

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Katello 3.11 Installation

These instructions are for installing Katello 3.11, but the latest stable is [3.18](#).

Note: After installation of Katello, be sure to trust Katello's CA certificate on your system. This is required for the encrypted NoVNC connections. You will find `katello-server-ca.crt` in the `/pub` directory of your Katello server (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
- 8 GB of memory (12 GB highly recommended)
- The filesystem holding `/var/lib/pulp` needs to be large, but may vary depending on how many different Operating Systems you wish to 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 OSes:

OS	
CentOS 7	X
RHEL 7	X

Katello can only run on an x86\_64 operating systems.

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

## Required Repositories

Select your Operating System:

```
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://fedorapeople.org/groups/katello/releases/yum/3.11/katello/el7/x86_64/katello-repos-latest.rpm  
yum -y localinstall https://yum.theforeman.org/releases/1.21/el7/x86_64/foreman-release.rpm  
yum -y localinstall https://yum.puppetlabs.com/puppetlabs-release-pc1-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.
```

## Forklift

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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 foreman-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"\n  --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%] [.....]\nSuccess!\n\nTo finish the installation, follow these steps:\n\n1. Ensure that the foreman-installer-katello package is installed on the system.\n2. Copy ~/myproxy.example.com-certs.tar to the system myproxy.example.com\n3. Run the following commands on the Smart Proxy (possibly with the customized\nparameters, see foreman-installer --scenario foreman-proxy-content --help and\ndocumentation for more info on setting up additional services):\n\nyum -y localinstall http://katello.example.com/pub/katello-ca-consumer-latest.noarch.rpm\nsubscription-manager register --org "Default_Organization"\nforeman-installer --scenario foreman-proxy-content\n    --foreman-proxy-content-parent-fqdn "katello.example.com"\n    --foreman-proxy-register-in-foreman "true"\n    --foreman-proxy-foreman-base-url  "https://katello.example.com"\n    --foreman-proxy-trusted-hosts    "katello.example.com"\n    --foreman-proxy-trusted-hosts    "myproxy.example.com"\n    --foreman-proxy-oauth-consumer-key "UVrAZfMaCfBiiWejoUVLYCZHT2xhzuFV"\n    --foreman-proxy-oauth-consumer-secret "ZhH8p7M577ttNU3WmUGWASag3JeXKgUX"\n\n    --certs-tar-file      "/root/myproxy.example.com-certs.tar"\n\nThe 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.](#)

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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:  ▾

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-5.noarch.rpm  
yum install -y https://yum.theforeman.org/client/1.21/el5/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
```

For Suse Clients, only `katello-host-tools` is supported:

```
zypper install katello-host-tools
```

## 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.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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/3.11/katello/el7/x86_64/katello-repos-latest.rpm  
yum update -y https://yum.theforeman.org/releases/1.21/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 3.11.

```
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 3.11.

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.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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/3.11/katello/el7/x86_64/katello-repos-latest.rpm  
yum update -y https://yum.theforeman.org/releases/1.21/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 3.11

```
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 3.11 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 manually below to aid in troubleshooting.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Client Upgrade

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

## Step 1 - Update Repositories

### Manually Added Clients

Update the Katello client release packages:

Select your Operating System:  

```
yum update -y https://yum.theforeman.org/client/1.21/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 3.11 client repositories to your Katello for each version needed. After you create the new repositories, they will then need to be added to the relevant content view(s) and the older versions removed. At this point, a new version of the content view can be published and promoted to the appropriate environments. Once the new package is available the clients can be updated following the next steps.

## Step 2: Update Packages

Clean the yum cache

```
yum clean all
```

Update packages:

```
yum update katello-agent
```

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Katello 3.11 (Beautiful Disaster) Release Notes

## Bug Fixes

This release contains a substantial set of bug fixes. Some notable ones include:

- hammer erratum list showing errata across Organizations when –organization-id is specified
- Vague errors given when publishing a Content View when Docker repositories have conflicting container names
- Capsule sync not being triggered when uploading a packing to a custom repository in Library
- Potential race condition when creating multiple repositories on the same Product
- Subscription Status widget showing incorrect information on the Dashboard

For the full release notes see the [Changelog](#).

