.0.5",
            "release": "7.el6_0",
            "arch": "x86_64"
          },
          {
            "src": "bzip2-1.0.5-7.el6_0.src.rpm",
            "name": "bzip2-devel",
            "sum": [
              "sha256",
              "7f63124e4655b7c92d23ec4c38226f5d3746568853dff750fc85e058e74b5cf6"
            ],
            "filename": "bzip2-devel-1.0.5-7.el6_0.x86_64.rpm",
            "epoch": "0",
            "version": "1.0.5",
            "release": "7.el6_0",
            "arch": "x86_64"
          },
          {
            "src": "bzip2-1.0.5-7.el6_0.src.rpm",
            "name": "bzip2-libs",
            "sum": [
              "sha256",
              "802f4399dbdd01476e254c3b32c40aff59cf5d23a45fa488c6917ce8904d6b4d"
            ],
            "filename": "bzip2-libs-1.0.5-7.el6_0.x86_64.rpm",
            "epoch": "0",
            "version": "1.0.5",
            "release": "7.el6_0",
            "arch": "x86_64"
          }
        ],
        "name": "Red Hat Enterprise Linux Server (v. 6 for 64-bit x86_64)",
        "short": "rhel-x86_64-server-6"
      }
    ],
    "status": "final",
    "updated": "2010-11-10 00:00:00",
    "description": "bzip2 is a freely available, high-quality data compressor. It provides both\nlibbz2 library must be restarted for the upd
ate to take effect.",
    "_last_updated": "2017-03-30T21:16:05Z",
    "pushcount": "",
    "rights": "Copyright 2010 Red Hat Inc",
    "solution": "Before applying this update, make sure all previously-released errata\nrelevant to your system have been applied.\n\nT
his update is available via the Red Hat Network. Details on how to\nuse the Red Hat Network to apply this update are available at\nht
tp://kbase.redhat.com/faq/docs/DOC-11259",
    "summary": "Updated bzip2 packages that fix one security issue",
    "release": "",
    "_id": "4b12197f-28b0-4d5c-bf8d-057dc0b378f1"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "_href": "/pulp/api/v2/content/units/erratum/5bc4a860-2872-461a-8061-b30626274615/",
    "issued": "2010-01-01 01:01:01",
    "references": [

    ],
    "pulp_user_metadata": {
    },
    "_content_type_id": "erratum",
    "id": "RHEA-2010:0002",
    "from": "lzap+pub@redhat.com",
    "severity": "",
    "title": "One package errata",
    "children": {
    },
    "version": "1",
    "reboot_suggested": false,
    "type": "security",
    "pkglist": [
      {
        "_pulp_repo_id": "scenario_test",
        "packages": [
          {
            "src": "http://www.fedoraproject.org",
            "name": "elephant",
            "sum": null,
            "filename": "elephant-0.3-0.8.noarch.rpm",
            "epoch": null,
            "version": "0.3",
            "release": "0.8",
            "arch": "noarch"
          }
        ],
        "name": "1",
        "short": ""
      }
    ],
    "status": "stable",
    "updated": "",
    "description": "One package errata",
    "_last_updated": "2017-03-30T21:16:05Z",
    "pushcount": "",
    "rights": "",
    "solution": "",
    "summary": "",
    "release": "1",
    "_id": "5bc4a860-2872-461a-8061-b30626274615"
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "_href": "/pulp/api/v2/content/units/erratum/c12277ae-b619-40cc-afbc-75c92e78ca13/",
    "issued": "2010-01-01 01:01:01",
    "references": [

    ],
    "pulp_user_metadata": {
    },
    "_content_type_id": "erratum",
    "id": "RHEA-2010:0001",
    "from": "lzap+pub@redhat.com",
    "severity": "",
    "title": "Empty errata",
    "children": {
    },
    "version": "1",
    "reboot_suggested": false,
    "type": "security",
    "pkglist": [

    ],
    "status": "stable",
    "updated": "",
    "description": "Empty errata",
    "_last_updated": "2017-03-30T21:16:05Z",
    "pushcount": "",
    "rights": "",
    "solution": "",
    "summary": "",
    "release": "1",
    "_id": "c12277ae-b619-40cc-afbc-75c92e78ca13"
  }
]
```

Request # 9: Fetch package group units for repository

POST /pulp/api/v2/repositories/scenario_test/search/units/

Backend Service: pulp

Description:

Request body

