



Red Hat Subscription Management 1

Using Subscription Asset Manager

Deploying and managing local subscription services

Edition 1.3

Red Hat Subscription Management 1 Using Subscription Asset Manager

Deploying and managing local subscription services

Edition 1.3

Red Hat Subscription Management Documentation Team

rhsm-docs@redhat.com

Legal Notice

Copyright © 2016 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

A guide to installing and using Red Hat Subscription Asset Manager.

Table of Contents

PREFACE	4
CHAPTER 1. INTRODUCTION TO ON-PREMISE SUBSCRIPTION MANAGEMENT	5
1.1. DEFINING SUBSCRIPTION ASSET MANAGER	5
1.2. WORKFLOWS AND USE CASES	6
1.2.1. Direct Subscription Assignments	6
1.2.1.1. The Environment: Small Businesses to Large Enterprises for Locally-Defined Structure	7
1.2.1.2. Workflow	7
1.2.1.3. Details and Options	9
1.2.2. Activation Keys	10
1.2.2.1. The Environment: Preconfigured Systems	10
1.2.2.2. Workflow	10
1.2.2.3. Details and Options	12
CHAPTER 2. INSTALLING SUBSCRIPTION ASSET MANAGER	14
2.1. PREREQUISITES	14
2.2. BASIC INSTALLATION AND SETUP FOR SUBSCRIPTION ASSET MANAGER	14
2.2.1. Installing Using yum	14
2.2.1.1. Configuring Red Hat SAM Manually with an HTTP Proxy	15
2.2.2. Installing Through an ISO Image	16
2.3. INSTALLING AND CONFIGURING ENHANCED REPORTING	17
2.4. ADDITIONAL EXAMPLES OF THE CONFIGURATION SCRIPT	19
2.5. SUBSCRIPTION ASSET MANAGER INSTALLATION LOGS	22
2.6. UPGRADING SUBSCRIPTION ASSET MANAGER	22
2.7. LOGGING INTO THE SUBSCRIPTION ASSET MANAGER WEB UI	24
2.7.1. Supported Browsers	24
2.7.2. The Web UI URL	24
2.7.3. The Default User	25
CHAPTER 3. MANAGING USERS AND ACCESS CONTROLS	26
3.1. ABOUT USERS, ROLES, AND ACCESS CONTROLS	26
3.2. MANAGING USER ACCOUNTS	27
3.2.1. Adding Users	27
3.2.2. Changing a Password	28
3.3. CREATING A ROLE	29
3.4. SETTING UP ACCESS CONTROLS	30
3.5. MANAGING USERS WITH LDAP AUTHENTICATION	34
3.5.1. Supported LDAP Service Types	34
3.5.2. Using LDAP Authentication	34
3.5.2.1. About LDAP Users in Subscription Asset Manager	34
3.5.2.2. Enabling LDAP Authentication	35
3.5.3. Using LDAP Group and Role Mappings	37
3.5.3.1. About LDAP Groups and Subscription Asset Manager Roles	37
3.5.3.2. Enabling LDAP Group-Role Mapping	38
CHAPTER 4. SETTING UP ORGANIZATIONS AND DISTRIBUTORS	40
4.1. ABOUT THE STRUCTURE OF ORGANIZATIONS AND DISTRIBUTORS	40
4.2. MANAGING ORGANIZATIONS	42
4.2.1. Creating a New Organization	42
4.2.2. Removing an Organization	44
4.2.3. Setting the Default Organization	45
4.2.4. Setting Autoattach Preferences	45

4.3. MANAGING DISTRIBUTORS (TECH PREVIEW)	46
4.4. IMPORTING AND MAINTAINING MANIFESTS	47
4.4.1. About Manifests	47
4.4.2. Managing Manifests for Organizations	49
4.4.2.1. Attaching Subscriptions to Organizations	49
4.4.2.2. Downloading the Manifest	52
4.4.2.3. Uploading a Subscription Manifest	52
4.4.2.4. Refreshing Manifests for Organizations	53
4.4.2.5. Viewing the Manifest Import History	56
4.4.3. Managing Manifests for Distributors (TECH PREVIEW)	57
CHAPTER 5. MANAGING SYSTEMS AND SUBSCRIPTIONS	60
5.1. ABOUT SUBSCRIPTIONS ON SYSTEMS	60
5.1.1. About Relationships Between Subscriptions and Systems	60
5.1.1.1. Interactions with Subscriptions, Products, and Systems	60
5.1.1.2. Counting Subscriptions	61
5.2. REGISTERING A SYSTEM	62
5.3. USING SYSTEM GROUPS	62
5.4. VIEWING SUBSCRIPTIONS FOR THE ORGANIZATION	63
5.5. VIEWING INSTALLED PRODUCTS FOR A SYSTEM	64
5.6. MANUALLY MANAGING SUBSCRIPTIONS	65
5.6.1. Attaching Subscriptions to a System	65
5.6.2. Removing Subscriptions	66
5.7. CONFIGURING AUTOATTACH PREFERENCES FOR A SYSTEM	66
5.8. RUNNING AUTOATTACH OPERATIONS MANUALLY	67
5.8.1. Running Autoattach on All Systems	68
5.8.2. Running Autoattach on a Single System	68
CHAPTER 6. USING ACTIVATION KEYS	69
6.1. ABOUT ACTIVATION KEYS	69
6.2. CREATING AN ACTIVATION KEY	69
6.3. ATTACHING AND UPDATING SUBSCRIPTIONS FOR AN ACTIVATION KEY	70
6.4. ASSIGNING SYSTEM GROUPS TO A KEY	71
6.5. REGISTERING A SYSTEM USING AN ACTIVATION KEY	72
6.5.1. Using Activation Keys from the GUI	72
6.5.2. Using Activation Keys from the Command Line	74
CHAPTER 7. MANAGING VIRTUAL HOSTS AND GUESTS	75
7.1. SUPPORTED HYPERVISORS	75
7.2. ABOUT HOST/GUEST ASSOCIATIONS	75
7.3. SETTING UP A RHEV (KVM) OR XEN HYPERVISOR	76
7.4. SETTING UP A VMWARE HYPERVISOR	76
7.5. REGISTERING GUEST INSTANCES	78
7.6. CREATING A DATA CENTER	78
7.7. REMOVING AND RESTORING HOST AND GUEST ACCOUNTS	79
7.7.1. Removing a Guest Entry	79
7.7.2. Removing a Hypervisor Entry	79
7.7.3. Reregistering a Hypervisor	80
CHAPTER 8. VIEWING SYSTEM-LEVEL SUBSCRIPTION INFORMATION	81
8.1. VIEWING THE HIGH-LEVEL INFORMATION IN THE DASHBOARD	81
8.2. VIEWING SERVER NOTIFICATIONS	82
8.3. VIEWING SYSTEM ADMINISTRATIVE NOTIFICATIONS	84
8.4. CHECKING INDIVIDUAL SYSTEM STATUS	85

8.5. REMEDIATING PROBLEM SUBSCRIPTION STATUS	87
8.6. CREATING SUBSCRIPTION ASSET MANAGER USAGE REPORTS	87
8.6.1. Prerequisites	87
8.6.2. Setting up Reporting After Installation	88
8.6.3. Creating Report Filters	88
8.6.4. Running Reports	90
8.6.5. Subscription Asset Manager Reports Results and Data	91
8.6.6. Enhanced Reporting Logs	93
8.7. USING SUBSCRIPTION ASSET MANAGER TO GENERATE SATELLITE USAGE REPORTS	93
8.7.1. About Satellite Consolidated Reports	93
8.7.1.1. The Advantages of Enhanced Reporting	93
8.7.1.2. Differences in Subscription Statuses from Satellite	94
8.7.1.3. Syncing Data from Satellite 5.6 to Subscription Asset Manager	95
8.7.1.4. Users in Satellite 5.6 and Subscription Asset Manager	96
8.7.2. Prerequisites	96
8.7.3. Configuring Reporting	97
8.7.4. Running Reports and Getting Results	103
8.7.4.1. Creating Report Filters	104
8.7.4.2. Running Reports	105
8.7.4.3. Subscription Asset Manager Reports Results and Data	106
8.7.5. Troubleshooting Enhanced Reports	108
8.7.5.1. Enhanced Reporting Logs	108
8.7.5.2. Common Problems	108
8.7.5.3. Other Known Issues	109
CHAPTER 9. MANAGING SUBSCRIPTION ASSET MANAGER INSTANCES	111
9.1. BACKING UP SUBSCRIPTION ASSET MANAGER	111
9.2. RESTORING SUBSCRIPTION ASSET MANAGER	112
9.3. REGENERATING AND REPLACING CA CERTIFICATES	114
9.4. SETTING UP SUBSCRIPTION ASSET MANAGER ON A MULTIHOMED SYSTEM	115
9.5. SUBSCRIPTION ASSET MANAGER LOG AND FILE LOCATIONS	119
CHAPTER 10. RED HAT ACCESS PLUG-IN	121
10.1. RED HAT ACCESS PLUG-IN	121
10.1.1. Searching for Solutions in the Red Hat Access Plug-in	121
10.1.2. Viewing Existing Support Cases Using the Red Hat Access Plug-in	121
10.1.3. Modifying Existing Support Cases Using the Red Hat Access Plug-in	122
10.1.4. Creating New Support Cases Using the Red Hat Access Plug-in	122
APPENDIX A. REVISION HISTORY	124

PREFACE

Red Hat Subscription Asset Manager provides local management over subscriptions that are assigned to systems within an infrastructure.

This guide is intended for Linux administrators who need to have clear reporting structures and control over software subscriptions. This guide covers the concepts and structure of Red Hat subscription tools, procedures for installing and configuring Subscription Asset Manager, and information on how to assign and update subscriptions for systems.

CHAPTER 1. INTRODUCTION TO ON-PREMISE SUBSCRIPTION MANAGEMENT

A subscription management application maintains a list of all subscriptions available to an organization and a list of all systems within that organization. More important, a subscription management application maps what subscriptions are attached to what systems.

IT hardware needs to be managed and clearly inventoried, and the software installed on those machines also needs to be managed and clearly inventoried. An *inventory* is simply a means to track *what* software is installed and *where* it is installed and *how many* copies are actively being used.

IT administrators face increasing pressure to have an accurate accounting of the software, not just from governmental regulations like Sarbanes-Oxley in the United States, but also to achieve critical industry certifications, such as Payment Card Industry Data Security Standard (PCI-DSS) or SAS-70. Generally, this accounting of software assets is called *software license management*; with Red Hat's subscription model, this is *subscription management*.

Subscription management establishes the *relationship* between the product subscriptions that are available and the elements of the IT infrastructure where those subscriptions are allocated.

With Red Hat's commitment to free and open software, subscription management is focused on delivering tools that help IT administrators monitor their software/systems inventory for their own benefit. Subscription management *does not* enforce or restrict access to products.

1.1. DEFINING SUBSCRIPTION ASSET MANAGER

The simplest model for assigning subscriptions and delivering content is for local systems to connect directly to Red Hat's hosted subscription and content network. However, for large environments, highly-secure environments, and many other situations, that hosted arrangement is not feasible.

For those infrastructures, it is possible to allocate a subset of the account subscriptions to an on-premise application. The on-premise application manages subscription and system inventories locally. This has performance benefits by lowering bandwidth, and it offers significant management benefits to administrators by allowing local and flexible control over subscription management.

Subscription Asset Manager is that on-premise subscription management application. It performs two backend management functions:

- Allocate subscriptions as a subscription service
- Work as a real-time proxy for Red Hat's content delivery network

As a subscription service, Subscription Asset Manager handles the system registration (verifying that the system is allowed to access the content). It also supplies the system with information on what products are available and handles a central list of subscriptions and remaining quantities for the entire account.

Subscription Asset Manager works as a proxy server for the local systems to connect to Red Hat's hosted content delivery network. Ultimately, Red Hat's hosted services are still responsible for delivering the content to the system when requested.

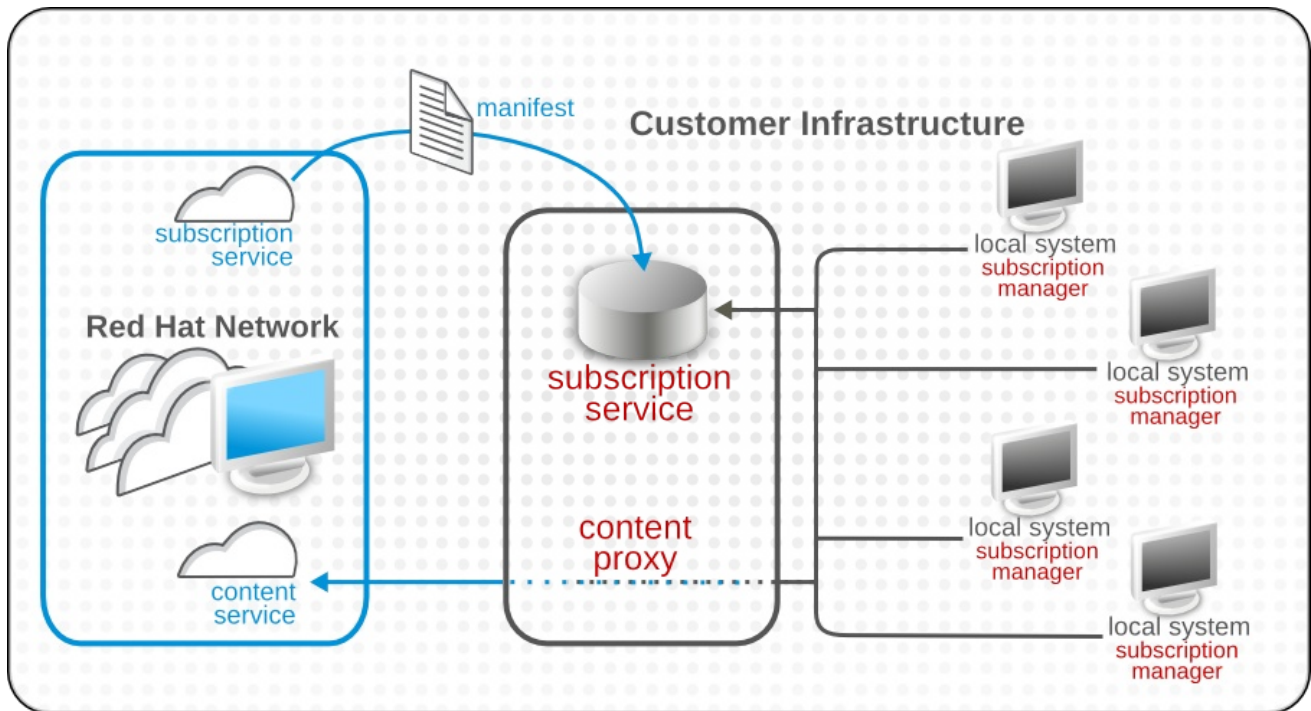


Figure 1.1. Hosted Content Delivery and On-Premise Subscription Services

Administrators may want to exert some control locally over subscription services or content delivery or both. Because each component in the subscription management framework is an independent application, with independent client configuration, different sources can be used.