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Hammer

## What is the CLI?

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

## Installation

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

Select your Operating System:  ▾

```
yum -y --disablerepo="*" --enablerepo=rhel-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/3.11/katello/el7/x86_64/katello-repos-latest.rpm  
yum -y localinstall https://yum.theforeman.org/releases/1.21/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](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Activation Keys

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

- Lifecycle Environment
- Content View
- Subscriptions
- Repository Enablement
- Host Collections

## Definitions

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

## General Features

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

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

## Create an Activation Key

To create a new key,

- navigate to: Content > Activation Keys
- click [New Activation Key](#)

The screenshot shows the Foreman web interface with a dark blue header bar. The header includes the 'FOREMAN' logo, a user icon labeled 'Admin User', and navigation links for 'Default\_Organization', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', and 'Administer'. Below the header, the main content area has a title 'Activation Keys'. On the left is a search bar and a button for '+ New Activation Key'. The main form is titled 'New Activation Key' and contains the following fields:

- Name\***: A text input field containing 'research'.
- Content Host Limit**: A checkbox labeled 'Unlimited Content Hosts' which is checked.
- Description**: A text area containing the placeholder text 'This is the activation key used by the research department.'
- Environment**: A selection tool showing three environments: 'Library' (selected with a checked checkbox), 'dev', and 'test'.
- Content View**: A dropdown menu set to 'Default Organization View'.

At the bottom of the form are 'Cancel' and 'Save' buttons.

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

The following example would use CV\_B’s content view:

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

Or equivalently:

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

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

## Add Subscriptions to an Activation Key

To add subscriptions to a key:

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

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

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

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

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

Finally, when ‘Auto-Attach’ is disabled, all subscriptions on the activation key will be 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.

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, Administer, and Admin User. Below the navigation is a search bar and a message indicating 1 total result. On the left, a sidebar lists activation keys, with 'research' selected and highlighted in blue. The main content area is titled 'Activation Key: research'. It has tabs for Details, Subscriptions (which is selected), Product Content, Host Collections, and Associations. Under 'Activation Key Type:', there's a section for 'Auto-Attach' with a 'Yes' option and an info box explaining it. Below this is a 'List/Remove' section with a 'Add' button. A table titled 'Subscriptions' shows one entry: 'Red Hat Enterprise Linux Server, Premium (8 sockets) (Up to 4 guests)' with a quantity of 1, attached, physical type, starts on 12/31/14, expires on 12/31/15, and account number 5481114.

## Change Repository Enablement for an Activation Key

To change repository enablement settings using a key:

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

View current settings:

**FOREMAN**

Default\_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

## Activation Keys

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

Name
research >

### Activation Key: research

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

Details Subscriptions **Product Content** Host Collections

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

**RHEL 6**

Content

Enabled by Default Defaults to Yes

Change current settings:

**FOREMAN**

Default\_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

## Activation Keys

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

Name
research >

### Activation Key: research

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

Details Subscriptions **Product Content** Host Collections

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

**RHEL 6**

Content

Enabled by Default

# Add Host Collections to an Activation Key

To add Host Collections to a key:

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

Activation Keys

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

0 Selected | Deselect All + New Activation Key

Name

research >

Activation Key: research

Details Subscriptions Product Content Host Collections

Host Collection Management

List/Remove Add

Filter

2 Selected | Deselect All Add Selected

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

## Register a Content Host using an Activation Key

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

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

[Click here for more information](#)

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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](#)

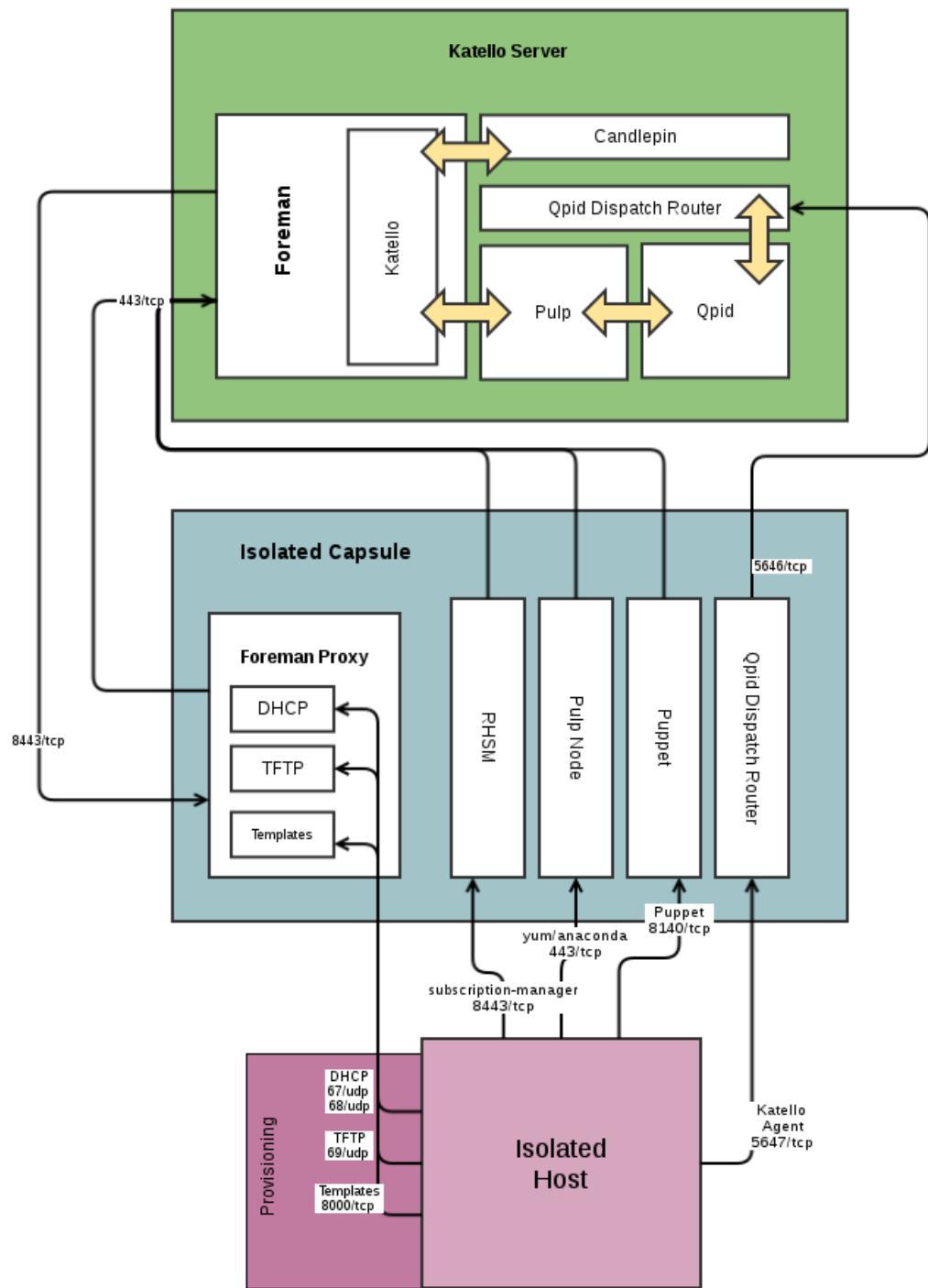
## Removal

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.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Content

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

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

- Reduce bandwidth usage and increase download speed by having client machines pull updates from Katello
- Provision hosts using local Repositories covered HERE TODO
- Customize content locally, covered HERE TODO

## Definitions

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

## Creating a Product

From the web UI, navigate to:

Content > Products > New Product (top right)

The screenshot shows the Katello web interface with a blue header bar. The title bar says 'New Product'. The main content area is titled 'Products' and shows a table with one row. The row has a checkbox column and a 'Name' column containing 'Epel'. To the right of the table is a 'New Product' dialog box. The dialog box has fields for 'Name\*' (Epel), 'Label\*' (Epel), 'GPG Key' (dropdown menu), 'Sync Plan' (dropdown menu with '+ New Sync Plan' link), and 'Description' (text area). At the bottom of the dialog box are 'Cancel' and 'Save' buttons. The top right corner of the dialog box has a 'Close' button.

## 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 recommend to leave this checked.
- URL: If you are syncing from an external Repository (yum or puppet), this would be filled in. This can be changed, added,

or removed later. For example if you are wanting to create a mirror of EPEL, you would set this to 'https://dl.fedoraproject.org/pub/epel/6/x86\_64/'.

The screenshot shows the Foreman web interface with the title 'New Repository'. The URL in the address bar is 192.168.122.75:3000/katello/products#/products/127/repositories/new. The top navigation bar includes 'MyCompany', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', 'Administer', and a user icon for 'Admin User'. Below the navigation is a search bar with placeholder 'Search...' and a results count 'Showing 1 of 1 (1 Total)'. A toolbar on the right contains '0 Selected | Deselect All', 'Bulk Actions', 'Repo Discovery', and '+ New Product'. On the left, a sidebar titled 'Name' shows a list with 'Epel' selected. The main content area is titled 'Product Epel' and contains tabs for 'Details', 'Repositories', and 'Tasks'. The 'Details' tab is active, showing the configuration for a 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: checked
- GPG Key: (empty dropdown)

At the bottom are 'Cancel' and 'Save' buttons.

## Syncing a Repository

From the web UI, navigate to:

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

The screenshot shows the Foreman web interface for managing repositories. The URL is 192.168.122.75:3000/katello/products#/products/127/repositories/80. The top navigation bar includes 'MyCompany', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', 'Administer', and a user icon for 'Admin User'. The main title is 'Products' with a sub-section 'Product Epel'. On the left, a sidebar lists repository names, with 'Epel' selected. The main content area displays the 'Basic Information' for the 'Epel x86\_64' repository, including fields like Name, Label, Type, URL, Publish via HTTPS, Publish via HTTP, Published At, and GPG Key. To the right, there are 'Content Counts' for Packages (0), Errata (0), and Package Groups (0). Buttons for 'Sync Now' and 'Remove Repository' are also present.

The progress will be displayed:

This screenshot shows the 'Task Details' for the synchronization of the 'Epel' repository. The URL is 192.168.122.75:3000/katello/products#/products/127/tasks/fbd1fd1d-8f38-4a4e-aae0-ae4a74df6846. The top navigation bar and sidebar are identical to the previous screenshot. The main content area shows the 'Task Details' for the task 'fb...'. The 'Action Type' is 'Synchronize', 'User' is 'admin', 'Started At' is '4/15/14 12:52 PM', 'Finished At' is '...', 'Parameters' include 'repository 'Epel x86\_64' ; product 'Epel' ; organization 'MyCompany'', 'State' is 'running', and 'Result' is 'pending'. A progress bar indicates the task is still pending. Below the task details, a 'Details' section shows 'New packages: 37/10694 (6.68 MB/9.78 GB)'.

## Syncing multiple repositories

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

Content > Sync Status

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

The screenshot shows the Foreman Sync Status interface. At the top, there are tabs for 'Repository: Epel x86\_6' and 'Sync Status'. The URL in the address bar is 'abed.usersys.redhat.com:3000/katello/sync\_management#!='.

The main area is titled 'Sync Status' and contains a table with the following data:

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

Below the table are buttons for 'Collapse All', 'Expand All', 'Select None', 'Select All', and 'Only show syncing.' A 'Synchronize Now' button is located at the bottom right of the table area.

## Uploading RPM Content

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

## Uploading Puppet Content

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

The screenshot shows the Foreman web interface with the title 'New Repository'. The URL is <http://abed.usersys.redhat.com:3000/katello/products#/products/175/repositories/new>. The top navigation bar includes 'MyCompany', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', 'Administer', and a user icon for 'Admin User'. A sidebar on the left shows 'Name' (selected), 'Configuration >', and 'Epel'. The main content area is titled 'Product Configuration' with tabs 'Details', 'Repositories' (selected), and 'Tasks'. A sub-section titled '<< Add New Repository' contains fields: 'Name\*' (Puppet), 'Label\*' (Puppet), 'Type\*' (puppet), 'URL' (empty), 'Publish via HTTP' (checked), and 'GPG Key' (empty). Buttons 'Cancel' and 'Save' are at the bottom.

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

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