```
{
  "criteria": {
    "type_ids": [
      "package_group
"
    ],
    "fields": {
      "unit": [

      ],
      "association": [
        "unit_id"
      ]
    }
  }
}
```

Response body

```
[
  {
    "metadata": {
      "_id": "919baa7e-e944-4602-b3ed-3aef2ae5b509",
      "_content_type_id": "package_group"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea21"
    },
    "unit_id": "919baa7e-e944-4602-b3ed-3aef2ae5b509",
    "unit_type_id": "package_group"
  },
  {
    "metadata": {
      "_id": "c55b5b16-7501-4863-8e3f-f7520bc795fd",
      "_content_type_id": "package_group"
    },
    "_id": {
      "$oid": "58dd7595e6919db96421ea20"
    },
    "unit_id": "c55b5b16-7501-4863-8e3f-f7520bc795fd",
    "unit_type_id": "package_group"
  }
]
```

Request # 10: Fetch package group ids for repository

POST /pulp/api/v2/content/units/package_group/search/

Backend Service: pulp

Description: Using the previously fetched ids, we fetch all package groups in the repo.

Request body

```
{
  "criteria": {
    "limit": 2,
    "skip": 0,
    "filters": {
      "_id": {
        "$in": [
          "919baa7e-e944-4602-b3ed-3aef2ae5b509"
,
          "c55b5b16-7501-4863-8e3f-f7520bc795fd"
        ]
      }
    }
  },
  "include_repos": true
}
```

Response body

```
[
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "mandatory_package_names": [
      "elephant,giraffe,cheetah,lion,monkey,penguin,squirrel,walrus",
      "penguin"
    ],
    "repo_id": "scenario_test",
    "name": "mammal",
    "user_visible": true,
    "default": true,
    "_last_updated": "2017-03-30T18:41:08Z",
    "children": {
    },
    "optional_package_names": [

    ],
    "translated_name": {
    },
    "_href": "/pulp/api/v2/content/units/package_group/919baa7e-e944-4602-b3ed-3aef2ae5b509/",
    "translated_description": {
    },
    "pulp_user_metadata": {
    },
    "default_package_names": [

    ],
    "_content_type_id": "package_group",
    "id": "mammal",
    "_id": "919baa7e-e944-4602-b3ed-3aef2ae5b509",
    "display_order": 1024,
    "conditional_package_names": [

    ]
  },
  {
    "repository_memberships": [
      "scenario_test"
    ],
    "mandatory_package_names": [
      "penguin"
    ],
    "repo_id": "scenario_test",
    "name": "bird",
    "user_visible": true,
    "default": true,
    "_last_updated": "2017-03-30T18:41:08Z",
    "children": {
    },
    "optional_package_names": [

    ],
    "translated_name": {
    },
    "_href": "/pulp/api/v2/content/units/package_group/c55b5b16-7501-4863-8e3f-f7520bc795fd/",
    "translated_description": {
    },
    "pulp_user_metadata": {
    },
    "default_package_names": [

    ],
    "_content_type_id": "package_group",
    "id": "bird",
    "_id": "c55b5b16-7501-4863-8e3f-f7520bc795fd",
    "display_order": 1024,
    "conditional_package_names": [

    ]
  }
]
```

Request # 11: Request applicability generation for consumers bound to the repository

POST /pulp/api/v2/repositories/actions/content/regenerate_applicability//

Backend Service: pulp

Description:

Request body

```
{
  "parallel": true,
  "repo_criteria": {
    "filters": {
      "id": {
        "$in": [
          "scenario_test
"
        ]
      }
    }
  }
}
```

Response body

```
{
  "group_id": "b0e268a7-f4bf-4598-90a3-5fc3b562cc95",
  "_href": "/pulp/api/v2/task_groups/b0e268a7-f4bf-4598-90a3-5fc3b562cc95/"
}
```

Request # 12: Monitor task group status

GET /pulp/api/v2/task_groups/b0e268a7-f4bf-4598-90a3-5fc3b562cc95/state_summary/

Backend Service: pulp

Description: Monitor status of the applicability generation

Request body

None

Response body