Using subscription management applications offers administrators control over configuration and the ability to improve performance based on network conditions or physical locations of machines.

All Red Hat Enterprise Linux subscriptions automatically include some tools for managing the subscription configuration:

- Red Hat Subscription Manager client tools to manage local systems
- Customer Portal Subscription Management to manage systems globally through the Customer Portal
- Subscription Asset Manager to manage systems locally

1.2. WORKFLOWS AND USE CASES

Subscription Asset Manager provides different avenues to manage how subscriptions are attached to systems, which provides options in how systems can be provisioned and updated.

1.2.1. Direct Subscription Assignments

The Red Hat hosted services have a flat, undefined structure where all systems are in a single pool. While convenient in some ways, this loses the structure that real IT environments have, where systems are loosely or strictly associated with different organizational divisions, physical locations, and content streams. Subscription Asset Manager introduces an organizational structure, where administrators can create local organizations (for subscriptions) and environments (for content streams) that allow them to arrange systems in a way that reflects real-life configurations.

1.2.1.1. The Environment: Small Businesses to Large Enterprises for Locally-Defined Structure

Subscription Asset Manager imposes a structure on the subscription service and content service that allows administrators to design those systems in a way that reflects the actual infrastructure configuration.

Subscriptions are grouped by organizations, which are top-level divisions. Each organization represents a separate subscription application entry to Customer Portal Subscription Management.

Within each organization, it is possible to create system groups which help structure systems for updates, access control, and content management.

Imposing an organizational structure on the subscription inventory gives Subscription Asset Manager the ability to better present information about the subscriptions and systems for an account. For large enterprises, this may be the only functionality from Subscription Asset Manager that is required, using Subscription Asset Manager for reporting and auditing while custom or enterprise-level applications are used to manage systems.

For small and medium business, Subscription Asset Manager offers more control over system management than is possible through hosted services alone:

- Enact security rules that require on-premise services rather than hosted services.
- Better manage virtual environments, particularly in private clouds or data centers, which require system to be created and removed on the fly.
- Define different content repositories for different types of systems, such as different sources for development and production systems.

The functionality of Subscription Asset Manager tracks closely with the functionality of Customer Portal Subscription Management and Red Hat Subscription Manager. Systems can be registered and have subscriptions automatically attached. Preferences like release versions and service levels can be used to govern software updates.

1.2.1.2. Workflow

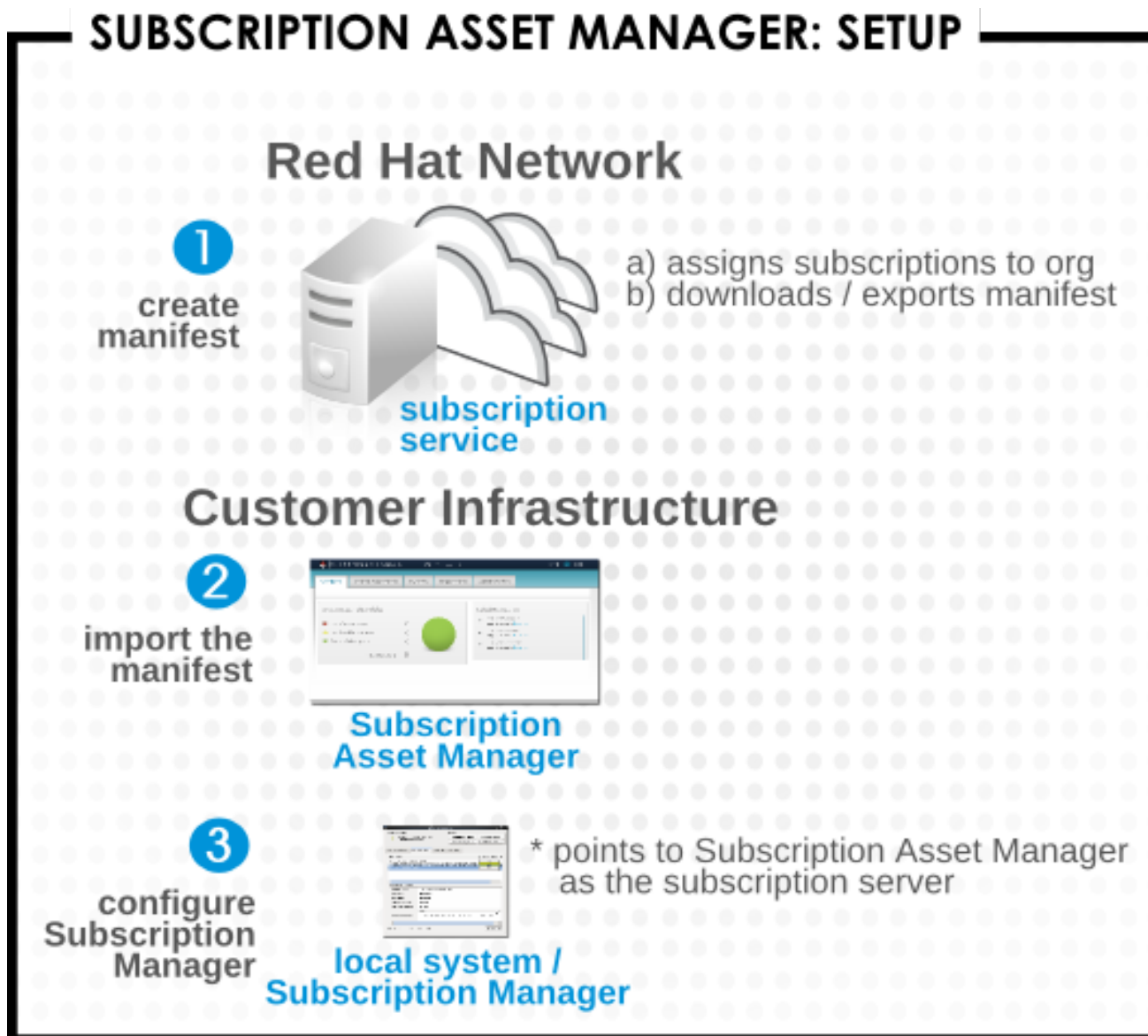


Figure 1.2. Subscription Asset Manager Setup

1. If necessary, create an entry in the Red Hat inventory for the organization ([Section 4.2.1, “Creating a New Organization”](#)). Every organization in Subscription Asset Manager must have a corresponding subscription service entry in the Red Hat inventory.
2. Assign a bloc of subscriptions to the organization ([Section 4.4.2.1, “Attaching Subscriptions to Organizations”](#)). This bloc of subscriptions is the *manifest* of subscriptions for that Subscription Asset Manager organization.
3. Export the manifest ([Section 4.4.2.2, “Downloading the Manifest”](#)).
4. Import the manifest into Subscription Asset Manager ([Section 4.4.2.3, “Uploading a Subscription Manifest”](#)).
5. Configure the Red Hat Subscription Manager client on the local system to use the Subscription Asset Manager subscription service and, optionally, the Subscription Asset Manager content proxy ([Section 5.2, “Registering a System”](#)).

The backend configuration for Subscription Asset Manager and its manifests only needs to be done once, and the Red Hat Subscription Manager configuration changes only need to be made once per system.

When that is complete, then the system can be registered and have subscriptions attached to it.

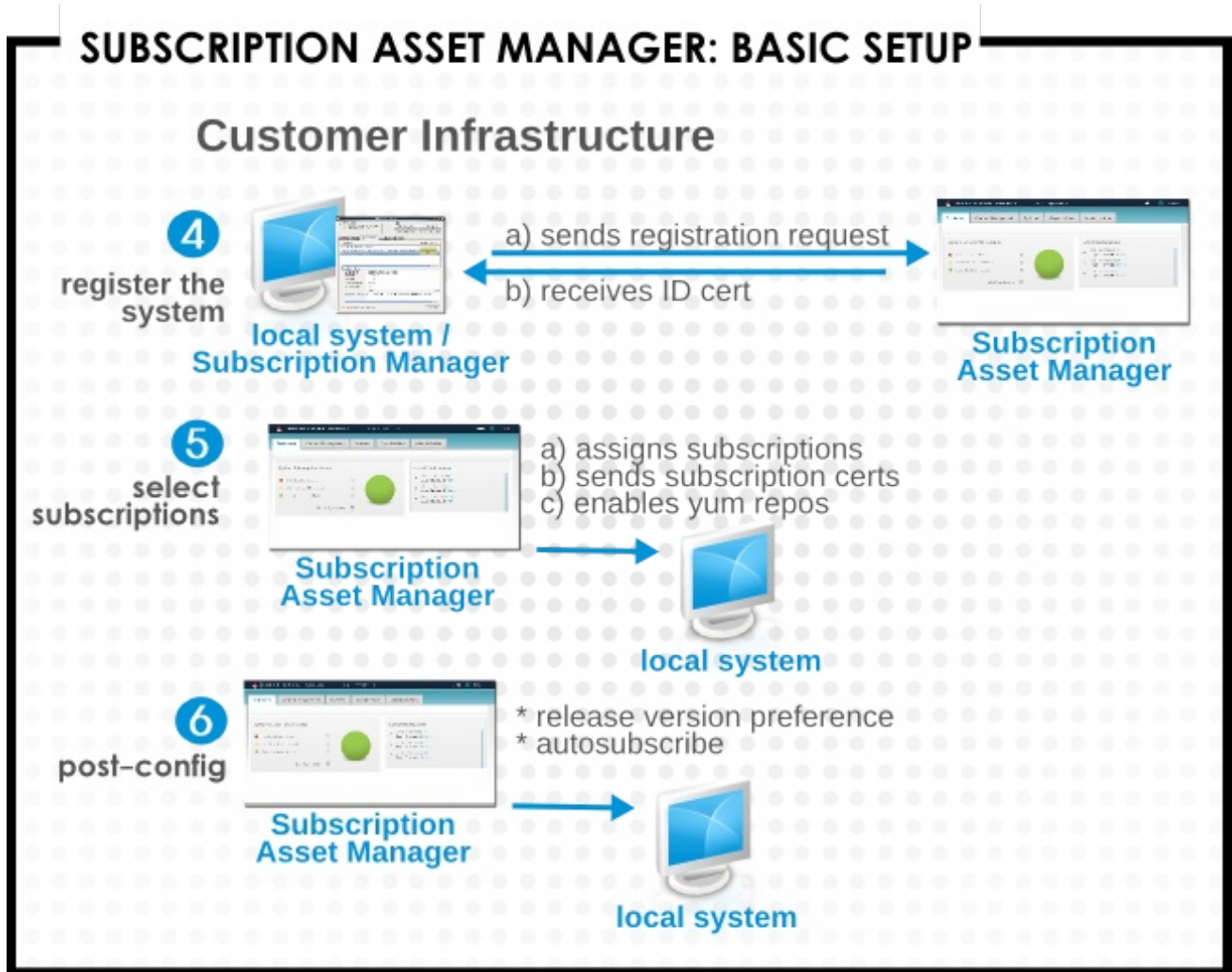


Figure 1.3. Registering with Subscription Asset Manager

1. Register the system ([Section 5.2, “Registering a System”](#)).

Using the **subscription-manager** CLI command, use the **register** command with the username and password for the Customer Portal Subscription Management account holder and the hostname of the Subscription Asset Manager server.

For the Red Hat Subscription Manager UI, autoattaching subscriptions is performed by default. Check the option to attach subscriptions later.

2. Select and attach the subscriptions, using the Subscription Asset Manager UI ([Section 5.6.1, “Attaching Subscriptions to a System”](#)).

1.2.1.3. Details and Options

After registering a system with Subscription Asset Manager, it can be managed using either Subscription Asset Manager or Red Hat Subscription Manager, but the options set in Red Hat Subscription Manager must match what is available in Subscription Asset Manager.

- Enable autoattaching for the system and, optionally, set a service level preference ([Section 5.7, “Configuring Autoattach Preferences for a System”](#)).

1.2.2. Activation Keys

Activation keys are a way to preconfigure subscriptions for a system before it is registered (or, when used with provisioning systems like kickstart, before the system is even created). This grants a great deal of flexibility and control to administrators over which subscriptions to attach to new systems, while simplifying the registration process for users.

1.2.2.1. The Environment: Preconfigured Systems

[Section 1.2.1, “Direct Subscription Assignments”](#) summarizes the benefits of using Subscription Asset Manager in general, and those same benefits apply, generally, to using Subscription Asset Manager to create activation keys.

In addition to the organization framework that Subscription Asset Manager can create using Subscription Asset Manager to create activation keys makes the overall registration and subscription process simpler, almost transparent, for users. They can pass the activation key to register the system and have all of the configured subscriptions apply *and the user never has to know what those subscriptions are or what quantities are required*.

Preconfiguring systems allows administrators to define system configuration consistently. For example, using activation keys and pre-configured subscriptions is common when supplying company-configured laptops or workstations to employees.