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

The screenshot shows the Foreman web interface with the title 'Repository: Puppet'. The URL is <http://abed.usersys.redhat.com:3000/katello/products#/products/175/repositories/30>. The top navigation bar and sidebar are identical to the previous screenshot. The main content area is titled 'Product Configuration' with tabs 'Details' (selected), 'Repositories', and 'Tasks'. It shows basic information for the 'Puppet' repository: Name (Puppet), Label (Puppet), Type (puppet), URL (empty), Publish via HTTPS (true), Publish via HTTP (checked), Published At (<http://localhost/pulp/repos/MyCompany/Library/custom/Configurati/Puppet>), and GPG Key (empty). It also displays content counts (Puppet Modules: 0) and a 'Sync Now' button. On the right, there is a 'Content Type' section and an 'Upload Puppet Module' form with 'Choose File' and 'Upload' buttons. A note at the bottom states: 'All dates and times below are relative to this server.'

# Subscribing a System to a Product for yum content

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

## Scheduling Repository Synchronization

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

### Creating a Sync Plan

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

To create a Sync Plan, navigate to:

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

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

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

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

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

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

### Assigning a Sync Plan to a Product

Navigate to:

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

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

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

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

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

Under "Product Management", there are buttons for "List/Remove" and "Add". A filter input field is present. Below these are buttons for "1 Selected | Deselect All" and "Add Selected".

A table lists repositories:

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

## GPG Keys

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Content Hosts

## What is a Host?

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

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

## What are Content Hosts?

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

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

## How is a Content Host registered?

Subscription Manager is the client for Katello that handles registration.

## Installing Subscription Manager

Depending on your Operating System, for:

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

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

## Registering with Subscription Manager

First install the bootstrap rpm from your Katello server:

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

Then register:

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

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

## Registering to a Content View

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

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

## Registering without using a username and password

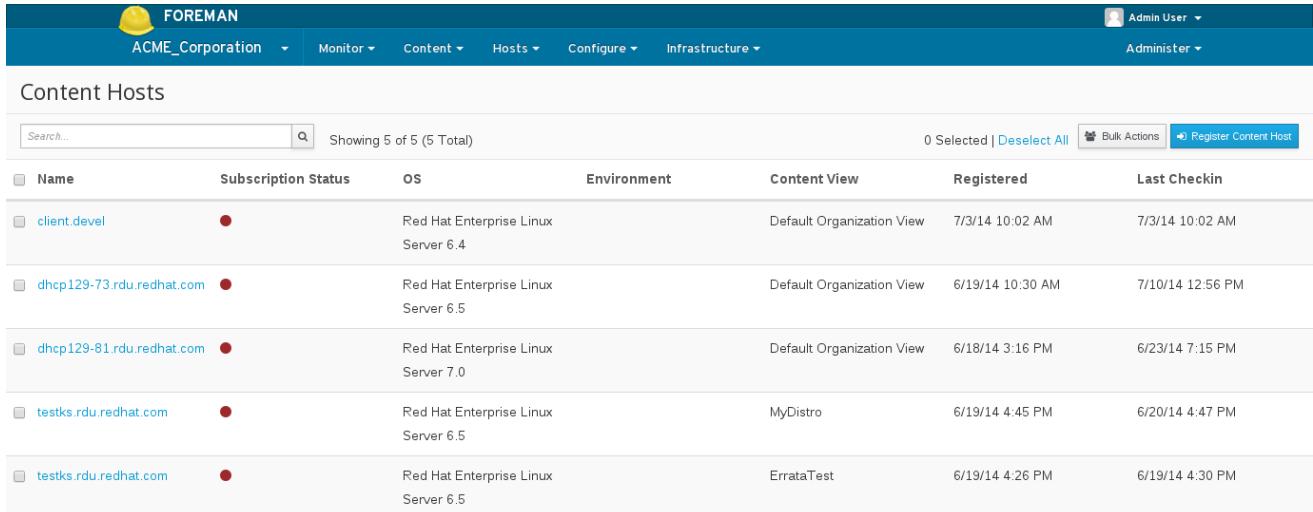
Activation Keys allow you to register and consume content without using a username and password. To create an Activation Key see the [Activation Key Guide](#)

Once you have created an activation key, register with:

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

## Actions with registered Content Hosts

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



The screenshot shows the Foreman web interface with the title "FOREMAN" and the organization "ACME\_Corporation". The top navigation bar includes links for Monitor, Content, Hosts, Configure, and Infrastructure. On the right, there are "Admin User" and "Administer" dropdowns. The main content area is titled "Content Hosts" and displays a table of five registered hosts. The columns are: Name, Subscription Status, OS, Environment, Content View, Registered, and Last Checkin. The hosts listed are: client.devel, dhcp129-73.rdu.redhat.com, dhcp129-81.rdu.redhat.com, testks.rdu.redhat.com, and testks.rdu.redhat.com. The table includes search and filter options at the top, and bulk actions like "Deselect All" and "Register Content Host" at the bottom.

Name	Subscription Status	OS	Environment	Content View	Registered	Last Checkin
client.devel	●	Red Hat Enterprise Linux Server 6.4		Default Organization View	7/3/14 10:02 AM	7/3/14 10:02 AM
dhcp129-73.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		Default Organization View	6/19/14 10:30 AM	7/10/14 12:56 PM
dhcp129-81.rdu.redhat.com	●	Red Hat Enterprise Linux Server 7.0		Default Organization View	6/18/14 3:16 PM	6/23/14 7:15 PM
testks.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		MyDistro	6/19/14 4:45 PM	6/20/14 4:47 PM
testks.rdu.redhat.com	●	Red Hat Enterprise Linux Server 6.5		ErrataTest	6/19/14 4:26 PM	6/19/14 4:30 PM

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

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

**Content Hosts**

Search...  Showing 5 of 5 (5 Total) 0 Selected | Deselect All Bulk Actions Register Content Host

Name
client.devel
<b>dhcp129-73.rdu.redhat.com</b>
dhcp129-81.rdu.redhat.com
testks.rdu.redhat.com
testks.rdu.redhat.com

**Content Host dhcp129-73.rdu.redhat.com**

Unregister Content Host × Close

Details Provisioning Details Subscriptions Host Collections Tasks Packages Errata Product Content

**Basic Information**

Name	dhcp129-73.rdu.redhat.com	<input checked="" type="checkbox"/>
UUID	2df6452c-852d-4a58-b6ac-728362f913c1	<input checked="" type="checkbox"/>
Description	Initial Registration Params	<input checked="" type="checkbox"/>
Type	Virtual Guest	

**Content Host Content**

Release Version	<input checked="" type="checkbox"/>
Content View	Default Organization View
Environment	<input checked="" type="checkbox"/>

Library Dev

**Subscriptions**

Subscription Status	invalid
Details	Red Hat Enterprise Linux Server - Not supported by a valid subscription.
Auto-Attach	Yes
Service Level	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Activation Keys	None

**Content Host Status**

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

**Content Host Properties**

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

**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**

Search... Showing 20 of 35 (35 Total) 0 Selected | Deselect All Bulk Actions Register Content Host

Name
client.devel
<b>dhcp129-73.rdu.redhat.com</b>
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**

Unregister Content Host Close

Details Provisioning Details Subscriptions Host Collections Tasks Packages Errata Product Content

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:

List/Remove Add

Run Auto-Attach

Subscriptions

Search... Showing 2 of 2 (2 Total) Subscriptions 1 Selected | Deselect All Add Selected

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

testProduct

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

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

## Package Management

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

**Content Hosts**

Search... Showing 5 of 5 (5 Total) 0 Selected | Deselect All Bulk Actions Register Content Host

Name
client.devel
<b>dhcp129-73.rdu.redhat.com</b>
dhcp129-81.rdu.redhat.com
testks.rdu.redhat.com
testks.rdu.redhat.com

**Content Host dhcp129-73.rdu.redhat.com**

Unregister Content Host Close

Details Provisioning Details Subscriptions Host Collections Tasks Packages Errata Product Content

Package Actions

Package Install Package/Group Name Perform

Installed Packages

Filter Showing 416 Packages Update All

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

From here you can:

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

## View and Install Applicable Errata

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

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

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

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

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

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

## Change Host Collection Asssignments

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

Content Hosts

Search... Showing 5 of 5 (5 Total)

0 Selected | Deselect All Bulk Actions Register Content Host

Unregister Content Host Close

Name	Description	Capacity
Database Servers		1 / Unlimited

## Bulk Actions

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

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

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

## Content Hosts

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

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

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

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

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Content Views

What can a Content View be used for?

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

## Definitions

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

## General Workflow

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

To utilize Content Views for filtering and snapshotting:

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

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

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

## Creating a Content View

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

Content > Content Views

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

The screenshot shows the Foreman web interface for creating a new Content View. On the left, there's a sidebar with categories: Name, COMPOSITE, Puppet Modules, and Yum Repos. The main content area is titled "New Content View". It has fields for "Name\*" (set to "New Content View"), "Label\*" (set to "New\_Content\_View"), and "Description" (containing the text "This is my new content view, not sure what it will contain yet."). Below these is a checkbox for "Composite View?" which is unchecked. At the bottom are "Cancel" and "Save" buttons.

From the CLI:

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

## Creating a Composite Content View

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

From the CLI:

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

## Adding Repositories

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

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

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

New Content View

Repository Selection

Name	Product	Last Sync	Sync State	Content
<input checked="" type="checkbox"/> Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server	Red Hat Enterprise Linux Server	N/A	N/A	0 Packages 0 Errata
<input type="checkbox"/> 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)

New Content View

Puppet Modules (selected)

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.

**FOREMAN**

content views Monitor Content Hosts Configure Infrastructure Administer

Content Views

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

New Content View

Versions Content Puppet Modules History Details Tasks

Module List

Select A New Puppet Module To Add

Filter Showing 4 of 4 (4 Total)

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