```
{
  "accepted": 0,
  "finished": 0,
  "running": 0,
  "canceled": 0,
  "waiting": 0,
  "skipped": 0,
  "suspended": 0,
  "error": 0,
  "total": 0
}
```

Request # 13: Fetch repository details

GET /pulp/api/v2/repositories/scenario_test/

Backend Service: pulp

Description: Unclear why

Request body

None

Response body

```
{
  "scratchpad": {
    "checksum_type": "sha256"
  },
  "display_name": "Scenario yum product",
  "description": null,
  "distributors": [
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:15:43Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/distributors/scenario_test_clone/"
,
      "last_override_config": {
      },
      "last_publish": null,
      "distributor_type_id": "yum_clone_distributor",
      "auto_publish": false,
      "scratchpad": {
      },
      "_ns": "repo_distributors",
      "_id": {
        "$oid": "58dd757f418a8a04f88de7da"
      },
      "config": {
        "destination_distributor_id": "scenario_test"
      },
      "id": "scenario_test_clone"
    },
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:16:05Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/distributors/scenario_test/",
      "last_override_config": {
      },
      "last_publish": "2017-03-30T21:16:05Z",
      "distributor_type_id": "yum_distributor",
      "auto_publish": true,
      "scratchpad": {
      },
      "_ns": "repo_distributors",
      "_id": {
        "$oid": "58dd757f418a8a04f88de7d8"
      },
      "config": {
        "checksum_type": "sha256",
        "protected": true,
        "http": false,
        "https": true,
        "relative_url": "scenario_test"
      },
      "id": "scenario_test"
    },
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:15:43Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/distributors/export_distributor/",
      "last_override_config": {
      },
      "last_publish": null,
      "distributor_type_id": "export_distributor",
      "auto_publish": false,
      "scratchpad": {
      },
      "_ns": "repo_distributors",
      "_id": {
        "$oid": "58dd757f418a8a04f88de7d9"
      },
      "config": {
        "http": false,
        "relative_url": "scenario_test",
        "https": false
      },
      "id": "export_distributor"
    }
  ],
  "last_unit_added": "2017-03-30T21:16:05Z",
  "notes": {
    "_repo-type": "rpm-repo"
  },
  "last_unit_removed": null,
  "content_unit_counts": {
    "package_group": 2,
    "distribution": 1,
    "package_category": 1,
    "rpm": 8,
    "erratum": 3
  },
  "_ns": "repos",
  "importers": [
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:15:43Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/importers/yum_importer/",
      "_ns": "repo_importers",
      "importer_type_id": "yum_importer",
      "last_override_config": {
        "num_threads": 4,
        "validate": true
      },
      "last_sync": "2017-03-30T21:16:05Z",
      "scratchpad": {
        "repomd_revision": 1321893800
      },
      "_id": {
        "$oid": "58dd757f418a8a04f88de7d7"
      },
      "config": {
        "feed": "file:///var/www/test_repos/zoo",
        "ssl_validation": true,
        "remove_missing": true,
        "download_policy": "immediate"
      },
      "id": "yum_importer"
    }
  ],
  "locally_stored_units": 15,
  "locally_stored_units": 15,
  "_id": {
    "$oid": "58dd757f418a8a04f88de7d6"
  },
  "total_repository_units": 15,
  "id": "scenario_test",
  "_href": "/pulp/api/v2/repositories/scenario_test/"
}
```

Repository Create

Request # 1: Create Content

POST /candlepin/owners/scenario_test/content/

Backend Service: candlepin

Description: Create Content object for repository, for subscription-manager content access

Request body

```
{
  "name": "Scenario yum product",
  "contentUrl": "/custom/Scenario_Product/Scenario_yum_product",
  "type": "yum",
  "label": "scenario_test_Scenario_Product_Scenario_yum_product",
  "metadataExpire": 1,
  "vendor": "Custom"
}
```

Response body