Activation keys create a set of subscriptions that can be attached to a system, without attaching them directly to a system. The activation key can be associated with a system group, within the Subscription Asset Manager configuration. From there, it can be used with any system. This results in several benefits for both users and administrators:

- Administrators have control over which subscriptions are installed to a system without having to create and configure every system first.
- Because activation keys are created within Subscription Asset Manager and do not rely on system settings or architecture, the target system does not have to exist yet.
- Users can register their system in a single step and automatically have all the proper subscriptions attached, without having to select and attach subscriptions manually and potentially miss a subscription.

1.2.2.2. Workflow

By default, systems are configured to use Red Hat hosted services. Before activation keys can be created and used, the appropriate backend infrastructure needs to be set up.

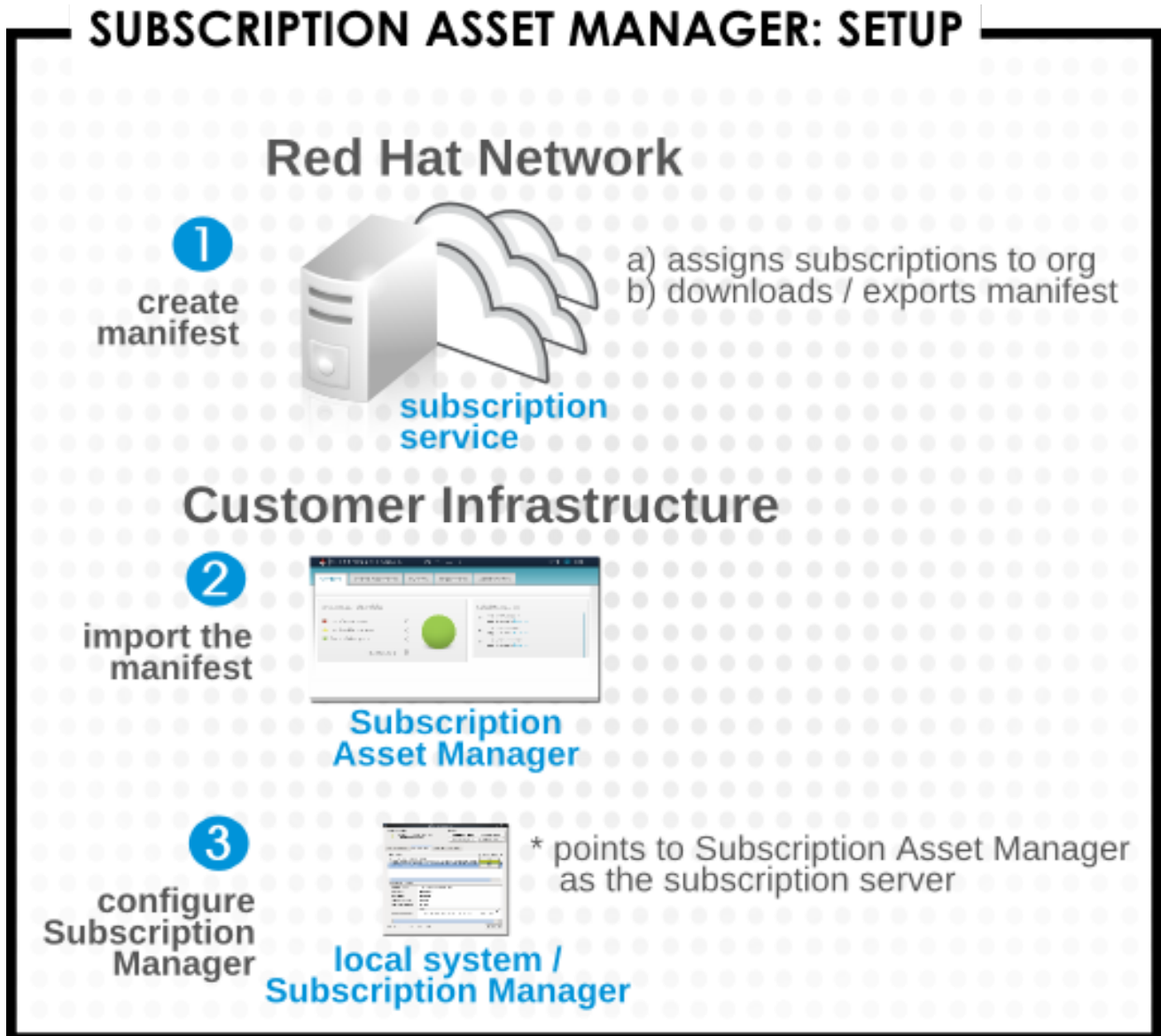


Figure 1.4. Setting up SAM

1. If necessary, create an entry in the Red Hat inventory for the organization ([Section 4.2.1, “Creating a New Organization”](#)). Every organization in Subscription Asset Manager must have a corresponding subscription service entry in the Red Hat inventory.
2. Assign a bloc of subscriptions to the organization ([Section 4.4.2.1, “Attaching Subscriptions to Organizations”](#)).

This bloc of subscriptions is the *manifest* of subscriptions for that Subscription Asset Manager organization.

3. Export the manifest ([Section 4.4.2.2, “Downloading the Manifest”](#)).
4. Import the manifest into Subscription Asset Manager ([Section 4.4.2.3, “Uploading a Subscription Manifest”](#)).
5. Configure the Red Hat Subscription Manager client on the local system to use the Subscription Asset Manager subscription service and, optionally, the Subscription Asset Manager content proxy ([Section 5.2, “Registering a System”](#)).

This can be done at any point before the system is registered, so it can even be performed after the activation key is created.

The backend configuration for Subscription Asset Manager and its manifests only needs to be done once, and the Red Hat Subscription Manager configuration changes only need to be made once per system.

When that is complete, then activation keys can be made.

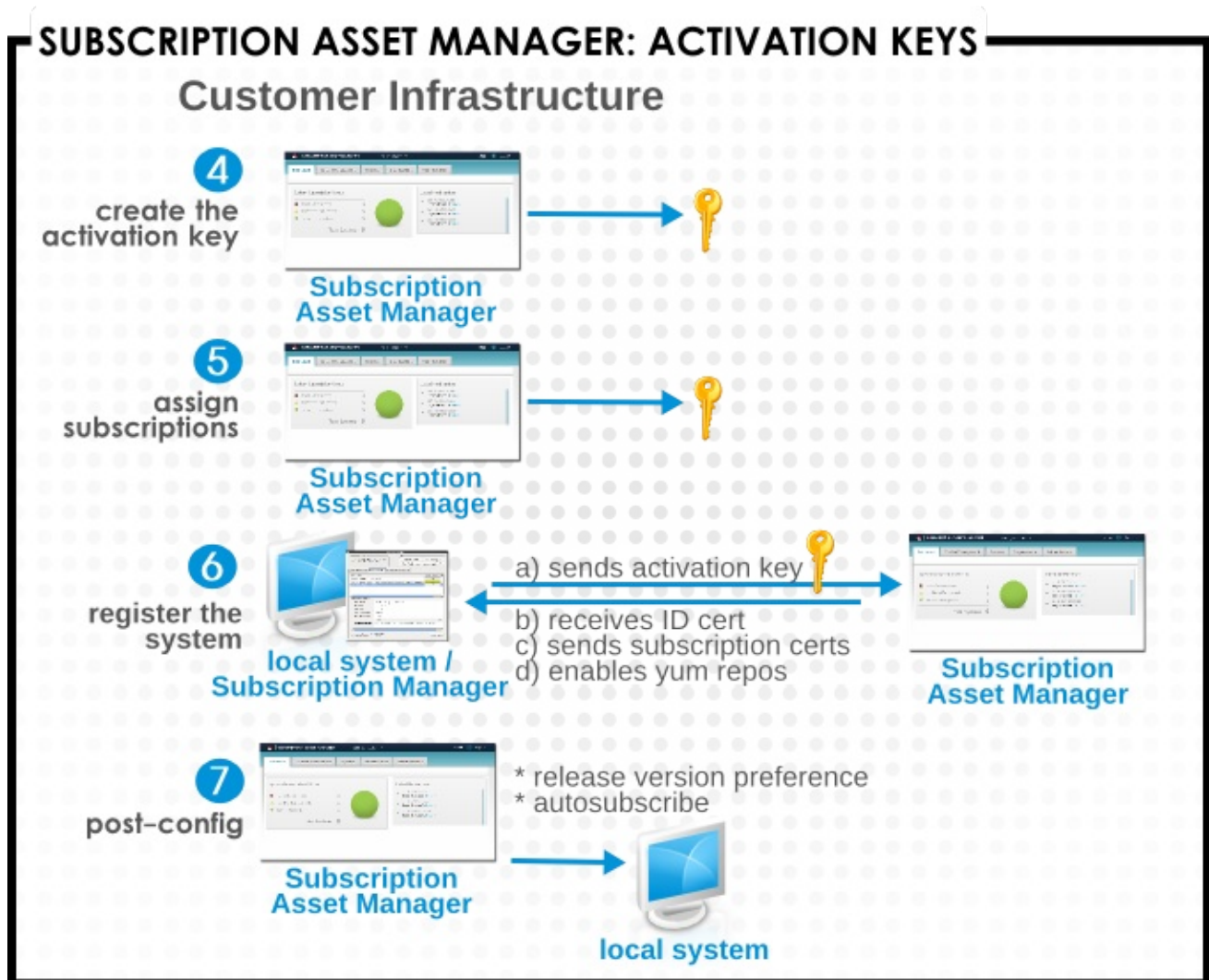


Figure 1.5. Registering with Activation Keys

1. Create the activation key ([Section 6.2, “Creating an Activation Key”](#)). This is a container entry that subscriptions can be attached to.
2. Attach subscriptions to the key ([Section 6.3, “Attaching and Updating Subscriptions for an Activation Key”](#)).
3. Register the local system using the activation key ([Section 6.5, “Registering a System Using an Activation Key”](#)).

This is basically an autoattach operation, only instead of using the Red Hat Subscription Manager evaluation to select best-matched subscriptions, it attaches the pre-configured subscriptions associated with the key.

1.2.2.3. Details and Options

After registering a system with Subscription Asset Manager, it can be managed using either Subscription Asset Manager or Red Hat Subscription Manager, but the options set in Red Hat Subscription Manager must match what is available in Subscription Asset Manager.

- Enable autoattaching for the system and, optionally, set a service level preference ([Section 5.7](#), “[Configuring Autoattach Preferences for a System](#)”).

CHAPTER 2. INSTALLING SUBSCRIPTION ASSET MANAGER

Subscription Asset Manager is included as part of a subscription to Red Hat Enterprise Linux. Subscription Asset Manager lets you administer subscriptions locally and lets you create a stronger organization structure to define machines, groups, and content streams.

Customize Subscription Asset Manager by using a setup script with simple defaults for all settings, and also lets you do environment-specific customization.

2.1. PREREQUISITES

The machine on which you install Subscription Asset Manager 1.4 must meet these requirements:

- Red Hat Enterprise Linux 6.6 or higher Server, 64-bit.
- The **rhel-6-server-sam-rpms** repository must be enabled, and the **rhel-server-rhsc1-6-eus-rpms** repository must be disabled.
- OpenJDK 1.6.
- A minimum of 1.5GB RAM.
- A minimum of 1GB of memory must be swap-enabled.
- The Subscription Asset Manager hostname must be fully resolvable in DNS. Both servers and any client systems must be able to resolve the Subscription Asset Manager hostname for authentication operations and other management tasks.
- Ports 443 and 8088 for HTTPS (secure HTTP) must be open.
- *For enhanced reporting.* An additional 4 GB disk space.

2.2. BASIC INSTALLATION AND SETUP FOR SUBSCRIPTION ASSET MANAGER

You can install Subscription Asset Manager by using the Red Hat hosted repositories and **yum**, or through an ISO image. The installation paths differ slightly, depending on the network and infrastructure.

2.2.1. Installing Using yum

1. Register the host system. Use the **--auto-attach** option to attach the required subscriptions for the operating system immediately.

```
[root@server ~]# subscription-manager register --auto-attach
Username: jsmith@example.com
Password:
```

2. Wait several minutes for the updated content repositories to be added to the system configuration.
3. Enable the **rhel-6-server-sam-rpms** repository.

If the Enhanced Updates (EUS) repository is enabled, then disable it, either as part of the **yum** configuration (as in this example) or when running **yum** to install packages. There are conflicts in the Ruby packages between the EUS repository and the Subscription Asset Manager repository.

```
[root@server ~]# subscription-manager repos --enable rhel-6-server-
sam-rpms --disable rhel-server-rhsc1-6-eus-rpms
Loaded plugins: product-id, refresh-packagekit
===== repo: rhel-6-server-sam-rpms
=====
[rhel-6-server-sam-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
https://cdn.redhat.com/content/dist/rhel/server/6/6Server/x86_64/sub
scription-asset-manager/1/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-6-server-sam-rpms
cost = 1000
enabled = 1
enablegroups = True
exclude =
failovermethod = priority
...
```

4. Install the **katello-headpin-all** package using **yum install**:

```
[root@server ~]# yum install -y katello-headpin-all
```

5. After installing the packages, run the Subscription Asset Manager configuration script, **katello-configure**.

The required parameters for the basic installation are the deployment type, which is **sam**, and an administrator password.

While not required, it is useful to specify an initial organization name for your deployment. If this is not set, then the first organization has a default name of *ACME_Corporation*.

This sets up the Subscription Asset Manager instance with the default user and database settings.

```
[root@server ~]# katello-configure --deployment=sam --
org=Example_Org --user-pass=admin
Starting Katello configuration
The top-level log file is [/var/log/katello/katello-configure-
20130904-210539/main.log]
```

2.2.1.1. Configuring Red Hat SAM Manually with an HTTP Proxy

For networks that go through an HTTP Proxy, use the following **katello-installer** options so that the Satellite Server completes the configuration successfully:

```
katello-installer --katello-proxy-url=http://myproxy.example.com --
katello-proxy-port=8080 --katello-proxy-username=proxy_username --katello-
proxy-password=proxy_password
```

Where:

- **--katello-proxy-url**: URL of the HTTP proxy server
- **--katello-proxy-port**: Port the HTTP proxy server is listening on
- **--katello-proxy-username**: (Optional) HTTP proxy username for authentication. If your HTTP proxy server does not require a username, you do not have to specify the username.
- **--katello-proxy-password**: (Optional) HTTP proxy password for authentication. If your HTTP proxy server does not require a password, you do not have to specify the password.