**FOREMAN**

content views Monitor Content Hosts Configure Infrastructure Administer

Content Views

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

New Content View

Versions Content Puppet Modules History Details Tasks

Module List / Select Module / apache

Select an Available Version of apache

Filter

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

The screenshot shows the Foreman web interface for managing content views. On the left, a sidebar lists options: Name, COMPOSITE, New Content View (selected), Puppet Modules, and Yum Repos. The main content area is titled 'New Content View' and shows a table of 'Currently Selected Puppet Modules'. One module, 'apache' by 'puppetlabs', is listed with its version as 'Latest (Currently 1.0.1)'. Buttons for 'Select new version' and 'Remove Module' are visible.

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

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

Then add the puppet module:

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

## Adding Content Views to a Composite Content View

Adding a version of a Content View to a Composite Content View means whenever the Composite Content View is published, all of the content contained within the specific version of that Content View is contained in the Composite Content View. If the Content Views contained within the Composite Content View are updated (i.e. a new version is published) or if their content is updated after publishing the Composite Content View, the Composite Content View will only contain the versions of the Content View(s) prior to syncing. A new version of the Composite Content View must be published in order for it to get the updated Content Views.

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

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

The screenshot shows the Foreman web interface under the 'Content Views' section. A composite content view named 'COMPOSITE' is selected. The interface displays a list of components that can be added to the composite view, including 'Puppet Modules' and 'Yum Repos'. A message at the top states: 'In order to add a content view to a composite view you must first publish an initial version of the content view.' A table below lists the components with their details.

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

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

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

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

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

## Creating a filter

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

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

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

**FOREMAN**

content views   Monitor   Content   Hosts   Configure   Infrastructure

Admin User   Administer

### Content Views

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

Name  
 COMPOSITE  
 New Content View >  
 Puppet Modules  
 Yum Repos

New Content View

Versions Content ▾ Puppet Modules History Details Tasks

Filters

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

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

**FOREMAN**

content views   Monitor   Content   Hosts   Configure   Infrastructure

Admin User   Administer

### Content Views

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

Name  
 COMPOSITE  
 New Content View >  
 Puppet Modules  
 Yum Repos

New Content View

Versions Content ▾ Puppet Modules History Details Tasks

< Filters List

Add New Filter

Name\* New Filter

Content Type\* Package

Type\* Exclude

Description Here is my new filter!

Cancel Save

The screenshot shows the Foreman web interface under the 'Content Views' section. On the left, there's a sidebar with options like 'Name', 'COMPOSITE', 'New Content View >', 'Puppet Modules', and 'Yum Repos'. The 'New Content View >' option is highlighted. The main area is titled 'New Content View' and shows a 'Filters / New Filter' dialog. This dialog has tabs for 'Content', 'Puppet Modules', 'History', 'Details', and 'Tasks', with 'Content' selected. Below this is a table for 'New Filter (Exclude Packages)'. The table has columns for 'Package Name' and 'Detail'. It contains two rows: one for 'something-else' with a dropdown set to 'Equal To' and a value of '10.0.0', and another for 'httpd' with a dropdown set to 'All Versions'. There are buttons for '+ Add' and '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)

New Content View

Versions Content Puppet Modules History Details Tasks

Filters / New Filter

New Filter (Exclude Packages)

Packages Affected repositories

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

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

All Products Filter Update Repositories

From the CLI, adding a Content View Filter:

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

## Publishing a Content View

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

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

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

New Content View

Versions Content Puppet Modules History Details Tasks

Publish New Version

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

Version Details

Version 1

Comment initial version.

Cancel Save

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

From the CLI:

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

## Registering a Content Host

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

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

This would register the Content Host to the Library environment and the my\_rhel\_view Content View.

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

Hosts > Content Hosts > Select the desired Content Host

The screenshot shows the Foreman web interface with the 'Systems' page selected. A specific system, 'dhcp129-211.rdu.redhat.com', is highlighted. In the top right, there's a 'Content View' dropdown set to 'Yum Repos'. Below it, under 'Environment', 'Library' is checked. The 'System Properties' and 'Networking' sections provide detailed information about the host.

From the CLI:

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

## Promoting a Content View

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

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

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

**FOREMAN**

content views   Monitor   Content   Hosts   Configure   Infrastructure

Admin User Administer

### Content Views

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

Name  
 COMPOSITE  
 **New Content View >**  
 Puppet Modules  
 Yum Repos

New Content View

Versions Content Puppet Modules History Details Tasks

Filter

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

**FOREMAN**

content views   Monitor   Content   Hosts   Configure   Infrastructure

Admin User Administer

### Content Views

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

Name  
 COMPOSITE  
 **New Content View >**  
 Puppet Modules  
 Yum Repos

New Content View

Versions Content Puppet Modules History Details Tasks

[« Back to Version List](#)

Promote Version 1

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

Library  test  prod

New Content View

Version	Status	Environments	Content	Author	Actions
Version 1	Promoting to 1 environment.	• Library • test	0 Packages 0 Errata (▲ 0 ⚡ 0 🔍 0)		<a href="#">Promote</a>

To promote a Content View in the CLI:

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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.

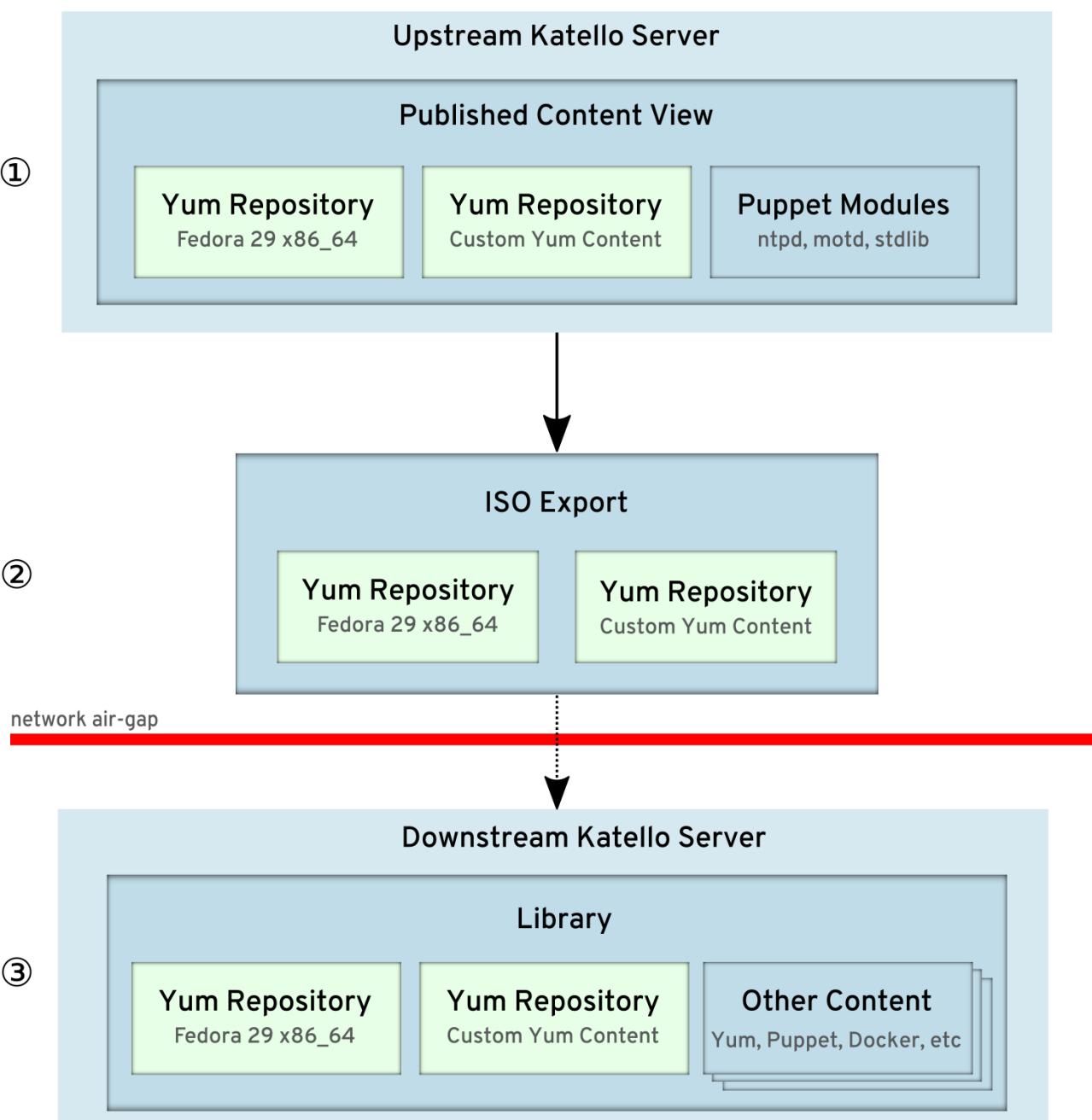
## Intro

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Docker Management

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

## What is Docker?

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

## How to sync a Docker repository

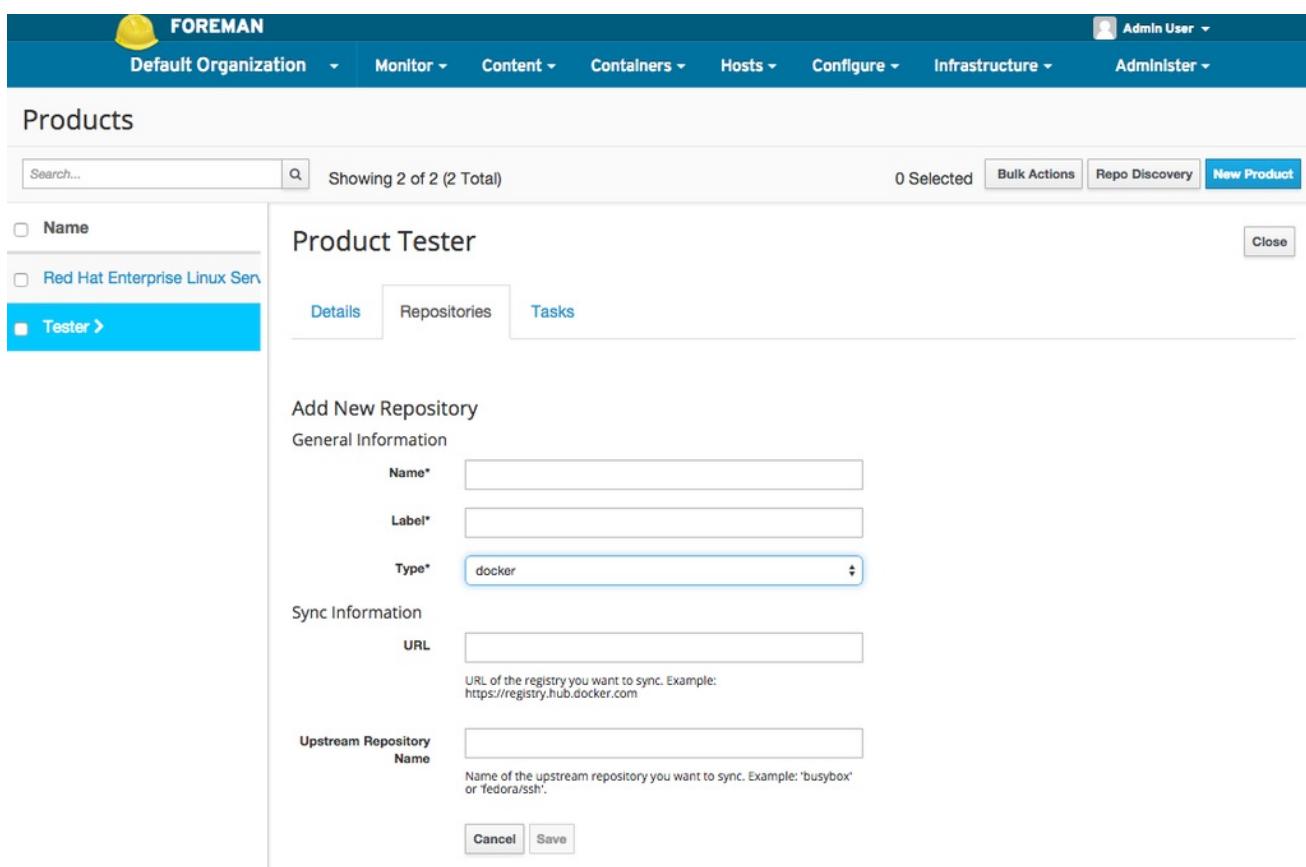
The easiest way to get Docker content into Katello is to sync it in. You can either sync Docker content from the Red Hat CDN (if you have subscriptions for the content) or from a registry such as Docker Hub.

### Red Hat Docker Images

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

### Docker Hub/Docker Registry

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



The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, and Administer. On the right, there's a user dropdown labeled "Admin User". Below the navigation, the main area is titled "Products". A search bar and a "Showing 2 of 2 (2 Total)" message are visible. On the left, there's a sidebar with checkboxes for "Name", "Red Hat Enterprise Linux Ser", and "Tester >". The "Tester >" checkbox is selected and highlighted in blue. The main content area is titled "Product Tester" and contains tabs for "Details", "Repositories", and "Tasks". Under "Repositories", a sub-section titled "Add New Repository" is shown. It has two tabs: "General Information" and "Sync Information". In "General Information", there are fields for "Name\*", "Label\*", and "Type\*" (set to "docker"). In "Sync Information", there are fields for "URL" and "Upstream Repository Name". Below these fields, there are "Cancel" and "Save" buttons. The entire "Add New Repository" section is enclosed in a light gray box.

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

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

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

# How to Upload Docker Images

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

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

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

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

# How to Publish and Promote Docker Content

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

The screenshot shows the Foreman interface with the 'Content Views' page selected. A Content View named 'Test' is displayed. The 'Docker Content' tab is active. A table lists five Docker repositories:

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

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

# How to View and Pull Docker Content

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

## Docker Tags



Showing 25 of 25 (25 Total)

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



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


**FOREMAN**

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

Admin User

## Docker Tags

Search...  Showing 25 of 25 (25 Total)

Name	redis:latest			<input type="button" value="x Close"/>
redis:2.8.14				
redis:2.8.15				
redis:2.8.16				
redis:2.8.17				
redis:2.8.18				
redis:2.8.19				
redis:2.8.6				
redis:2.8.7				
redis:2.8.8				
redis:2.8.9				
busybox:buildroot-2013.08.1				
busybox:buildroot-2014.02				
tester:latest				
<b>redis:latest &gt;</b>				
busybox:latest				
busybox:ubuntu-12.04				
busybox:ubuntu-14.04				

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

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

## How to Provision Docker Content

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

First, proceed to the new Container page by accessing it from the Containers menu at the top. Then, select the Local Content tab on the second step. This will allow you to select a Docker image from a published Katello repository which is in an environment/content view/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.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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".