```
{
  "created": "2017-03-30T21:15:43+0000",
  "updated": "2017-03-30T21:15:43+0000",
  "uuid": "4028f9515b20f31e015b2112fba4000d",
  "id": "1490908543901",
  "type": "yum",
  "label": "scenario_test_Scenario_Product_Scenario_yum_product",
  "name": "Scenario yum product",
  "vendor": "Custom",
  "contentUrl": "/custom/Scenario_Product/Scenario_yum_product",
  "requiredTags": null,
  "gpgUrl": null,
  "metadataExpire": 1,
  "modifiedProductIds": [

  ],
  "arches": null,
  "releaseVer": null
}
```

Request # 2: Associate content object

POST /candlepin/owners/scenario_test/products/272869743822/content/1490908543901

Backend Service: candlepin

Description: Add the Content object to the product

Request body

None

Response body

```
{
  "created": "2017-03-30T21:15:42+0000",
  "updated": "2017-03-30T21:15:44+0000",
  "uuid": "4028f9515b20f31e015b2112fc03000e",
  "id": "272869743822",
  "name": "Scenario Product",
  "multiplier": 1,
  "attributes": [
    {
      "created": "2017-03-30T21:15:42+0000",
      "updated": "2017-03-30T21:15:44+0000",
      "name": "arch",
      "value": "ALL"
    }
  ],
  "dependentProductIds": [

  ],
  "href": "/products/4028f9515b20f31e015b2112fc03000e",
  "productContent": [
    {
      "content": {
        "created": "2017-03-30T21:15:43+0000",
        "updated": "2017-03-30T21:15:43+0000",
        "uuid": "4028f9515b20f31e015b2112fba4000d",
        "id": "1490908543901",
        "type": "yum",
        "label": "scenario_test_Scenario_Product_Scenario_yum_product"
,
        "name": "Scenario yum product",
        "vendor": "Custom",
        "contentUrl": "/custom/Scenario_Product/Scenario_yum_product",
        "requiredTags": null,
        "gpgUrl": null,
        "metadataExpire": 1,
        "modifiedProductIds": [

        ],
        "arches": null,
        "releaseVer": null
      },
      "enabled": true
    }
  ]
}
```

Request # 3: Retrieve candlepin environment

GET /candlepin/environments/119c4753ff6d3b7bd0b76de6d5a5f94a

Backend Service: candlepin

Description: Retrieve the environment object (TODO WHY?)

Request body

None

Response body

```
{
  "owner": {
    "id": "4028f9515b20f31e015b2112f2550004",
    "key": "scenario_test",
    "displayName": "scenario_test",
    "href": "/owners/scenario_test"
  },
  "name": "Library",
  "description": null,
  "id": "119c4753ff6d3b7bd0b76de6d5a5f94a",
  "environmentContent": [

  ],
  "created": "2017-03-30T21:15:41+0000",
  "updated": "2017-03-30T21:15:41+0000"
}
```

Request # 4: Create Pulp Repository

POST /pulp/api/v2/repositories/

Backend Service: pulp

Description:

Request body

```
{
  "id": "scenario_test",
  "display_name": "Scenario yum product",
  "importer_type_id": "yum_importer",
  "importer_config": {
    "feed": "file:///var/www/test_repos/zoo",
    "ssl_ca_cert": null,
    "ssl_client_cert": null,
    "ssl_client_key": null,
    "ssl_validation": true,
    "download_policy": "immediate",
    "remove_missing": true
  },
  "notes": {
    "_repo-type": "rpm-repo"
  },
  "distributors": [
    {
      "distributor_type_id": "yum_distributor",
      "distributor_config": {
        "relative_url": "scenario_test",
        "http": false,
        "https": true,
        "protected": true
      },
      "auto_publish": true,
      "distributor_id": "scenario_test"
    },
    {
      "distributor_type_id": "export_distributor",
      "distributor_config": {
        "http": false,
        "https": false,
        "relative_url": "scenario_test"
      },
      "auto_publish": false,
      "distributor_id": "export_distributor"
    },
    {
      "distributor_type_id": "yum_clone_distributor"
,
      "distributor_config": {
        "destination_distributor_id": "scenario_test"
      },
      "auto_publish": false,
      "distributor_id": "scenario_test_clone"
    }
  ]
}
```

Response body