After configuring SAM to go through the HTTP Proxy, you must ensure that **yum** or **subscription-manager** can connect to the Red Hat Content Delivery Network (CDN) and that SAM can synchronize its repositories to the CDN. Follow these steps:

Procedure 2.1. Configuring SAM to Allow Red Hat Subscription Manager Access to the CDN

1. On the network gateway and the HTTP Proxy, open the following hostnames, ports and protocols:

Table 2.1. Required Hostnames, Ports and Protocols

Hostname	Port	Protocol
subscription.rhn.redhat.com	443	https
cdn.redhat.com	443	https
*.akamaiedge.net	443	https

2. In the Satellite Server, complete the following details in the file:

/etc/rhsm/rhsm.conf:

```
# an http proxy server to use (enter server FQDN)
proxy_hostname = http_proxy.example.com

# port for http proxy server
proxy_port = 3128

# user name for authenticating to an http proxy, if needed
proxy_user =

# password for basic http proxy auth, if needed
proxy_password =
```

2.2.2. Installing Through an ISO Image

Prerequisite: ISO installations require imported Red Hat GPG keys before installation. Run the following command as root before running the installation script:

```
# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

The following procedure details how to install SAM on a host through ISO.

1. Download the ISO image from the Red Hat SAM repository in the Red Hat Customer Portal.
2. As the root user, mount the ISO image to a directory:

```
# mkdir /media/iso
# mount -o loop [iso_filename] /media/iso
```

3. Change directory to **/media/iso**.
4. Run the installer script in the mounted directory:

```
# ./install_packages
```

SAM is now installed on your host system.

2.3. INSTALLING AND CONFIGURING ENHANCED REPORTING

Subscription Asset Manager includes an optional reporting module which can pull in cross-organization (within Subscription Asset Manager) and even cross-subscription service (Subscription Asset Manager and Satellite 5.6) data to generate reports about subscription status on systems.

The reporting module requires additional packages to create a reporting database and a sync service.

When using enhanced reporting, there are some additional system requirements:

- The **crond** service must be running.
- An additional 4 GB of disk space must be available for the reporting database journal.
- Additional packages
 - `splice`
 - `ruby193-rubygem-splice_reports`
 - `spacewalk-splice-tool`



IMPORTANT

If enhanced reporting will be used with a Satellite 5.6 instance, the Subscription Asset Manager instance must be dedicated only as a reporting server for that Satellite 5.6 server. It cannot be used to manage other systems.

To install Subscription Asset Manager with enhanced reporting configured:

1. Register the host system. Use the **--auto-attach** option to attach the required subscriptions for the operating system immediately.

```
[root@server ~]# subscription-manager register --auto-attach
Username: jsmith@example.com
Password:
```

- Wait several minutes for the updated content repositories to be added to the system configuration.
- Enable the **rhel-6-server-sam-rpms** repository.

If the Enhanced Updates (EUS) repository is enabled, then disable it, either as part of the **yum** configuration (as in this example) or when running **yum** to install packages. There are conflicts in the Ruby packages between the EUS repository and the Subscription Asset Manager repository.

```
[root@server ~]# subscription-manager repos --enable rhel-6-server-sam-rpms --disable rhel-server-rhsc1-6-eus-rpms
Loaded plugins: product-id, refresh-packagekit
===== repo: rhel-6-server-sam-rpms
=====
[rhel-6-server-sam-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
https://cdn.redhat.com/content/dist/rhel/server/6/6Server/x86_64/subscription-asset-manager/1/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-6-server-sam-rpms
cost = 1000
enabled = 1
enablegroups = True
exclude =
failovermethod = priority
...
```

- Install Subscription Asset Manager and reporting packages:

```
[root@server ~]# yum install -y katello-headpin-all splice ruby193-rubygem-splice_reports spacewalk-splice-tool
```

- After installing the packages, run the Subscription Asset Manager configuration script, **katello-configure**.

The required parameters for the basic installation are the deployment type, which is **sam**, and an administrator password.

While not required, it is useful to specify an initial organization name for your deployment. If this is not set, then the first organization has a default name of *ACME_Corporation*.

This sets up the Subscription Asset Manager instance with the default user and database settings.

```
[root@server ~]# katello-configure --deployment=sam --org=Example_Org --user-pass=admin
Starting Katello configuration
```

The top-level log file is [/var/log/katello/katello-configure-20130904-210539/main.log]

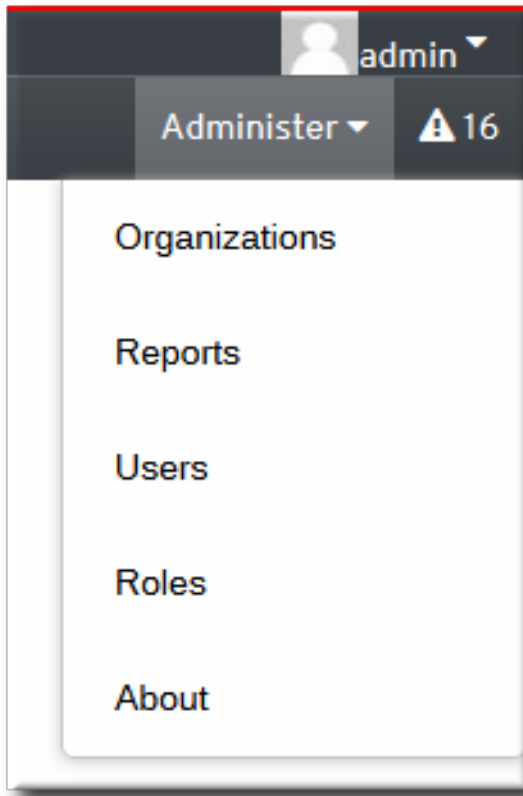


Figure 2.1. The Reports Menu Item

Using reports with Subscription Asset Manager is covered in [Section 8.6, “Creating Subscription Asset Manager Usage Reports”](#). Using Subscription Asset Manager as a reporting server for Satellite 5.6 is covered in [Section 8.7, “Using Subscription Asset Manager to Generate Satellite Usage Reports”](#).

2.4. ADDITIONAL EXAMPLES OF THE CONFIGURATION SCRIPT

Subscription Asset Manager is configured automatically using the **katello-configure** script. This creates the associated subscription service databases, a default administrative user, and default server settings. Any of these deployment settings can be altered by invoking the appropriate arguments in the configuration script.

All of the default values for the configuration are defined in a configuration file, **/usr/share/katello/install/default-answer-file**. All of the attributes in the file can be passed when the **katello-configure** script to allow more relevant values to be set.



NOTE

The complete list of parameters for the configuration script are covered in the **katello-configure** help output and man page.

These examples show some common areas that administrators may want to define. It is also possible to set up proxy servers to use with Subscription Asset Manager, change the Subscription Asset Manager server database information, and the subscription database information.

- [Example 2.1, “Setting the Org and Deployment Type”](#)

- [Example 2.2, “Creating an Admin User”](#)
- [Example 2.3, “Setting up LDAP Authentication”](#)
- [Example 2.4, “Using an Answer File”](#)

Example 2.1. Setting the Org and Deployment Type

There are two parameters that are required from a logical perspective, even if they have functional default values. These are the *deployment type* and the *organization name*.

Subscription Asset Manager is comprised of different components, including a subscription service (candlepin), a web UI (headpin), and APIs that allow communication between components (katello). Additionally, Subscription Asset Manager is itself a component of Satellite 6. Therefore, the configuration script can configure individual parts of Subscription Asset Manager or a full Subscription Asset Manager server.

In real-world deployments, only the **sam** option should ever be used, but this option must be explicitly stated (otherwise, it is set to **katello**).

```
[root@server ~]# katello-configure --deployment=sam --user-pass=admin
```

Additionally, the initial Subscription Asset Manager configuration must define an organization; one organization is always required. The organization is an artificial construct that allows administrators to order and categorize systems within their infrastructure according to environment and content streams. An organization can be named anything, but it is generally useful to give meaningful names to organizations. The default organization name is **ACME_Corporation**; using the **--org** option overwrites the default to something more relevant.

```
[root@server ~]# katello-configure --deployment=samm --user-pass=admin -  
-org=QA_Lab_West_Datacenter
```

Example 2.2. Creating an Admin User

An initial administrative user is created as part of the Subscription Asset Manager configuration. This user has the username **admin** and the password **admin**.

Most administrators will want to reset those values to something more secure. This can be done using the **--user-name** and **--user-pass** options. (There is one other configuration option for the admin user, to set that user's email address.)

For example:

```
[root@server ~]# katello-configure --deployment=sam --user-name=samadmin  
--user-pass=secret --user-email=admin@example.com
```

Example 2.3. Setting up LDAP Authentication

By default, Subscription Asset Manager maintains a database of its own configuration entries, including user entries. That database is used when processing user authentication requests. However, in many environments, there is already a comprehensive configuration of users and roles in

an LDAP directory; optionally, this LDAP directory can be used for Subscription Asset Manager authentication requests instead of its local database.

As covered in [Section 3.5.2.2, “Enabling LDAP Authentication”](#), the LDAP configuration is done in two parts. One file sets the authentication method and tells Subscription Asset Manager to use it both for users and for roles (**katello.yml**) and the other file sets the connection information for the LDAP directory (**ldap_fluff.yml**).

Configuration for both files can be passed at the time that Subscription Asset Manager is configured. The full list of LDAP attributes is covered in the Subscription Asset Manager man page and help output. This example shows the required settings for a POSIX LDAP directory such as Red Hat Directory Server or OpenLDAP. It is also possible to configure Subscription Asset Manager to use Microsoft Active Directory and Red Hat Identity Management.

```
[root@server ~]# katello-configure --deployment=sam --user-pass=admin --
auth-method=ldap --ldap-roles=true --ldap-server=ldap.example.com --
ldap-port=389 --ldap-server-type=":posix" --ldap-encryption=start_tls --
ldap-users-basedn=People,dc=example,dc=com --ldap-groups-
basedn="ou=Groups,dc=example,dc=com" --ldap-anon-queries=true
```

Example 2.4. Using an Answer File

Instead of passing the command parameters inline, it is possible to specify them in an answer file, which can then be passed with the script. This can allow an administrator to use kickstart or other automated provisioning systems and pass machine- or environment-specific information cleanly.

A default answer file is available at **/usr/share/katello/install/default-answer-file**. This can be copied to create your specific answer file.

The file itself contains a list of the arguments and default values; simply edit the desired lines to set the appropriate values for the environment.

```
# Path of the answer file.
answer-file =

# Katello administrative user (default: admin)
user-name = samadmin

# Katello user's password (default: admin)
user-pass = admin

# Katello user's email (default: root@localhost)
user-email = admin@example.com

# Katello initial Organization (default: ACME_Corporation)
org-name = Example_Org

....
# Deployment type (one of "katello", "headpin", "cfse", "sam")
deployment = sam

....
```

Then use the edited file with the **katello-configure** command, using the **--answer-file** argument. For example, if the file is **sam-config-file** in the **/tmp** directory:

```
[root@server ~]# katello-configure --answer-file=/tmp/sam-config-file
```

No **--deployment** option is needed because it was set to **sam** in the answer file.

2.5. SUBSCRIPTION ASSET MANAGER INSTALLATION LOGS

Subscription Asset Manager is comprised of different underlying components, including Candlepin (the subscription service), Thumbslug (the web UI), and Tomcat. Each individual component has its own logs, as does Subscription Asset Manager as a whole (under the name *katello*). Any installation errors are recorded in the component log.

- **/var/log/katello**
- **/var/log/thumbslug**
- **/var/log/candlepin**
- **/var/log/tomcat6**

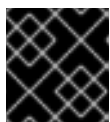
2.6. UPGRADING SUBSCRIPTION ASSET MANAGER

1. Back up the Subscription Asset Manager instance before beginning the upgrade procedure. Backing up a Subscription Asset Manager instance is covered in [Section 9.1, “Backing up Subscription Asset Manager”](#).
2. Stop all Subscription Asset Manager services:

```
[root@server]# katello-service stop
```

If the **katello-service** command is not available, then stop the associated services manually:

```
[root@server]# service katello stop
[root@server]# service katello-jobs stop
[root@server]# service httpd stop
[root@server]# service tomcat6 stop
[root@server]# service elasticsearch stop
```



IMPORTANT

Do not turn off the **postgresql** service.

3. Flush the indexes.

1. Start the Elasticsearch service:

```
# service elasticsearch start
```

2. Flush the indexes:

```
# curl localhost:9200/_flush
```

3. Stop the Elasticsearch service:

```
# service elasticsearch stop
```

4. Upgrade the Subscription Asset Manager packages with **yum**:

```
[root@server]# yum upgrade
```

5. **yum** avoids overwriting the current Subscription Asset Manager configuration by creating a set of **.rpmnew** and **.rpmsave** files. Check these files against the existing configuration for any new changes.

For a list of configuration files, run the following command:

```
[root@server]# rpm -ql katello-configure | grep erb | grep etc | sed  
's/.*etc/\n/etc/' | sed 's/\n.erb//'
```

For a list of **.rpmnew** files, run the following command:

```
[root@server]# find /etc -name *rpmnew
```

6. Upgrade Subscription Asset Manager using the upgrade script.

```
[root@server]# katello-upgrade
```

The upgrade script prompts for confirmation for each part of the upgrade process. For example:

```
1/1: Update Candlepin (0003_update_candlepin.sh)  
  Updates Candlepin database schema to the latest version  
  
Do you want to proceed? (y/n): y  
  
Update Candlepin OK.
```

7. Run **katello-configure** to correct any configuration files and analyze log output for errors.