The screenshot shows the Foreman web interface with the 'Edit User' page open. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, Administer, and a user icon labeled 'Admin User'. Below the navigation is a breadcrumb trail: Administer > Users > Edit User. The main content area has tabs for User (selected), Mail Preferences, Locations, Organizations, and Roles. Under the 'General' tab, there's a checkbox for 'Mail enabled' which is checked. In the 'Notifications' tab, there are five entries: 'Katello host advisory' set to 'Weekly' (description: 'A summary of available and applicable errata for your hosts'); 'Katello promote errata' set to 'Subscribe' (description: 'A post-promotion summary of hosts with available errata'); 'Katello sync errata' set to 'Subscribe' (description: 'A summary of new errata after a repository is synchronized'); 'Puppet error state' set to 'No emails' (description: 'A notification when a host reports a puppet error'); and 'Puppet summary' set to 'Weekly' (description: 'A summary of eventful puppet reports'). At the bottom of the form are 'Cancel' and 'Submit' buttons.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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 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: libtiff 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

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Containers, Hosts, Configure, Infrastructure, Administer, and a user dropdown. Below the navigation is a search bar and a message indicating 40 of 2809 total errata. A sidebar on the left lists various errata IDs, with RHSA-2014:1948 selected and highlighted in blue. The main content area displays detailed information about RHSA-2014:1948, including its ID, type (Security Advisory), severity (Important), and issue date (12/1/14). It also includes a 'Topic' section describing the update to NSS, NSS-util, and NSS-softokn packages to fix the CVE-2014-3566 issue. A 'Description' section provides technical details about the update, mentioning support for TLS\_FALLBACK\_SCSV and the prevention of SSL 3.0 downgrade attacks.

# View Affected Content Hosts

To view the Affected Content Hosts of an Errata:

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

Note the following option:

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

**FOREMAN**

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

Errata

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

Errata ID

RHSA-2014:1984

RHSA-2014:1983

RHBA-2014:1970

RHSA-2014:1974

RHBA-2014:1964

RHBA-2014:1965

RHBA-2014:1967

RHBA-2014:1961

RHSA-2014:1919

**RHSA-2014:1948 >**

RHEA-2014:1918

RHBA-2014:1909

RHSA-2014:1911

RHBA-2014:1886

RHBA-2014:1884

RHBA-2014:1883

RHBA-2014:1875

RHBA-2014:1867

RHEA-2014:1869

RHSA-2014:1870

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

Details Content Hosts **Repositories**

Apply To Content Hosts

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

Filter by Environment

Search... Showing 1 of 1 (1 Total) 0 Selected + Apply to Hosts

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

## View Repositories Containing Errata

To view the Repositories Containing an Errata:

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

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

**FOREMAN**

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

Errata

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

Errata ID

RHSA-2014:1984

RHSA-2014:1983

RHBA-2014:1970

RHSA-2014:1974

RHBA-2014:1964

RHBA-2014:1965

RHBA-2014:1967

RHBA-2014:1961

RHSA-2014:1919

**RHSA-2014:1948 >**

RHEA-2014:1918

RHBA-2014:1909

RHSA-2014:1911

RHBA-2014:1886

RHBA-2014:1884

RHBA-2014:1883

RHBA-2014:1875

RHBA-2014:1867

RHEA-2014:1869

RHSA-2014:1870

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

Details Content Hosts **Repositories**

Repositories Containing Errata RHSA-2014:1948

Library Default Organization View

Search... Showing 1 of 1 (1 Total) 0 Selected

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

The screenshot shows the Foreman web interface under the 'Content' tab, specifically the 'Errata' section. At the top, there's a navigation bar with 'Default Organization', 'Monitor', 'Content', 'Containers', 'Hosts', 'Configure', and 'Infrastructure'. On the right, it shows 'Admin User' and 'Administrator'. Below the header, the page title is 'Errata'. There are filters for 'All Repositories' (set to 'All'), 'Applicable' (checked), and 'Installable' (unchecked). It displays 'Showing 19 of 19 (19 Total)'.

The main content area has a sidebar on the left listing various errata IDs, with 'RHSA-2014:1948' selected and highlighted in blue. The main panel shows a summary message: 'Important: nss, nss-util, and nss-softokn security, bug fix, and enhancement update'. Below this are tabs for 'Details', 'Content Hosts' (which is selected), and 'Repositories'. Under 'Content Hosts', there's a checkbox for '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.' A dropdown for 'Filter by Environment' and a search bar are present. The results table shows one host selected: 'walden-rhel7.rdu.redhat.com' (Name), 'Red Hat Enterprise Linux Server 7.0' (OS), 'dev' (Environment), and 'RHEL' (Content View). There are buttons for '1 Selected' and 'Apply to Hosts'.

## Errata

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

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

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

**Details** **Content Hosts** **Repositories**

[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.

**Cancel** **Confirm**

## 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

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

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

**FOREMAN**

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

### Errata

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

1 Selected

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

### Apply Errata

[Errata List](#) » [Select Content Hosts](#)

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

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

1 Selected

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

**FOREMAN**

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

### Errata

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

1 Selected

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

### Apply Errata

[Errata List](#) » [Select Content Hosts](#) » [Confirm](#)

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.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Glossary

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

## Activation Key

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

## Application Lifecycle Environment

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

## Attach

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

## Capsule

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

## Change Set

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

## Compute Profile

Default attributes for new virtual machines on a compute resource.

## Compute Resource

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

## Content

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

## Content Delivery Network (CDN)

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

## Content View

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

## External Node Classifier

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

## Factor

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

## Hammer

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

## Host

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

## Host Group

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

## Location

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

## Library

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

## Manifest

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

## Organization

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

## Permission

The ability to perform an action.

## Product

A collection of content repositories.

**Promote**

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

**Provisioning Template**

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

**Puppet Agent**

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

**Puppet Class**

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

**Puppet Manifest**

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

**Puppet Master**

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

**Puppet Module**

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

**Pulp Node**

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

**Repository**

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

**Role**

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

**Smart Proxy**

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

**Smart Variable**

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

**Standard Operating Environment (SOE)**

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

**Subscription**

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

**Syncing**

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

**Sync Plans**

Scheduled execution of syncing content.

**Usergroup**

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

**User**

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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.

The screenshot shows the Foreman web interface. The top navigation bar includes 'Default Organization' and 'Default Location'. On the left, a sidebar menu lists 'Monitor', 'Content' (which is selected), 'Hosts', 'Configure', 'Infrastructure', 'Administer', and 'Toolbox'. The main content area has a title 'Create Content Credential' and a sub-section 'Content Credential > New Content Credential'. A form is displayed with fields for 'Name' (containing 'My GPG Key') and 'Type' (set to 'GPG Key'). Below these is a large text area containing a PGP public key block, which is a long string of characters starting with '-----BEGIN PGP PUBLIC KEY BLOCK-----'. At the bottom of the form are buttons for 'Choose File' (with 'No file chosen'), 'Upload Content Credential file', 'Save' (highlighted in blue), and '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

Details Repositories Tasks

Basic Information

Name: Katello  
Label: Katello  
GPG Key: My GPG Key

Sync Status

Sync Plan:  
Sync Interval:  
Last Sync:  
Next Sync:  
Sync State:

SSL CA Cert:  
SSL Client Cert:  
SSL Client Key:  
Description:  
Number of Repositories: 1  
Active Tasks: 0

Select Action

## 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

Details Products Repositories

Filter...

Name	Used as	Repositories
Katello	GPG Key	1

Remove Content Credential

# 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**

The screenshot shows the Foreman web interface. The left sidebar has links for Monitor, Content (which is selected), Hosts, Configure, Infrastructure, Administer, and Toolbox. The main content area is titled 'SSL-cert' and shows the 'Content Credential > SSL-cert > Repositories' path. There are tabs for Details, Products, and Repositories, with 'Repositories' being active. A 'Filter...' input field is present. A 'Remove Content Credential' button is in the top right. The table below lists one repository:

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Managing Content Hosts using Host Collections

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

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