```
{
  "scratchpad": {
  },
  "display_name": "Scenario yum product",
  "description": null,
  "last_unit_added": null,
  "notes": {
    "_repo-type": "rpm-repo"
  },
  "last_unit_removed": null,
  "content_unit_counts": {
  },
  "_ns": "repos",
  "_id": {
    "$oid": "58dd757f418a8a04f88de7d6"
  },
  "id": "scenario_test",
  "_href": "/pulp/api/v2/repositories/scenario_test/"
}
```

Request # 5: Add Content To environment

POST /candlepin/environments/119c4753ff6d3b7bd0b76de6d5a5f94a/content

Backend Service: candlepin

Description: Associates content object to “Library” environment, so it is accessible by clients registered to Library

Request body

```
[
  {
    "contentId": "1490908543901"
  }
]
```

Response body

```
{
  "id": "regen_entitlement_cert_of_envda871caa-8138-4edd-acc9-605195f992a4",
  "state": "CREATED",
  "startTime": null,
  "finishTime": null,
  "result": null,
  "principalName": "foreman_admin",
  "targetType": null,
  "targetId": null,
  "ownerId": null,
  "resultData": null,
  "statusPath": "/jobs/regen_entitlement_cert_of_envda871caa-8138-4edd-acc9-605195f992a4",
  "done": false,
  "group": "async group",
  "created": "2017-03-30T21:15:44+0000",
  "updated": "2017-03-30T21:15:44+0000"
}
```

Request # 6: Fetch repository information

GET /pulp/api/v2/repositories/scenario_test/

Backend Service: pulp

Description:

Request body

None

Response body

```
{
  "scratchpad": {
  },
  "display_name": "Scenario yum product",
  "description": null,
  "distributors": [
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:15:43Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/distributors/scenario_test_clone/"
,
      "last_override_config": {
      },
      "last_publish": null,
      "distributor_type_id": "yum_clone_distributor",
      "auto_publish": false,
      "scratchpad": {
      },
      "_ns": "repo_distributors",
      "_id": {
        "$oid": "58dd757f418a8a04f88de7da"
      },
      "config": {
        "destination_distributor_id": "scenario_test"
      },
      "id": "scenario_test_clone"
    },
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:15:43Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/distributors/scenario_test/",
      "last_override_config": {
      },
      "last_publish": null,
      "distributor_type_id": "yum_distributor",
      "auto_publish": true,
      "scratchpad": {
      },
      "_ns": "repo_distributors",
      "_id": {
        "$oid": "58dd757f418a8a04f88de7d8"
      },
      "config": {
        "protected": true,
        "http": false,
        "https": true,
        "relative_url": "scenario_test"
      },
      "id": "scenario_test"
    },
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:15:43Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/distributors/export_distributor/",
      "last_override_config": {
      },
      "last_publish": null,
      "distributor_type_id": "export_distributor",
      "auto_publish": false,
      "scratchpad": {
      },
      "_ns": "repo_distributors",
      "_id": {
        "$oid": "58dd757f418a8a04f88de7d9"
      },
      "config": {
        "http": false,
        "relative_url": "scenario_test",
        "https": false
      },
      "id": "export_distributor"
    }
  ],
  "last_unit_added": null,
  "notes": {
    "_repo-type": "rpm-repo"
  },
  "last_unit_removed": null,
  "content_unit_counts": {
  },
  "_ns": "repos",
  "importers": [
    {
      "repo_id": "scenario_test",
      "last_updated": "2017-03-30T21:15:43Z",
      "_href": "/pulp/api/v2/repositories/scenario_test/importers/yum_importer/",
      "_ns": "repo_importers",
      "importer_type_id": "yum_importer",
      "last_override_config": {
      },
      "last_sync": null,
      "scratchpad": null,
      "_id": {
        "$oid": "58dd757f418a8a04f88de7d7"
      },
      "config": {
        "feed": "file:///var/www/test_repos/zoo",
        "ssl_validation": true,
        "remove_missing": true,
        "download_policy": "immediate"
      },
      "id": "yum_importer"
    }
  ],
  "locally_stored_units": 0,
  "_id": {
    "$oid": "58dd757f418a8a04f88de7d6"
  },
  "total_repository_units": 0,
  "id": "scenario_test",
  "_href": "/pulp/api/v2/repositories/scenario_test/"
}
```

Request # 7: Publish repository metadata

POST /pulp/api/v2/repositories/scenario_test/actions/publish/

Backend Service: pulp

Description: Publish the repository, generating empty metadata so that existing clients of that product do not get a 404.

Request body