```
[root@server]# katello-configure
```



NOTE

In some cases, the configuration process is not finished after running **katello-configure** once. You will need to run **katello-configure** again to finish the configuration process. You can safely run **katello-configure** as many times as required to finish the configuration process, but twice is usually sufficient.

8. Start the Subscription Asset Manager server:

```
[root@server]# katello-service start
```

9. Test the Subscription Asset Manager server:

```
[root@server]# katello-service status  
[root@server]# katello -u admin -p admin password ping
```

10. Update the port settings on every Subscription Asset Manager agent system.

The QPIDD port changed from 5674 to 5671 after Subscription Asset Manager 1.1. All systems connecting to Subscription Asset Manager using **katello-agent** must be updated to use port 5671.

1. Upgrade your system to the latest version of **katello-agent** and **goferd**.
2. Open the Katello plug-in configuration file:

```
[root@server]# vim /etc/gofer/plugins/katelloplugin.conf
```

3. Edit the **url** line to change port number.

```
url=ssl://$(host):5671
```

4. Restart the **katello-agent** and **goferd** services.
5. Open port 5671 on your Subscription Asset Manager firewall. For example, add the following rule to **iptables**:

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 5671 -j  
ACCEPT
```

2.7. LOGGING INTO THE SUBSCRIPTION ASSET MANAGER WEB UI

Subscription Asset Manager uses a web UI, which is configured when Subscription Asset Manager is installed.

2.7.1. Supported Browsers

The Subscription Asset Manager web UI can be accessed using these browsers:

- Firefox 21 and 22
- Internet Explorer 9

2.7.2. The Web UI URL

After it is configured, the Subscription Asset Manager UI is available using the hostname of the machine and the **/sam** directory, **https://hostname/sam**. For example:

```
https://server.example.com/sam
```

2.7.3. The Default User

Unless another username and password are created when Subscription Asset Manager is configured, the initial user has the username **admin** and the password **admin**.

CHAPTER 3. MANAGING USERS AND ACCESS CONTROLS

Access controls are implemented per-organization through roles which define what users can access what elements of the organization.

3.1. ABOUT USERS, ROLES, AND ACCESS CONTROLS

Security establishes precise relationships between users, resources, and the tasks users can perform. Interactions between users and resources are ordered by including or excluding those users and resources (through groups) in defined roles, and then granting the role the ability to perform tasks.

When a user is allowed to perform a certain operation, that is called a *permission*.

Users are granted permissions (such as read, edit, create, and delete) to elements within a specific organization. These permissions must be explicitly granted; by default, all actions are implicitly denied to users.

Permissions are granted to users through *roles*. A role defines three elements:

- The organization or organizations to which is applies
- Users which belong to the role
- The permissions which those users have within the organization

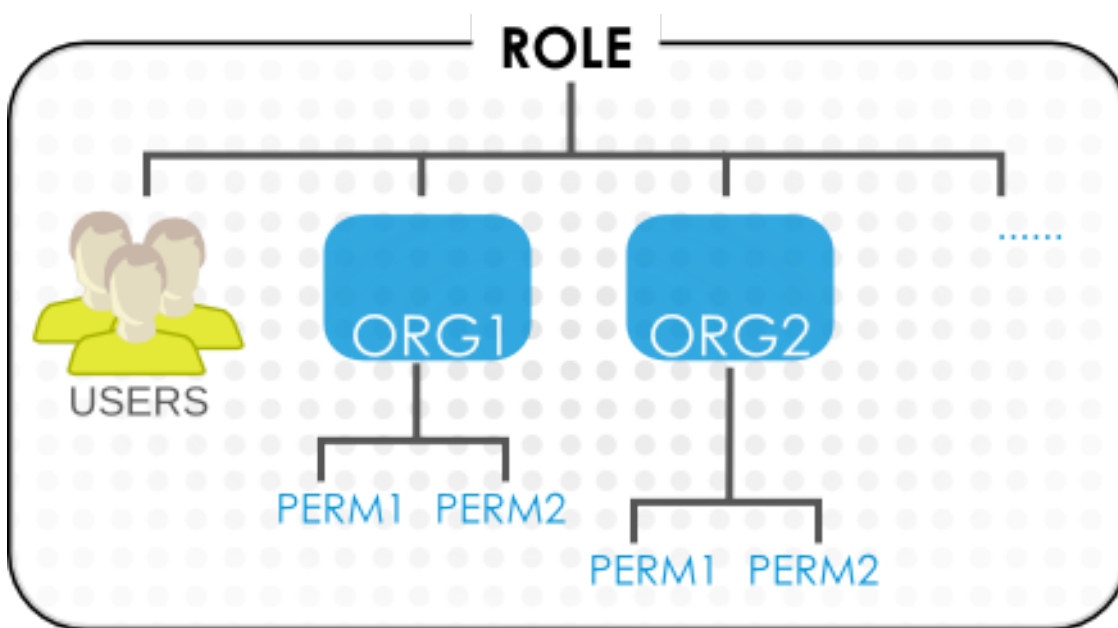


Figure 3.1. Users, Organizations, and Permissions in a Role

A single role can be associated with multiple organization, but the permissions are set on each organization individually. So, configure all the permissions for one organization and then select another and configure all of the permissions for that.

The permissions within Subscription Asset Manager are highly specific. The permissions themselves define both an action and the target to which that action is permitted. For example, one permission is *register systems*. That defines both the action (register) and the target (systems within the organization). Registering other objects is a separate permission, as is performing other tasks on systems.

[Table 3.1, “Subscription Asset Manager Components and Allowed Permissions”](#) lists the available permissions. The number of permissions allow substantial flexibility in creating roles that meet business

needs and providing adequate controls on access. The specificity of the permissions makes defining access controls easier since the action and target are always clear.

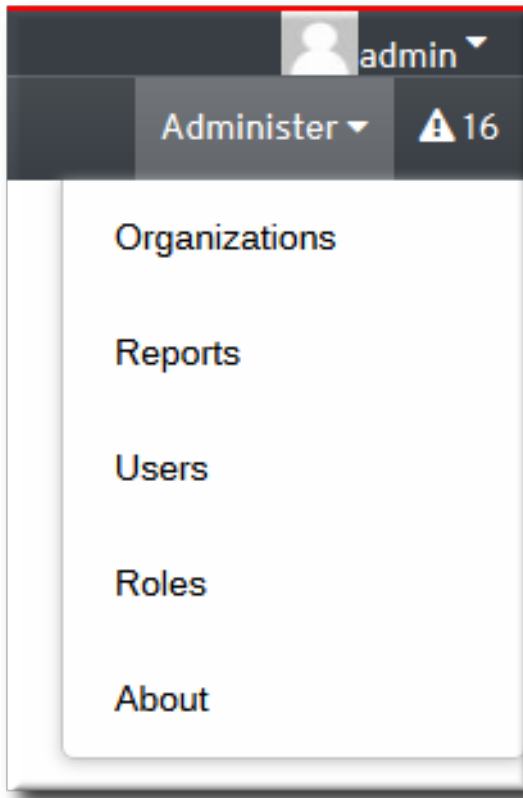
Table 3.1. Subscription Asset Manager Components and Allowed Permissions

Component	Possible Permissions
Organization: Organization Entries	<ul style="list-style-type: none"> • Modify • Read
Organization: Distributor Entries	<ul style="list-style-type: none"> • Register • Read • Modify • Delete
Organization: System Entries	<ul style="list-style-type: none"> • Register • Read • Modify • Delete
Activation Keys	<ul style="list-style-type: none"> • Read • Modify
System Groups	<ul style="list-style-type: none"> • Modify the system group • Read the system group • Modify systems in the group • Read systems in the group
Providers	<ul style="list-style-type: none"> • Read • Modify

3.2. MANAGING USER ACCOUNTS

3.2.1. Adding Users

1. Hover over the **Administer** menu, and click the **Users** item.



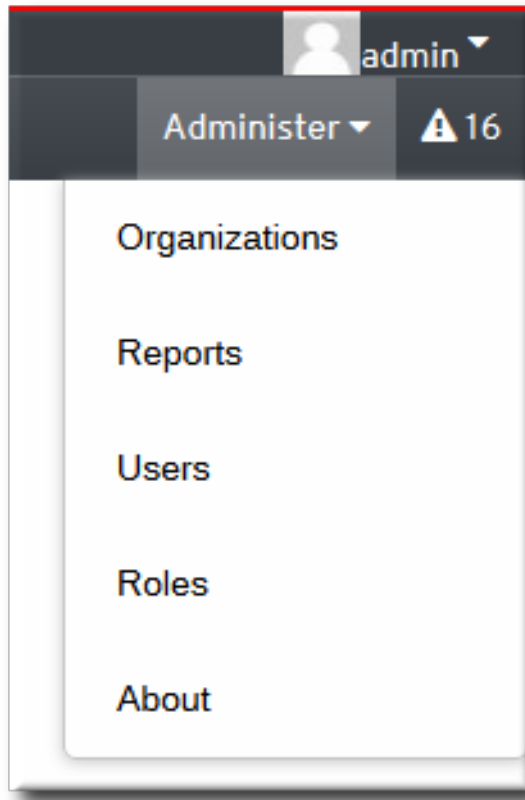
2. Click the + **New User** link in the left column of the **Users** page.
3. Enter the information for the new user, including a username and password.

A screenshot of the 'New User' form in the Subscription Asset Manager UI. The form is divided into two main sections. On the left, under the 'Users' header, there is a table with two columns: 'USERNAME' and 'USERID'. The table contains two rows: 'admin' and 'jsmith'. Above the table, it says 'Viewing 2 of 2 results (2 Total Users)'. On the right, under the 'New User' header, there are input fields for 'Username' (containing 'bjensen'), 'Email Address' (containing 'bjensen@example.com'), 'Password' (masked with dots), and 'Confirm' (masked with dots). Below these fields is a section titled 'System Registration Defaults' with a dropdown menu for 'Organization' (set to 'No Default') and a 'Save User' button.

4. Select a default organization. This is the organization which automatically comes up when the user logs into the Subscription Asset Manager UI. If no default is set, then the default in the Subscription Asset Manager configuration is used.
5. Click the **Save User** button.

3.2.2. Changing a Password

1. Hover over the **Administer** menu, and click the **Users** item.

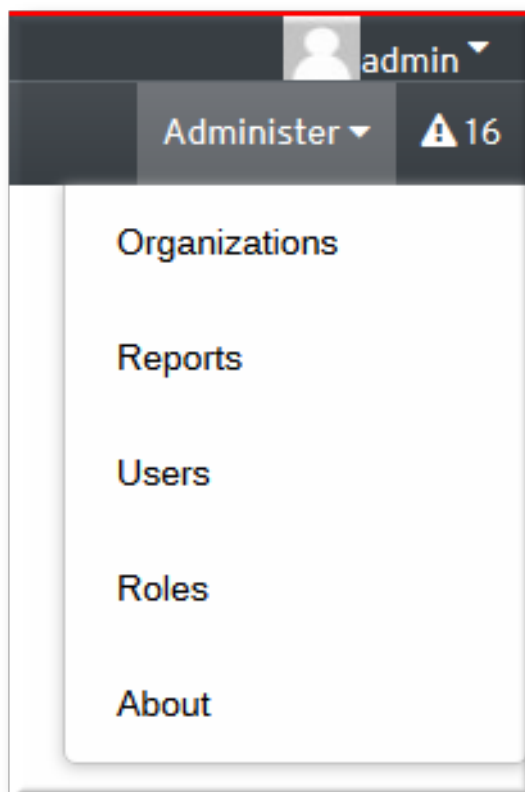
**NOTE**

To change your own personal password, click the username displayed in the admin menu.

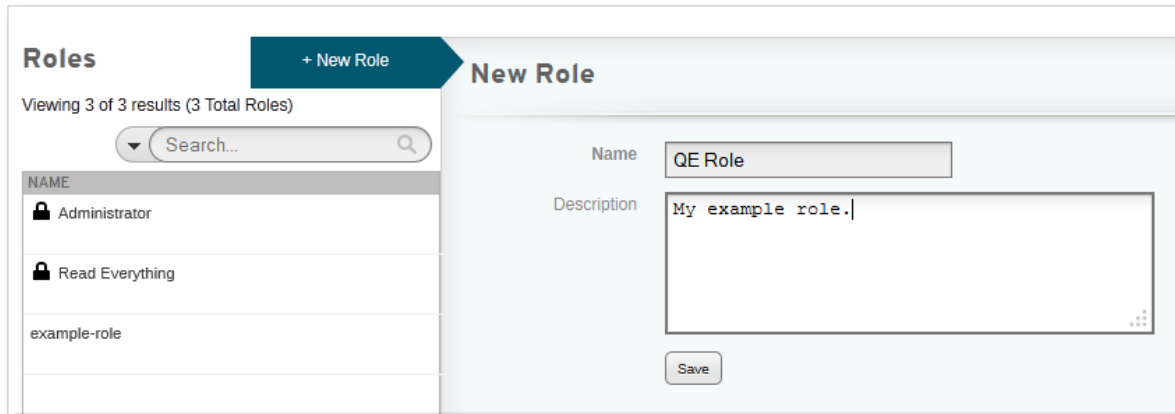
2. Select the user from the list in the column on the left of the **Users** page.
3. Enter the new password in the **Change Password:** field, and then re-enter it to confirm it.
4. Click the **Save** button.

3.3. CREATING A ROLE

1. Hover over the **Administer** menu, and click the **Roles** item.



2. Click the **+ New Role** link in the left column of the **Roles** page.
3. Enter the name of the role and, optionally, a description.