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

## Definitions

- [Content Host](#)

## General Features

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

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

## Create a Host Collection

To create a new collection,

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

Note the following option:

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

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with the Foreman logo, the organization name "Default\_Organization", and links for Monitor, Content, Hosts, Configure, Infrastructure, Administer, and a user profile. Below the navigation is a search bar and a message indicating 0 selected hosts. A prominent modal window titled "New Host Collection" is open, prompting for a "Name\*" (set to "research"), "Content Host Limit" (set to "Unlimited Content Hosts"), and a "Description". The modal includes "Cancel" and "Save" buttons.

## 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

Search... Showing 1 of 1 (1 Total) 0 Selected | Deselect All + New Host Collection

Name

research >

Host Collection: research

Details Content Hosts Collection Actions

List/Remove Add

Search... Showing 2 of 2 (2 Total) Content Hosts 2 Selected | Deselect All Add Selected

Name	Environment	Content View
<input checked="" type="checkbox"/> host1.example.com	Library	Default Organization View
<input checked="" type="checkbox"/> 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

Search...  Showing 1 of 1 (1 Total) 0 Selected | Deselect All + New Host Collection

Name

research >

Host Collection: research

Details Content Hosts Collection Actions

List/Remove Add

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

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

Copy  
New Name: development  
Create Cancel

## Perform Actions on a Host Collection

To perform an action on Content Hosts within a collection:

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

 FOREMAN

Default\_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾ Administer ▾

## Host Collections

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

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

### Host Collection: research

Details Content Hosts **Collection Actions**

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

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

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Lifecycle Environments

What can a Lifecycle Environments be used for?

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

## Definitions

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

## General Workflow

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

## Viewing the list of lifecycle environments

From the web UI, navigate to:

Content -> Lifecycle Environments



The screenshot shows the Foreman web interface with a dark blue header. The header includes the 'FOREMAN' logo, a user dropdown ('Admin User'), and a navigation bar with links like 'Default\_Organization', 'Monitor', 'Content', 'Hosts', 'Configure', 'Infrastructure', and 'Administrator'. Below the header, the main content area has a light gray background. The title 'Lifecycle Environment Paths' is at the top. Underneath, there's a section titled 'Environment Path' with a table-like structure. The first and only row in the table contains the word 'Library' followed by a small blue plus sign icon. In the top right corner of this section, there is a blue button with white text that says '+ New Environment Path'.

## Creating a lifecycle environment

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



## Lifecycle Environment Paths

Environment Path



+ New Environment Path

Close

### Create Environment

Name\*

Label\*

Description

# Creating a lifecycle environment path

Click on the New Environment Path



## Lifecycle Environment Paths

Environment Path



+ New Environment Path

Close

### Create Environment

Name\*

Label\*

Description



FOREMAN

Default\_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾

Admin User ▾ Administer ▾

Lifecycle Environment Paths

Create Successful.

Environment Path

Library > Dev2 > +

Library > Dev > QA > Prod > +

## View/Updating environment name

Click on the name of the environment.

FOREMAN

Default\_Organization ▾ Monitor ▾ Content ▾ Hosts ▾ Configure ▾ Infrastructure ▾

Admin User ▾ Administer ▾

Lifecycle Environment Paths

+ New Environment Path

Environment Path

Library > Dev2 > +

Library > Dev > QA > Prod > +

Environment Dev

Name: Dev

Label: Dev

Description:

Close

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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](#)

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Managing Puppet Content

## Importing the Puppet Forge

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

To import the puppet forge navigate to

Content > Products

Click on the [+New Product](#) button.

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

The screenshot shows the Foreman web interface. At the top, there's a navigation bar with links for Default Organization, Monitor, Content, Hosts, Configure, Infrastructure, Administer, and a user icon. Below the navigation is a search bar and a message indicating 1 of 1 total results. On the left, there's a sidebar with a 'Products' section containing a 'Puppet' item. The main content area is titled 'Product Puppet'. It has three tabs: Details (selected), Repositories, and Tasks. Under the 'Repositories' tab, there's a form for adding a new repository. The form fields are: Name\* (forged), Label\* (forge), Type\* (puppet), and URL (https://forge.puppetlabs.com/). There are 'Cancel' and 'Save' buttons at the bottom of the form. The background of the main content area is light gray, and the overall interface is clean and modern.

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.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# Red Hat Content

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

## Definitions

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

## General Workflow

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

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

## Create a Subscription Manifest Using the Red Hat Customer Portal

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

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

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

## Import the Subscription Manifest

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

To import a manifest,

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

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

## Enable Red Hat Repositories

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

To enable Red Hat repositories,

- navigate to: Content > Red Hat Repositories
- select the content type: RPMs, Source RPMs, Debug RPMs, Beta, ISOs or Other
- select one or more Red Hat products (e.g. Red Hat Enterprise Linux Server)
- select one or more Repository Sets (e.g. Red Hat Enterprise Linux 6 Server (RPMs))
- select one or more Repositories (e.g. Red Hat Enterprise Linux 6 Server RPMs x86\_64 6Server)

Note:

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

Katello

fortello.devel:3000/katello/providers/redhat\_provider#!=

**FOREMAN**

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

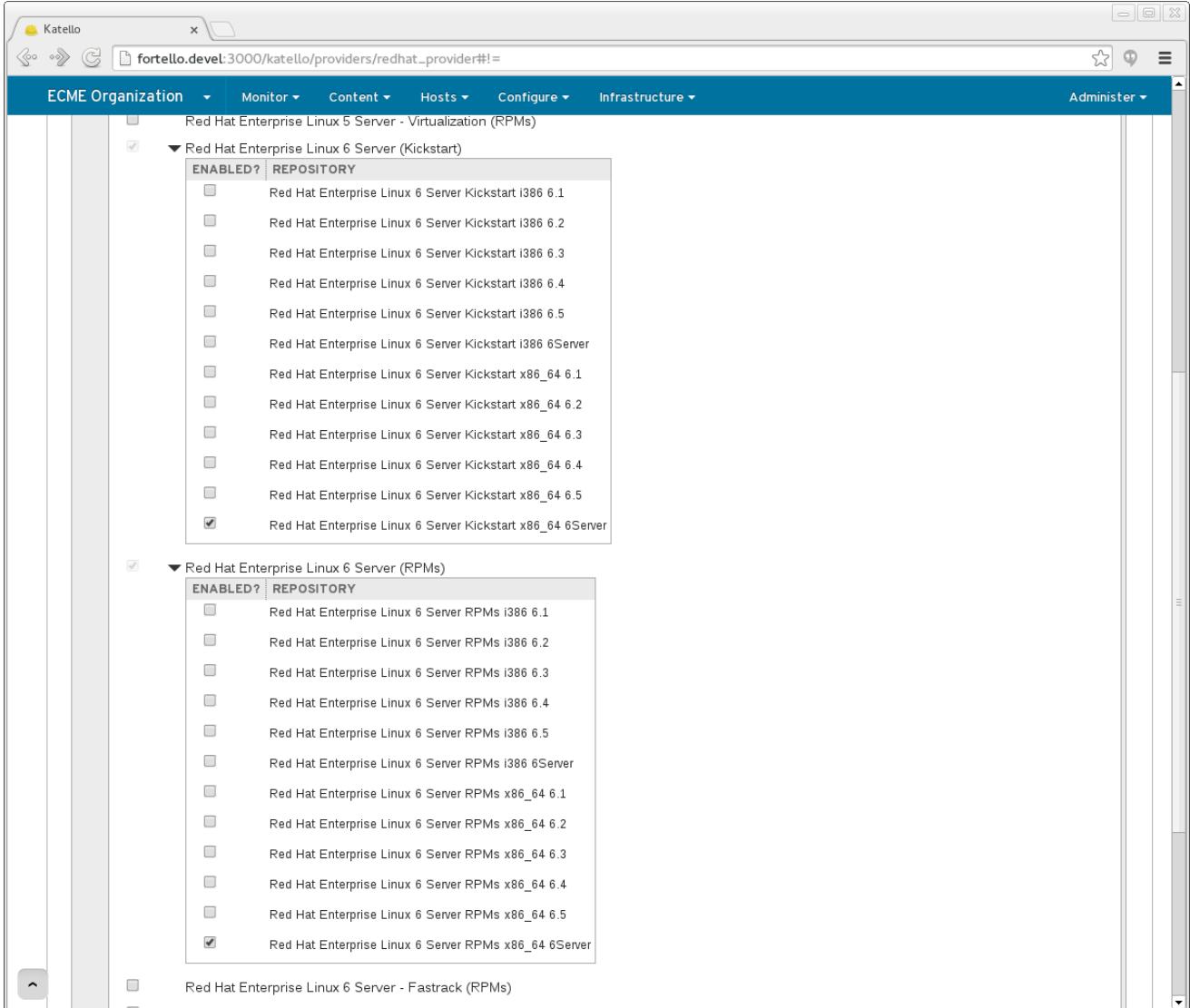
Expand each Red Hat Product below to examine the different repository sets available. When enabling a repository set, the different repositories within are discovered and may be enabled individually. X

**Enable Red Hat Repositories**

RPMs Source RPMs Debug RPMs Beta ISOs Other

**PRODUCT**

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



## Synchronize Repositories

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

To sync multiple repositories as well as track their progress,

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

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

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

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

## Schedule Repository Synchronization

### Creating a Sync Plan

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

To create a Sync Plan:

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

Note the following options:

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

New Sync Plan

fortello.devel:3000/katello/sync\_plans#/new

FOREMAN

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

Sync Plans

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

**New Sync Plan**

Name\* Weekly Sync

Description

Interval weekly

Start Date\* 2014-04-15

Start Time (-0400 GMT)\*  
17 : 06

Cancel Save

## Assigning a Sync Plan to a Red Hat Product

To assign a sync plan to a product,

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

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

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

## Attach a Content Host to a Product for Red Hat Content

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

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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 with 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”.

SUSE Customer Center account X

---

Name *	<input type="text"/>
Login *	<input type="text"/>
Use your 'Organization credentials' obtained from the SUSE Customer Center.	
Password *	<input type="password"/>
Base URL *	<input type="text" value="https://scc.suse.com"/>
<input type="button" value="Test Connection"/>	
<input type="button" value="Cancel"/> <input type="button" value="Submit"/>	

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

Filter ...
x
 Search
▼
Add SCC account

Name	Products	Last synced	Actions
SUSE	235	2018-01-26 14:56:08 UTC	<input style="width: 100px; height: 20px; border: none; background-color: #f0f0f0; padding: 2px 5px; margin-right: 5px;" type="button" value="Select products"/> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block; width: 150px; height: 20px; vertical-align: middle; position: relative;"> <span style="position: absolute; top: -5px; left: 50%; transform: translateX(-50%);">▼</span> </div> <div style="display: inline-block; vertical-align: middle; margin-top: 5px;"> <input style="width: 100px; height: 20px; border: none; background-color: #f0f0f0; padding: 2px 5px; margin-right: 5px;" type="button" value="Sync"/> <input style="width: 100px; height: 20px; border: none; background-color: #f0f0f0; padding: 2px 5px;" type="button" value="Delete"/> </div>

20 ▼ per page
f 1

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

[Cancel](#) [Submit](#)

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

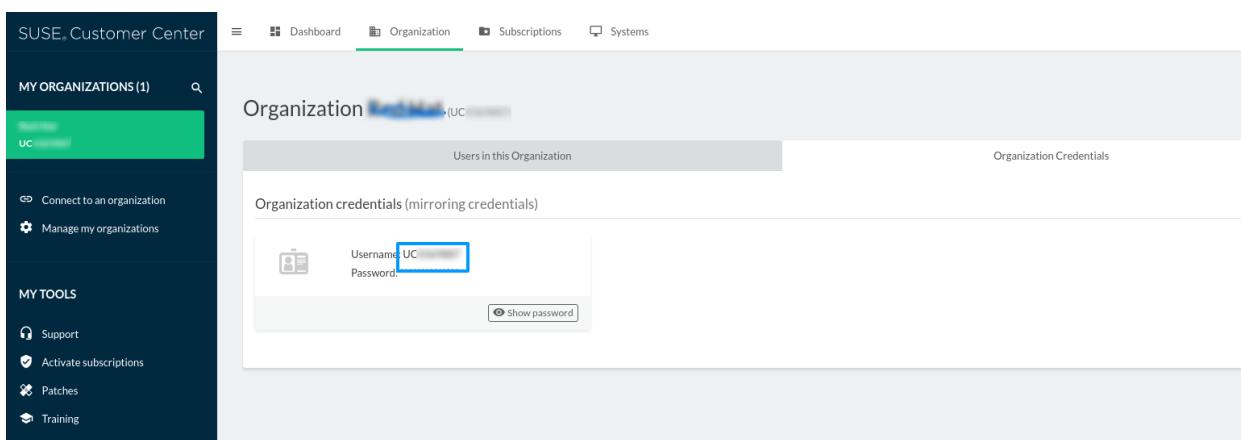
### Set up SUSE account and SMT server

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



- 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/`

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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 and Candlepin and dedicated users owning them
  - prepare Mongo DB with user owning the pulp\_database
  - 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

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_ba \
ckup/candlepin.dump
mongorestore --host mongo.example.com --db pulp_database --username pulp --password <PULP_PASSWORD> /tmp/migration_ba \
ckup/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 <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
```