```
{
  "id": "scenario_test",
  "override_config": {
    "force_full": false
  }
}
```

Response body

```
{
  "spawned_tasks": [
    {
      "_href": "/pulp/api/v2/tasks/dfdb7cae-e465-4885-b3a4-7bb29343b69e/"
,
      "task_id": "dfdb7cae-e465-4885-b3a4-7bb29343b69e"
    }
  ],
  "result": null,
  "error": null
}
```

Request # 8: Poll Task

GET /pulp/api/v2/tasks/dfdb7cae-e465-4885-b3a4-7bb29343b69e/

Backend Service: pulp

Description: Monitor the task status of the publish

Total Requests for this URL: 6

Request body

None

Response body

```
{
  "exception": null,
  "task_type": "pulp.server.managers.repo.publish.publish",
  "_href": "/pulp/api/v2/tasks/dfdb7cae-e465-4885-b3a4-7bb29343b69e/",
  "task_id": "dfdb7cae-e465-4885-b3a4-7bb29343b69e",
  "tags": [
    "pulp:repository:scenario_test",
    "pulp:action:publish"
  ],
  "finish_time": "2017-03-30T21:15:44Z",
  "_ns": "task_status",
  "start_time": "2017-03-30T21:15:44Z",
  "traceback": null,
  "spawned_tasks": [

  ],
  "progress_report": {
    "scenario_test": [
      {
        "num_success": 1,
        "description": "Initializing repo metadata",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "e3a30d24-1111-4080-92f8-0a3dc979ae9c",
        "num_processed": 1
      },
      {
        "num_success": 0,
        "description": "Publishing Distribution files",
        "step_type": "distribution",
        "items_total": 0,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "c2edbc90-716d-4570-a442-bd0c22748f88",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Publishing RPMs",
        "step_type": "rpms",
        "items_total": 0,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "bfea7f15-21d9-4cf0-af2a-5cce0c1c172b",
        "num_processed": 0
      },
      {
```