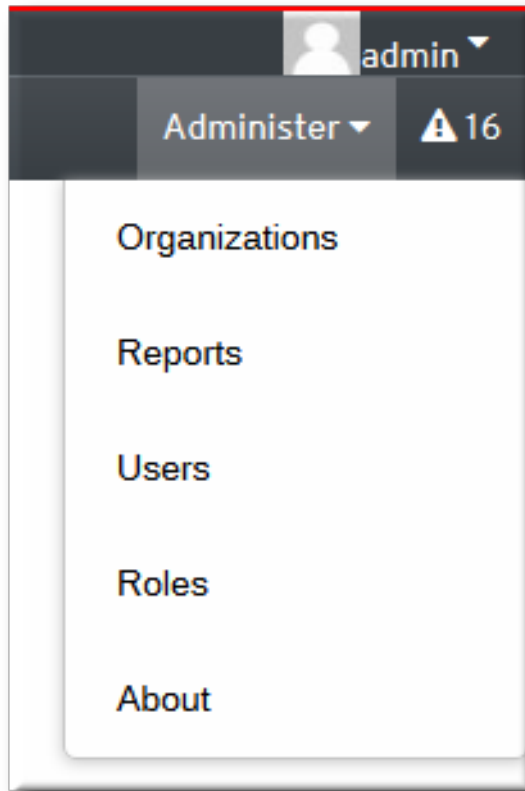


4. Click the **Save** button.

3.4. SETTING UP ACCESS CONTROLS

Access controls are configured by adding users and permissions to a role.

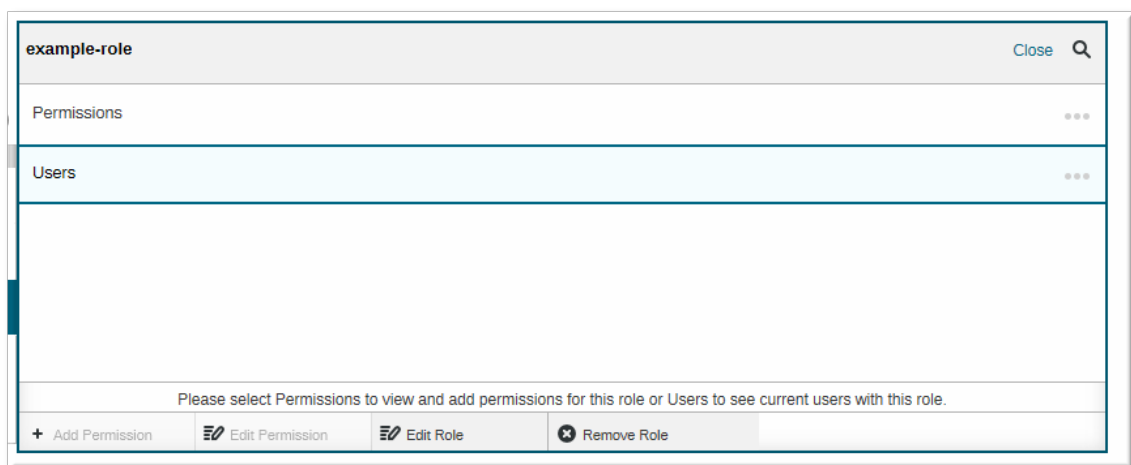
1. Hover over the **Administer** menu, and click the **Roles** item.



2. Click the name of the role.
3. Set up the permissions for the role.

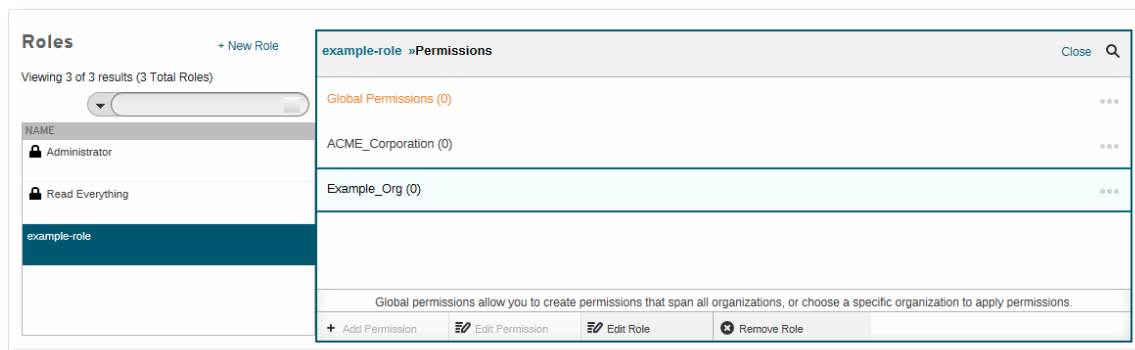
The **Permissions** area defines two things: the organizations which are associated with the role and the permissions granted for those organizations.

1. Click **Permissions**.

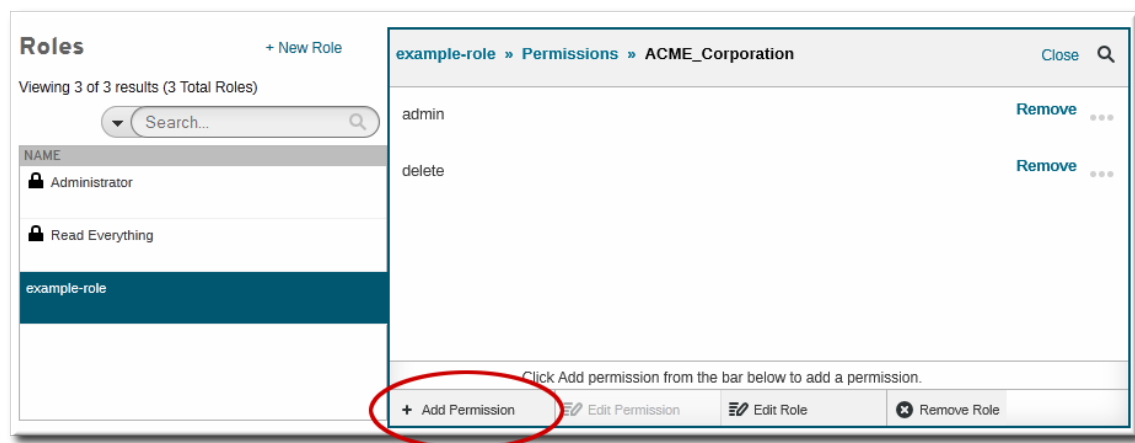


2. Select the organization.

A single role can be associated with multiple organization, but the permissions are set on each organization individually. So, configure all the permissions for one organization and then select another and configure all of the permissions for that.



- At the bottom of the window, click the **Add Permission** item.



- Select the component (organization, activation key, provider, or group) for which to add the permission.
- Click **Next**.
- When the target is selected, then there is a selection box which lists the available permissions for that target. The available permissions are outlined in [Table 3.1, "Subscription Asset Manager Components and Allowed Permissions"](#).

7. Click **Next**.
8. Enter a name and, optionally, a description for the permission.
9. Click the **Done**.
4. Click the name of the role at the top of the role edit window.

5. Next, add at least one administrator user for the role. Even if LDAP authentication ([Section 3.5, “Managing Users with LDAP Authentication”](#)) is used for other users in the role, at least one administrator from within Subscription Asset Manager is required.

1. Click the **Users** item.

2. All of the users in Subscription Asset Manager are listed. Click the **Add** nad **Remove** labels by the usernames to manage membership in the role.



3.5. MANAGING USERS WITH LDAP AUTHENTICATION

Most environments will have a user directory already deployed with users, passwords, and groups already configured. Subscription Asset Manager can tap into an existing LDAP directory and use it for authentication or for both authentication and role management.

3.5.1. Supported LDAP Service Types

Subscription Asset Manager can use several common LDAP directories as a user backend:

- Red Hat Directory Server (as a POSIX directory)
- OpenLDAP (as a POSIX directory)
- Red Hat Identity Management
- Microsoft Active Directory

3.5.2. Using LDAP Authentication

3.5.2.1. About LDAP Users in Subscription Asset Manager

When LDAP authentication is enabled, it is a form of *pass-through authentication*. The Subscription Asset Manager server receives the username and password and forwards that to the configured LDAP server. The Subscription Asset Manager server never stores or processes the user information; it depends on the response from the LDAP server to determine whether to allow the user to log in.

AUTHENTICATION PROCESS

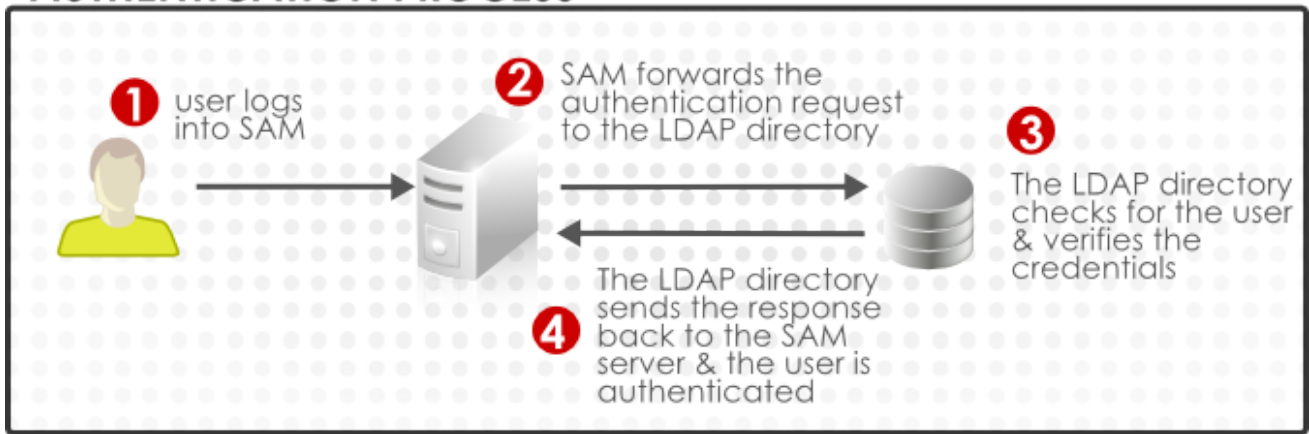


Figure 3.2. User Authentication with LDAP

LDAP authentication allows the security measures in the LDAP server such as password complexity and account deactivation to be applied to Subscription Asset Manager users. This means that corporate standards can be consistently and transparently applied to Subscription Asset Manager users.

There are some caveats when using an LDAP directory for Subscription Asset Manager authentication:

- Only Subscription Asset Manager database authentication *or* LDAP authentication can be used, not both.
- **All users must already exist in the LDAP directory for them to be able to access Subscription Asset Manager.** Subscription Asset Manager pulls information from LDAP. It cannot create an LDAP user.

If no corresponding LDAP user account exists, then attempting to log into Subscription Asset Manager fails with this error:

```
User must exist in ldap before defining here
```

- A corresponding Subscription Asset Manager user account is created whenever an LDAP user first authenticates to Subscription Asset Manager.

3.5.2.2. Enabling LDAP Authentication

LDAP authentication is configured in two files.

- The **katello.yml** configuration file is what flags Subscription Asset Manager to use LDAP authentication (**warden:**) and to use LDAP for roles (**ldap_roles:**).
- Subscription Asset Manager uses the Ruby LDAP Fluff module for authentication details. The information on the LDAP server, such as its type, hostname and port, and user base distinguished name (DN), are defined in the **ldap_fluff.yml** file.

All of these configuration parameters can be set using the **katello_configure** command.

1. Create a new Subscription Asset Manager administrative user that has the same username as an administrative user in the LDAP directory. Creating users is covered in [Section 3.2.1, “Adding Users”](#).

**WARNING**

Once LDAP authentication is enabled, no existing Subscription Asset Manager users can log into the Subscription Asset Manager instance. If no LDAP user is seeded into the Subscription Asset Manager instance, then you could be locked out of the server once LDAP authentication is enabled.

2. Add the admin user to a global administrator role.
3. Set the Subscription Asset Manager instance to use LDAP authentication instead of its local database.

```
[root@server ~]#katello-configure --auth-method=ldap
```

4. Set the LDAP connection information.

The required settings are slightly different, depending on what type of LDAP directory is used.

There are some settings that are required by all directory types:

- **--ldap-server-type** to identify the backend LDAP directory
- **--ldap-server** for the hostname or IP address of the LDAP server
- **--ldap-port** for the standard LDAP port
- **--ldap-users-basedn** for the DN (or directory location) of the users subtree
- **--ldap-groups-basedn** for the DN (or directory location) of the groups subtree

Then each directory has addition configuration requirements.

For POSIX directories

- **--ldap-encryption**, which must be set to **Start_tls**
- **--ldap-anon-queries** set to **true** to allow anonymous searches

For example:

```
[root@server ~]#katello-configure --ldap-server-type=":posix" --
ldap-server=rhds.example.com --ldap-port=389 --ldap-users-
basedn=ou=people,dc=example,dc=com --ldap-groups-
basedn=ou=groups,dc=example,dc=com --ldap-encryption=start_tls --
ldap-anon-queries=true
```

For Active Directory directories

- **--ldap-ad-domain** to pass the Active Directory domain name
- **--ldap-anon-queries** set to **false** to prevent anonymous searches

- **--ldap-service-user**, which sets an identity to use for directory queries
- **--ldap-service-pass**, which gives the password for the service user

For example:

```
[root@server ~]#katello-configure --ldap-server-
type=":active_directory" --ldap-server=ads.example.com --ldap-
port=389 --ldap-users-basedn=cn=Users,dc=win-ads,dc=example,dc=com -
--ldap-groups-basedn=cn=Domain Admins,cn=Users,dc=win-
ads,dc=example,dc=com --ldap-anon-queries=false --ldap-service-
user=Administrator --ldap-service-pass=secret --ldap-ad-
domain=ads.example.com
```

For Red Hat Identity Management directories

- **--ldap-anon-queries** set to **false** to prevent anonymous searches
- **--ldap-service-user**, which sets an identity to use for directory queries
- **--ldap-service-pass**, which gives the password for the service user

For example:

```
[root@server ~]#katello-configure --ldap-server-type=":free_ipa" --
ldap-server=ipa.example.com --ldap-port=389 --ldap-users-
basedn=dc=example,dc=com --ldap-groups-
basedn=cn=groups,cn=accounts,dc=example,dc=com --ldap-anon-
queries=false --ldap-service-user=admin --ldap-service-pass=secret
```

5. Restart Subscription Asset Manager:

```
[root@server ~]# katello-service restart
```

3.5.3. Using LDAP Group and Role Mappings

3.5.3.1. About LDAP Groups and Subscription Asset Manager Roles

Subscription Asset Manager can optionally use LDAP for group and role configuration, as well. Access controls in Subscription Asset Manager are applied through roles. When LDAP groups are enabled, then an LDAP group is mapped directly to a Subscription Asset Manager role, almost like a member of the role.