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.](#)

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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 overridden 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.

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

## 1. Installation

1.1 Katello

1.2 Smart proxy

1.3 Clients

## 2. Upgrade

2.1 Katello

2.2 Smart proxy

2.3 Clients

## 3. Release Notes

## 4. CLI

## 5. User Guide

5.1 Activation Keys

5.2 Backup

5.3 Foreman proxy

5.4 Content

5.5 Content Hosts

5.6 Content Views

5.7 Disconnected

5.8 Docker Management

5.9 Email Notifications

5.10 Errata

5.11 Glossary

5.12 Content Credentials

5.13 Host Collections

5.14 Lifecycle Environments

5.15 Provisioning

5.16 Puppet Integration

5.17 Red Hat Content

5.18 SUSE Content

5.19 Remote Databases

5.20 Content View Import/Export

## 6. Troubleshooting

## 7. Api Documentation

## 8. Annotated Backend Requests

# Katello Troubleshooting

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

## Table of Contents

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

## Sub-services status

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

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

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

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

`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` :

The screenshot shows a browser window with the URL `https://katello.example.com/foreman_tasks/tasks?utf8=%E2%9C%93&search=result+%3D+error`. The page title is "Tasks". A search bar at the top contains the query "result = error". Below the search bar is a table with the following columns: Action, State, Result, Started at, and User. The table lists several tasks, all of which have failed (Result: error). The tasks include publishing content views, synchronizing repositories, and creating/destroying organizations.