```
      {
        "num_success": 0,
        "description": "Publishing Delta RPMs",
        "step_type": "drpms",
        "items_total": 1,
        "state": "SKIPPED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "47621356-70b3-4e77-a8f2-ee1ec0c5c2ee",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Publishing Errata",
        "step_type": "errata",
        "items_total": 0,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "8e46b5e2-89f5-4ef5-80bd-0d2aab9dd2a0",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Publishing Comps file",
        "step_type": "comps",
        "items_total": 0,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "050aba89-abec-4949-b3cf-28409d9519e0",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Publishing Metadata.",
        "step_type": "metadata",
        "items_total": 0,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "70efd383-905d-4fe2-8c9b-4e18311e76d5",
        "num_processed": 0
      },
      {
        "num_success": 1,
        "description": "Closing repo metadata",
        "step_type": "close_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "e3ae5910-8f8d-4e53-abd6-007a9495e0b4",
        "num_processed": 1
      },
      {
        "num_success": 0,
        "description": "Generating sqlite files",
        "step_type": "generate sqlite",
        "items_total": 1,
        "state": "SKIPPED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "ae023865-a176-4434-b8ab-77a080306602",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Generating HTML files",
        "step_type": "repoview",
        "items_total": 1,
        "state": "SKIPPED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "num_failures": 0,
        "step_id": "4d7ca329-7b33-478e-a0a5-777771d8c7c6",
        "num_processed": 0
      },
      {
        "num_success": 1,
        "description": "Publishing files to web",
        "step_type": "publish_directory",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "0989644f-85dc-43ff-9e0f-10c814a8d5e8",
        "num_processed": 1
      },
      {
        "num_success": 1,
        "description": "Writing Listings File",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "04b588ee-11fe-4b09-8a42-2ea465247c0b",
        "num_processed": 1
      }
    ]
  },
  "queue": "reserved_resource_worker-1@dev.example.com.dq",
  "state": "finished",
  "worker_name": "reserved_resource_worker-1@dev.example.com",
  "result": {
    "result": "success",
    "exception": null,
    "repo_id": "scenario_test",
    "started": "2017-03-30T21:15:44Z",
    "_ns": "repo_publish_results",
    "completed": "2017-03-30T21:15:44Z",
    "traceback": null,
    "distributor_type_id": "yum_distributor",
    "summary": {
      "generate sqlite": "SKIPPED",
      "initialize_repo_metadata": "FINISHED",
      "rpms": "FINISHED",
      "close_repo_metadata": "FINISHED",
      "drpms": "SKIPPED",
      "comps": "FINISHED",
      "distribution": "FINISHED",
      "repoview": "SKIPPED",
      "publish_directory": "FINISHED",
      "errata": "FINISHED",
      "metadata": "FINISHED"
    },
    "error_message": null,
    "distributor_id": "scenario_test",
    "id": "58dd7580418a8a0648b9bc40",
    "details": [
      {
        "num_success": 1,
        "description": "Initializing repo metadata",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "e3a30d24-1111-4080-92f8-0a3dc979ae9c",
        "num_processed": 1
      },
      {
        "num_success": 0,
        "description": "Publishing Distribution files",
        "step_type": "distribution",
        "items_total": 0,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "c2edbc90-716d-4570-a442-bd0c22748f88",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Publishing RPMs",
      "description": "Publishing RPMs",
      "step_type": "rpms",
      "items_total": 0,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "bfea7f15-21d9-4cf0-af2a-5cce0c1c172b",
      "num_processed": 0
    },
    {
      "num_success": 0,
      "description": "Publishing Delta RPMs",
      "step_type": "drpms",
      "items_total": 1,
      "state": "SKIPPED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "47621356-70b3-4e77-a8f2-ee1ec0c5c2ee",
      "num_processed": 0
    },
    {
      "num_success": 0,
      "description": "Publishing Errata",
      "step_type": "errata",
      "items_total": 0,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "8e46b5e2-89f5-4ef5-80bd-0d2aab9dd2a0",
      "num_processed": 0
    },
    {
      "num_success": 0,
      "description": "Publishing Comps file",
      "step_type": "comps",
      "items_total": 0,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "050aba89-abec-4949-b3cf-28409d9519e0",
      "num_processed": 0
    },
    {
      "num_success": 0,
      "description": "Publishing Metadata.",
      "step_type": "metadata",
      "items_total": 0,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "70efd383-905d-4fe2-8c9b-4e18311e76d5",
      "num_processed": 0
    },
    {
      "num_success": 1,
      "description": "Closing repo metadata",
      "step_type": "close_repo_metadata",
      "items_total": 1,
      "state": "FINISHED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "e3ae5910-8f8d-4e53-abd6-007a9495e0b4",
      "num_processed": 1
    },
    {
      "num_success": 0,
      "description": "Generating sqlite files",
      "step_type": "generate sqlite",
      "items_total": 1,
      "state": "SKIPPED",
      "error_details": [

      ],
      "details": "",
      "num_failures": 0,
      "step_id": "ae023865-a176-4434-b8ab-77a080306602",
        "num_processed": 0
      },
      {
        "num_success": 0,
        "description": "Generating HTML files",
        "step_type": "repoview",
        "items_total": 1,
        "state": "SKIPPED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "4d7ca329-7b33-478e-a0a5-777771d8c7c6",
        "num_processed": 0
      },
      {
        "num_success": 1,
        "description": "Publishing files to web",
        "step_type": "publish_directory",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "0989644f-85dc-43ff-9e0f-10c814a8d5e8",
        "num_processed": 1
      },
      {
        "num_success": 1,
        "description": "Writing Listings File",
        "step_type": "initialize_repo_metadata",
        "items_total": 1,
        "state": "FINISHED",
        "error_details": [

        ],
        "details": "",
        "num_failures": 0,
        "step_id": "04b588ee-11fe-4b09-8a42-2ea465247c0b",
        "num_processed": 1
      }
    ]
  },
  "error": null,
  "_id": {
    "$oid": "58dd7580e6919db96421ea12"
  },
  "id": "58dd7580e6919db96421ea12"
}
```