The role membership is then essentially maintained in the LDAP directory. Whenever a user is added to a group, that user is automatically a member of any Subscription Asset Manager role to which the LDAP group belongs. Likewise, when that user is removed from the group, it no longer belongs to the Subscription Asset Manager role. This allows for more dynamic role management, since it incorporates groups rules on the LDAP server.

Using LDAP groups in roles works in tangent with LDAP authentication. It is possible to use LDAP authentication (users) without using LDAP groups. However, using LDAP groups requires also using LDAP users.

3.5.3.2. Enabling LDAP Group-Role Mapping

1. Set the **ldap-roles** configuration setting to **TRUE**:

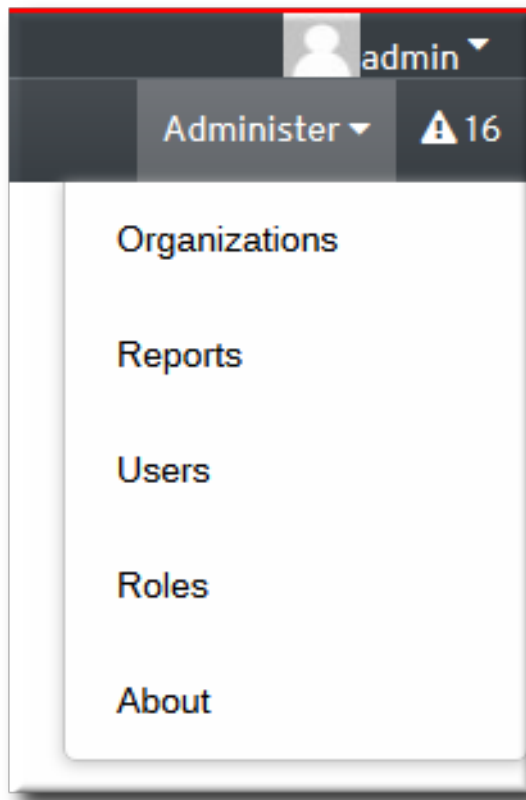
```
[root@server ~]# katello-configure --ldap-roles=true
```

2. Restart the **katello** service to load the new configuration.

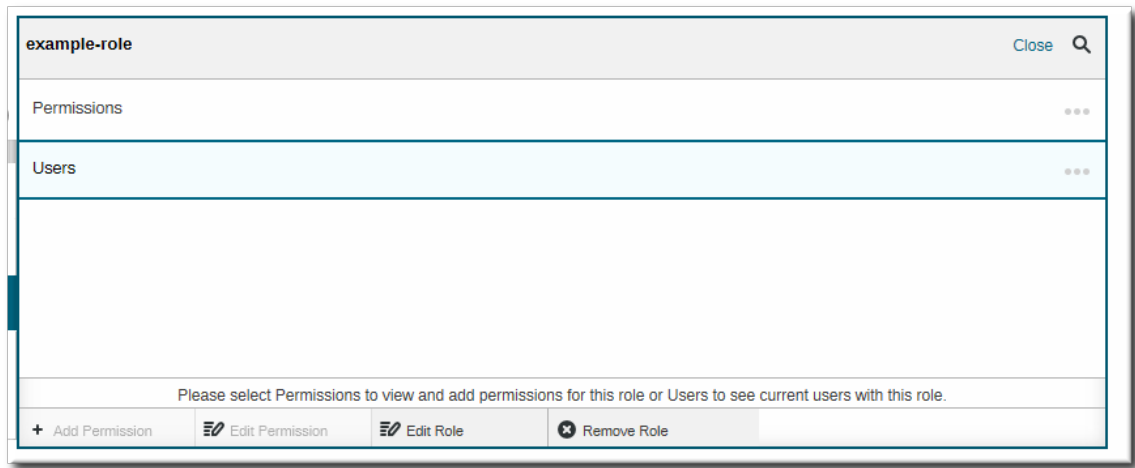
```
[root@server ~]# service katello restart
```

3. Add the LDAP group to the desired Subscription Asset Manager roles.

1. Hover over the **Administer** menu, and click the **Roles** item.



2. Click the name of the role to edit in the list in the left column of the **Roles** page.
3. Click the **Users** option for the role.



4. Click **LDAP Groups**, and select as many LDAP groups as required for this role.

Next time a user from the chosen LDAP group logs in, they will be assigned to the mapped roles.

CHAPTER 4. SETTING UP ORGANIZATIONS AND DISTRIBUTORS

Subscription Asset Manager manages subscriptions on-premise, as opposed to the default configuration where both subscriptions and content are administered through Customer Portal Subscription Management.

Subscriptions are distributed from the main account (at the Portal) to an organization within a Subscription Asset Manager instance. This is an important point — a Subscription Asset Manager server does not manage anything. It is organizations *within* the Subscription Asset Manager instance that are registered to the Portal and then subsequently handle local subscriptions and systems.

4.1. ABOUT THE STRUCTURE OF ORGANIZATIONS AND DISTRIBUTORS

The organizations in Subscription Asset Manager provide structure to the systems within the infrastructure.

The organization structure can be flat, with a single local organization. Alternatively, it can have multiple organizations which are independent from each other. Organizations are always opaque to one another, with separate subscription and system inventories. Using multiple organizations allows multiple, independent groups to be attached and to manage their own subscriptions and systems through local services.

Subscription Asset Manager does not host content, only subscription services. Local systems still must connect to Customer Portal Subscription Management for content or go through the Subscription Asset Manager server as a proxy.

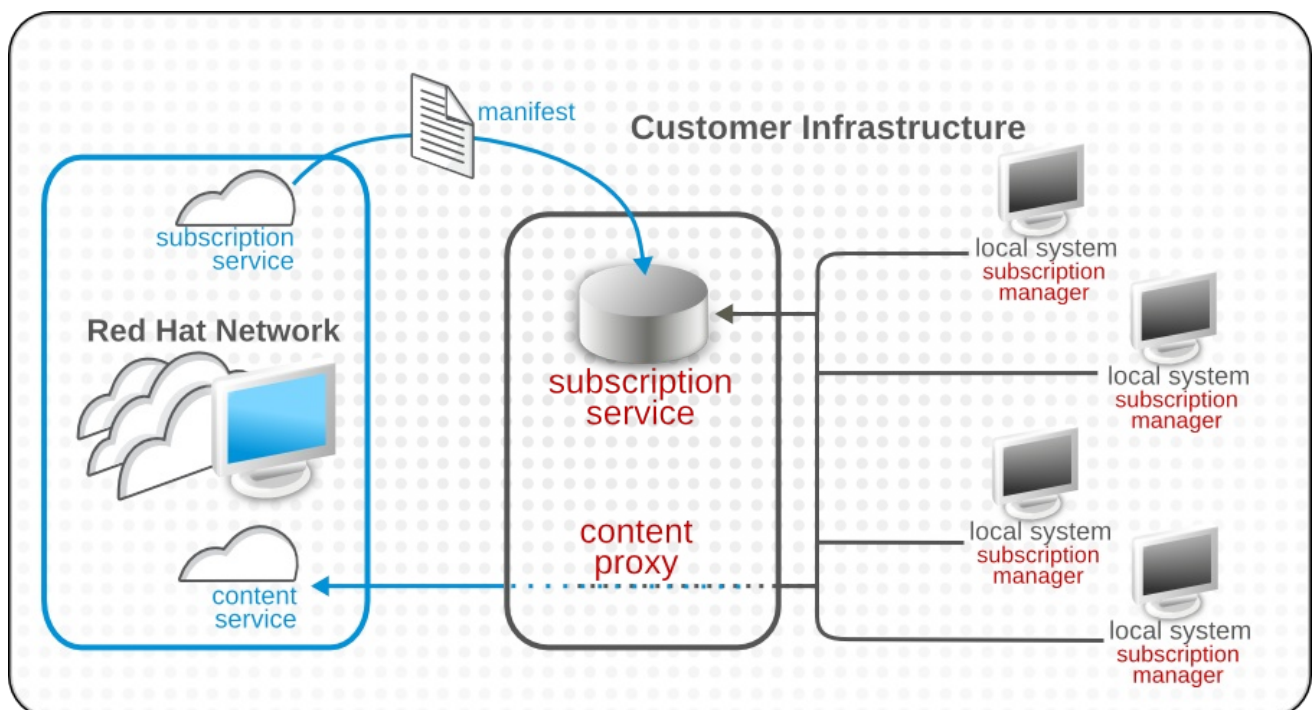


Figure 4.1. Hosted Content Delivery and On-Premise Subscription Service

A system is assigned to one organization. It is identified with that organization. An organization must be defined for every system, as part of its registration process.

An organization within Subscription Asset Manager is registered to the account in the Portal. This is a high-level division within the infrastructure.



IMPORTANT

Creating distributors is a tech preview feature.

The Portal and the Subscription Asset Manager organization communicate over a secure but external network connection. There may be some instances where a particular Subscription Asset Manager organization cannot or should not be accessible over a public network or a business reason why a Subscription Asset Manager organization should be subdivided into child organizations, which are local to each other.

Subscription Asset Manager allows *distributors* to be created underneath a Subscription Asset Manager organization. This is essentially an instance within an instance. The parent organization attaches subscriptions to the child organization and creates a manifest locally.

A distributor is never created in the Customer Portal and is never visible there. It is created and maintained under another Subscription Asset Manager instance.

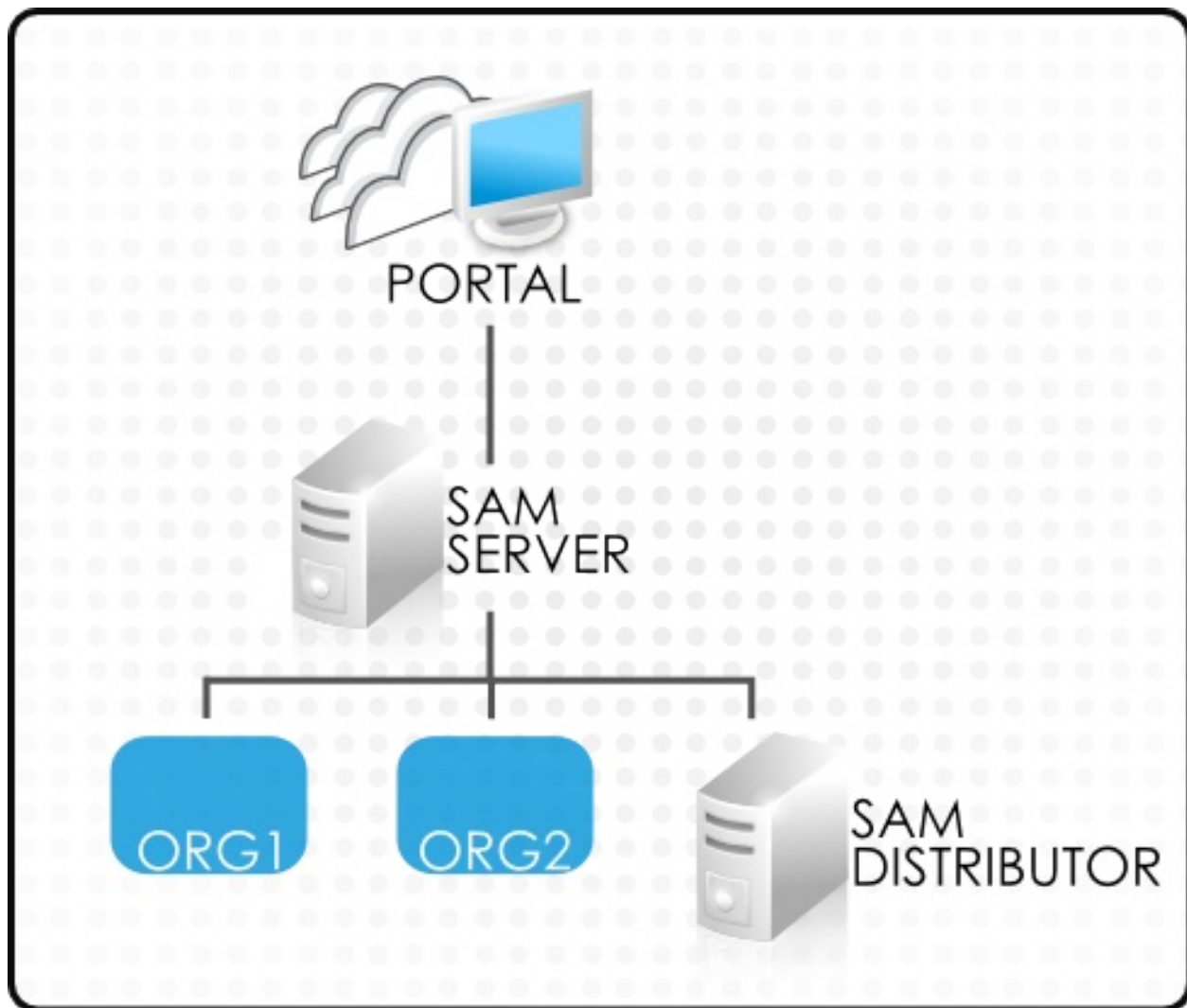


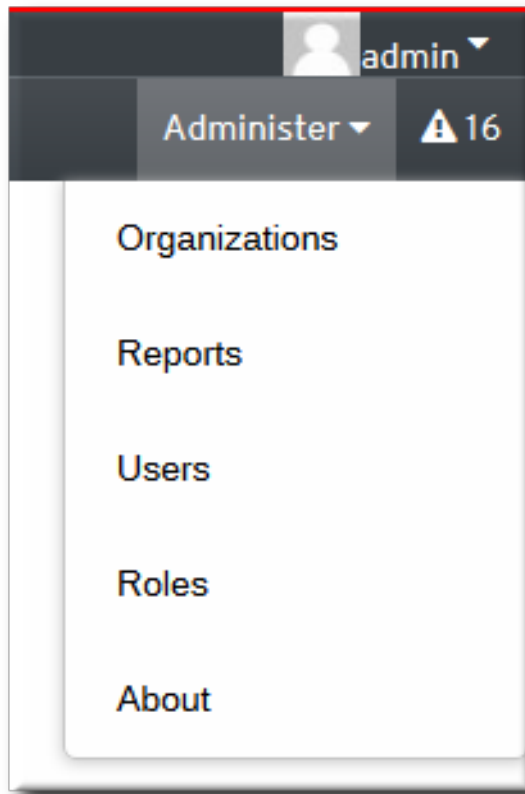
Figure 4.2. A Hierarchy with Distributors

A distributor is one way for a Subscription Asset Manager instance to subdivide its subscriptions.