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` :

The screenshot shows a browser window with the URL `https://katello.example.com/foreman_tasks/tasks?utf8=%E2%9C%93&search=state+%3D+paused`. The page title is "Tasks". A search bar at the top contains the query "state = paused". Below the search bar is a table with the following columns: Action, State, Result, Started at, and User. The table lists one task, which is currently paused (State: paused) and has failed (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

Below the table, a message says "Displaying 1 entry".

## Dealing with paused task

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

The screenshot shows the Foreman web interface with the URL [https://katello.example.com/foreman\\_tasks/tasks?utf8=%E2%9C%93&search=state+%3D+paused](https://katello.example.com/foreman_tasks/tasks?utf8=%E2%9C%93&search=state+%3D+paused). The task details page is open, showing a task in a 'Paused' state. The action is 'Actions::Pulp::Repository::CopyDistribution' and the input is a JSON object:

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

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

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

The screenshot shows the Dynflow Console with the URL [https://katello.example.com/foreman\\_tasks/dynflow/d0fe4202-87f7-4218-9cbc-cd8bfb25afcd](https://katello.example.com/foreman_tasks/dynflow/d0fe4202-87f7-4218-9cbc-cd8bfb25afcd). The task history is displayed, showing various steps and their execution times. Step 20, 'Actions::Pulp::Repository::CopyDistribution', failed with an error. The task started at 2014-09-12 08:52:23 UTC and ended at 2014-09-12 08:57:14 UTC. The real time taken was 291.56s.

Step	Action	Time
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 ] <a href="#">Skip</a>

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

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

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

## Dealing with Long Running Tasks

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

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

The screenshot shows a Foreman interface with a 'Dynflow Console' tab open. A modal dialog is displayed, showing a task configuration. The 'Action' field contains:

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

The 'Input' field contains:

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

The 'Output' field contains:

```
{"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:

The screenshot shows the Foreman Dynflow Console with a task timeline. The timeline includes steps like 'sequence' and 'concurrence'. A specific step, '20: Actions::Pulp::Repository::CopyDistribution', is highlighted with an orange border and labeled 'suspended'.

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

Plan Run Finalize

sequence concurrence

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

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

## Locking

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

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

The screenshot shows the Foreman interface with the title bar "Publish Content View" and "Tasks" open. The URL in the address bar is [https://katello.example.com/content\\_views/3/publish](https://katello.example.com/content_views/3/publish). The main content area is titled "Content Views". A search bar shows "Showing 1 of 1 (1 Total)". A table lists a single item: "My view" under the "Name" column. Below the table, a red error message box contains the text: "Required lock is already taken by other running tasks. Please inspect their state, fix their errors and resume them. × Required lock: read Conflicts with tasks: - https://katello.example.com/foreman\_tasks/tasks/8cb306e7-91a9-49cc-8172-77fbd152e0c5". At the bottom, there's a section titled "Publish New Version" with a note: "A new version of My view and promoted to the Library environment. It can be promoted to other environments from the Versions tab of this Content View."

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

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

## Debug Certificate

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

### Generating a Debug Certificate

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

## Edit Default Organization

**Name \***

**Label \***

**Description**

**Default System SLA**

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

**Users**

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

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

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

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

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

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

## Using Firefox to browse content

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

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

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

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

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

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

## Frequently Asked Questions

Can I use pulp-admin with Katello?

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

1. Install needed packages

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

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

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

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

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

5. Use pulp-admin by specifying the admin username and password:

```
pulp-admin -u admin -p rGox3G9QhfCRD8fTsNR7FxqdgbvfJfSJ repo list
```

## Using pulp-admin without password

Using the 'pulp-admin login' command does not function and is not supported with Katello in an attempt to limit access to the certificate authority generated at installation time.

Katello 3.0 generates a client cert at installation time which allows usage of pulp-admin without specifying the username and password. To use this:

1. `mkdir ~/.pulp/`
2. Copy the public client cert and private key to a file together:

```
# sudo cat /etc/pki/katello/certs/pulp-client.crt /etc/pki/katello/private/pulp-client.key > ~/.pulp/user-cert.pem
```

3. Run pulp-admin without username and password:

```
pulp-admin repo list
```

## How can I sync a repository like Katello does directly from the console?

Sometimes you want to debug why a synchronization of a repository from Katello is failing and rather than dig through log files and error messages it can often be easier to try to sync the repo with the "grinder" tool which is what Katello uses to download repositories. The tool can be ran from a terminal on your Katello server:

```
$ grinder yum --label=sync-test --url=https://fedorapeople.org/groups/katello/releases/yum/1.0/RHEL/6Server/x86_64/
grinder.RepoFetch: INFO    fetchYumRepo() repo_label = sync-test, repo_url =
https://fedorapeople.org/groups/katello/releases/yum/1.0/RHEL/6Server/x86_64/, basepath = ./, verify_options = {}
grinder.RepoFetch: INFO    sync-test, https://fedorapeople.org/groups/katello/releases/yum/1.0/RHEL/6Server/x86_64/
Calling RepoFetch with: cacert=<None>, clicert=<None>, clikey=<None>, proxy_url=<None>, proxy_port=<3128>, proxy_user=<None>,
proxy_pass=<NOT_LOGGED>, sslverify=<1>, max_speed=<None>, verify_options=<{}>, filter=<None>
...
grinder.ParallelFetch: INFO    5 threads are active. 8 items left to be fetched
grinder.ParallelFetch: INFO    4 threads are active. 4 items left to be fetched
grinder.ParallelFetch: INFO    WorkerThread deleting ActiveObject
grinder.ParallelFetch: INFO    Thread ending
grinder.ParallelFetch: INFO    3 threads are active. 3 items left to be fetched
grinder.ParallelFetch: INFO    WorkerThread deleting ActiveObject
grinder.ParallelFetch: INFO    Thread ending
```

You now have a directory called sync-test off of your current working directory:

```
$ ls sync-test/
converge-ui-devel-0.8.3-1.el6.noarch.rpm
elasticsearch-0.18.4-13.el6.noarch.rpm
katello-1.0.6-1.el6.noarch.rpm
katello-agent-1.0.6-1.el6.noarch.rpm
katello-all-1.0.6-1.el6.noarch.rpm
katello-certs-tools-1.1.7-1.el6.noarch.rpm
lucene3-contrib-3.4.0-2.el6.noarch.rpm
repodata
rubygem-actionmailer-3.0.10-3.el6.noarch.rpm
...
```

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

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

[View our Privacy Policy](#)

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

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

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

Foreman v2

Foreman API v2 is currently the default API version.

## Resources

### Activation keys

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

### Architectures

Resource	Description
<a href="#">GET /api/architectures</a>	List all architectures

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

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

[View our Privacy Policy](#)

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

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

HOME

GET STARTED ▾

GET HELP ▾

GET INVOLVED ▾

NEWS ▾

# Katello 3.11 Documentation

3.11 ▾

- 1. Installation
  - 1.1 Katello
  - 1.2 Smart proxy
  - 1.3 Clients
- 2. Upgrade
  - 2.1 Katello
  - 2.2 Smart proxy
  - 2.3 Clients
- 3. Release Notes
- 4. CLI
- 5. User Guide
  - 5.1 Activation Keys
  - 5.2 Backup
  - 5.3 Foreman proxy
  - 5.4 Content
  - 5.5 Content Hosts
  - 5.6 Content Views
  - 5.7 Disconnected
  - 5.8 Docker Management
  - 5.9 Email Notifications
  - 5.10 Errata
  - 5.11 Glossary
  - 5.12 Content Credentials
  - 5.13 Host Collections
  - 5.14 Lifecycle Environments
  - 5.15 Provisioning
  - 5.16 Puppet Integration
  - 5.17 Red Hat Content
  - 5.18 SUSE Content
  - 5.19 Remote Databases
  - 5.20 Content View Import/Export
- 6. Troubleshooting
- 7. Api Documentation
- 8. Annotated Backend Requests

# 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": "32cee4d1-6e51-419a-9833-5f2c88e3efd9",
        "num_processed": 1
      }
    ],
    "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"
},
```

```
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",
"drpm": "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": [
]
```

```

"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": "fbcc75485-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/831256a124718bf39166b564d8e689954ff0a8f0f479ba24cfa26350109bc5",  
      "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-8841-093d05c8eac3",  
                    "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": "3fc2c927de9e13bf68469032a28b139d3e5ad2e58564fc210fd6e48635be694",  
        "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": "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/"  
    }]
```

```
"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": "251768bdd15f13d78487c27638aa6aec01551e253756093cde1c0ae878a17d2",
"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"  
                ]  
            }  
        }  
    },  
    "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",  
        "pkclist": [  
            {  
                "_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",  
                            "c9f064a6862573fb9f2a6aff7c3621f1940b492df2edfc2ebbd0b8305f51147"  
                        ]  
                    }  
                ]  
            }  
        ]  
    }  
]
```

```
{
  "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 libbz2 library must be restarted for the update 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 relevant to your system have been applied.\n\nThis update is available via the Red Hat Network. Details on how to use the Red Hat Network to apply this update are available at http://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": "izap+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",
  "pkglst": [
  ],
  "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
```

```
{  
    "_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,  
    "username": null,  
    "password": null  
  }  
}
```

```

    "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": "Pushing 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": 1
    }
]
```

```
"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"
}
```

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

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

[View our Privacy Policy](#)

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

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