4.2. MANAGING ORGANIZATIONS

4.2.1. Creating a New Organization

1. Create the organization in Subscription Asset Manager.
 1. Hover over the **Administer** menu, and click the **Organizations** item.

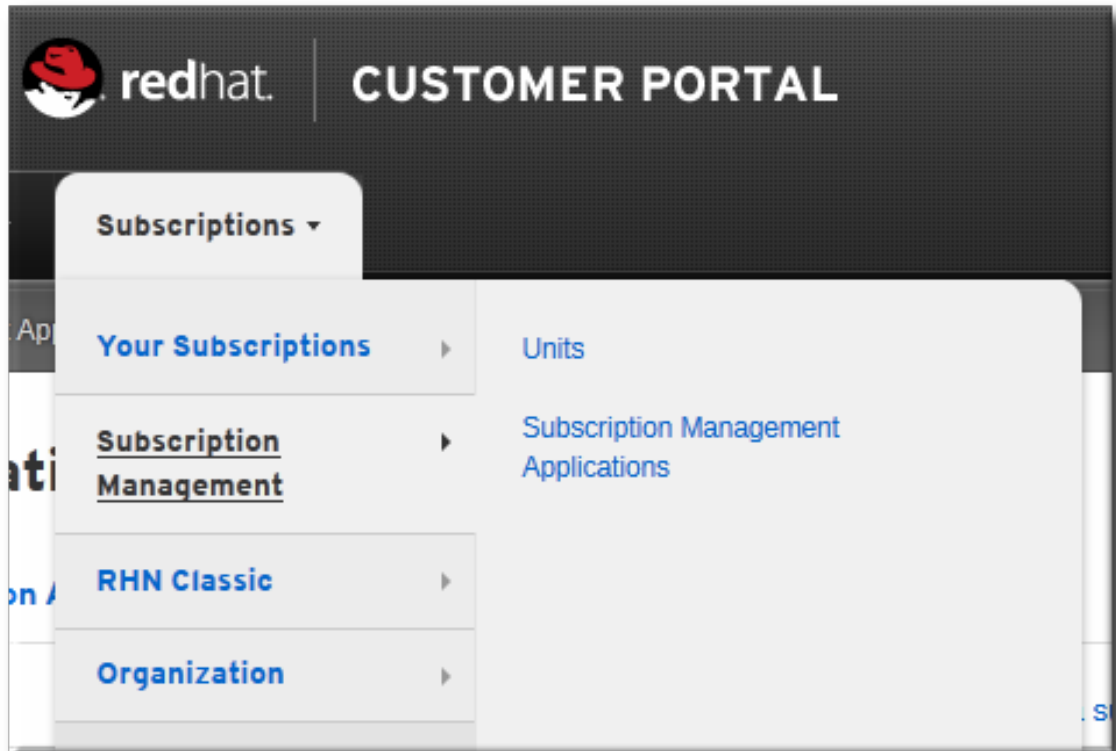


2. In the left column, click the **+ New Organization** link.
3. Fill in the name of the new organization, and optionally, a description. A label (internal identifier) is automatically created based on the name, but this can be edited.

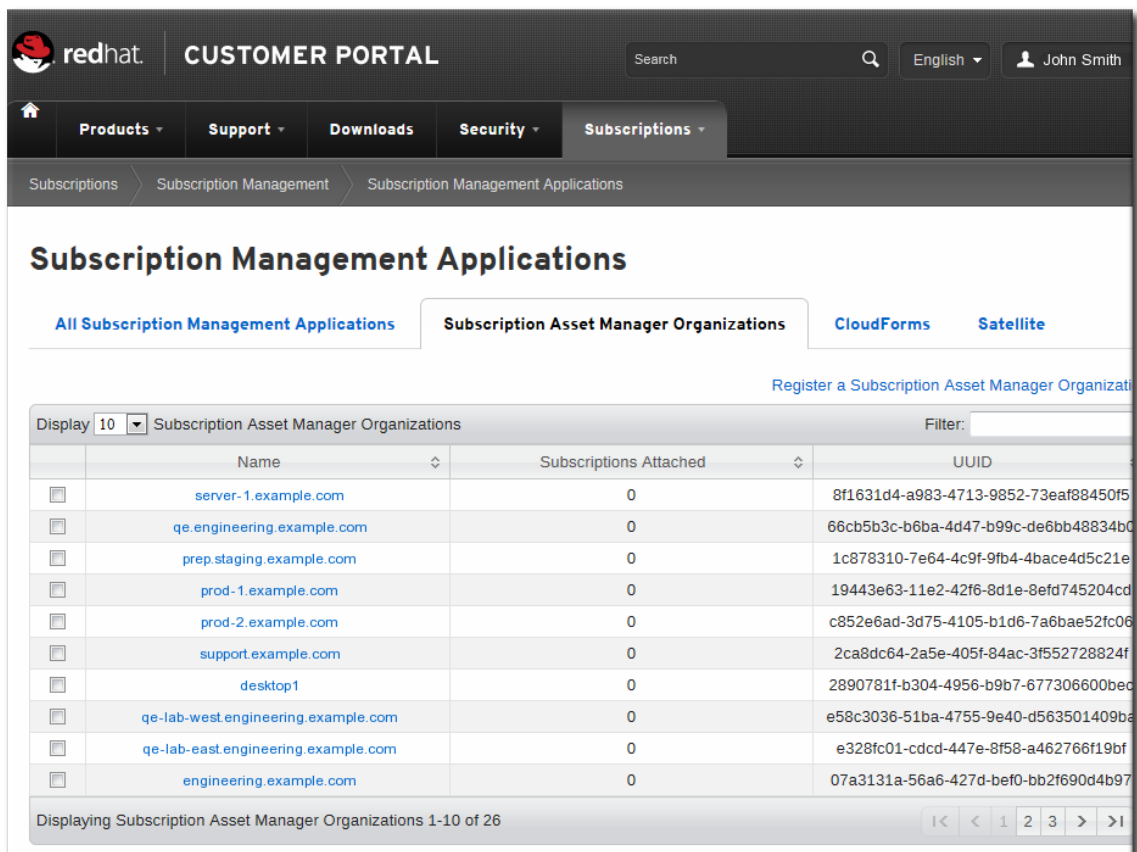
A screenshot of the 'New Organization' form in the Subscription Asset Manager. The interface is split into two main sections. On the left, there is a table titled 'Organizations' showing a list of existing organizations: 'ACME_Corporation' and 'Example_Org'. Above the table is a search bar and a '+ New Organization' button. On the right, there is a form titled 'New Organization' with three input fields: 'Name' (containing 'New Org'), 'Label' (containing 'New_Org'), and 'Description' (containing 'My new organization.'). Below the form is a 'Save' button.

4. Click the **Save** button.
2. Register the organization in the Customer Portal Subscription Management. There is a direct relationship between the organization in Subscription Asset Manager and the organization in the Red Hat account configuration.

1. Log into the Customer Portal.
2. Open the **Subscriptions** tab, hover over the **Subscription Management** item, and select the **Subscription Management Applications** item.

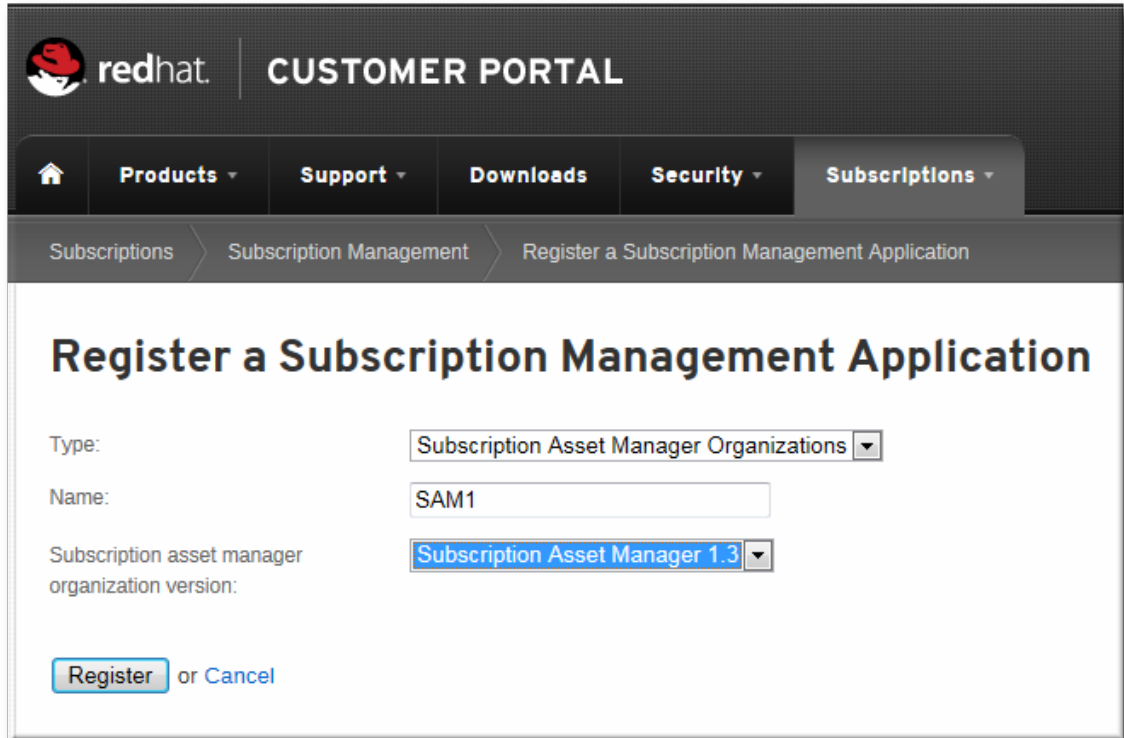


3. In the **Subscription Asset Manager Organizations** tab, click the **Register a Subscription Asset Manager Organization** link.



4. Fill in the required information for the new organization:

- The name for the organization
- The type of the organization; the options are supplied based on the available subscriptions for the account
- The version of the Subscription Asset Manager instance; the options are based on the available subscriptions for the account



The screenshot shows the Red Hat Customer Portal interface. The top navigation bar includes the Red Hat logo and the text 'CUSTOMER PORTAL'. Below this is a secondary navigation bar with links for Home, Products, Support, Downloads, Security, and Subscriptions. The 'Subscriptions' link is active, leading to a breadcrumb trail: Subscriptions > Subscription Management > Register a Subscription Management Application. The main content area is titled 'Register a Subscription Management Application'. It contains a form with the following fields: 'Type' (a dropdown menu set to 'Subscription Asset Manager Organizations'), 'Name' (a text input field containing 'SAM1'), and 'Subscription asset manager organization version' (a dropdown menu set to 'Subscription Asset Manager 1.3'). At the bottom of the form are two buttons: 'Register' and 'Cancel'.



NOTE

This name should correspond to the organization name set in Subscription Asset Manager.

5. Click the **Register** button.

3. In the Customer Portal, attach subscriptions to the organization, and download and import the manifest in the Subscription Asset Manager organization entry, as described in [Section 4.4.2, “Managing Manifests for Organizations”](#).

4.2.2. Removing an Organization

To remove an organization, click the **Remove** link on the organization's entry and confirm it.

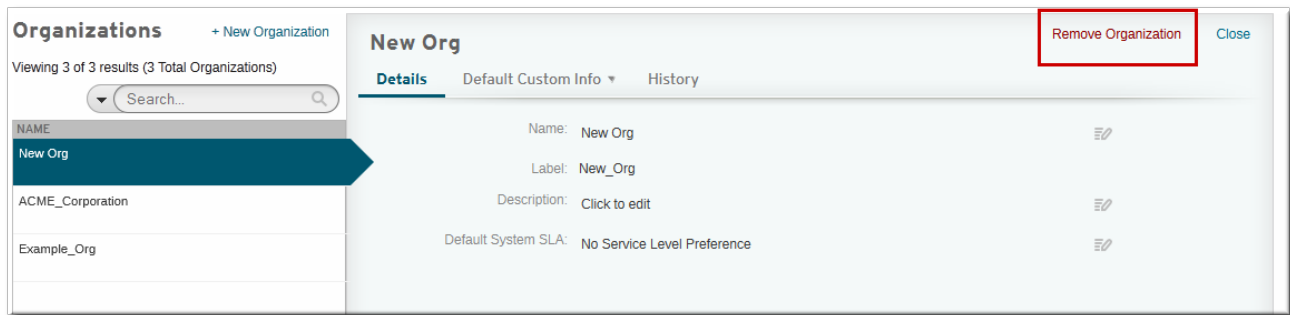


Figure 4.3. Removing an Organization

Removing an organization does more than delete the entry. Every system registered against that organization is immediately unregistered. Likewise, every subscription attached to those systems is removed and can be applied to other systems and organizations.

4.2.3. Setting the Default Organization

Every Subscription Asset Manager instance has a default organization. This is the first organization which opens in the web UI, if no other is selected.

To change the default organization, click the **Make this my default organization** link at the bottom of the desired organization's entry page.

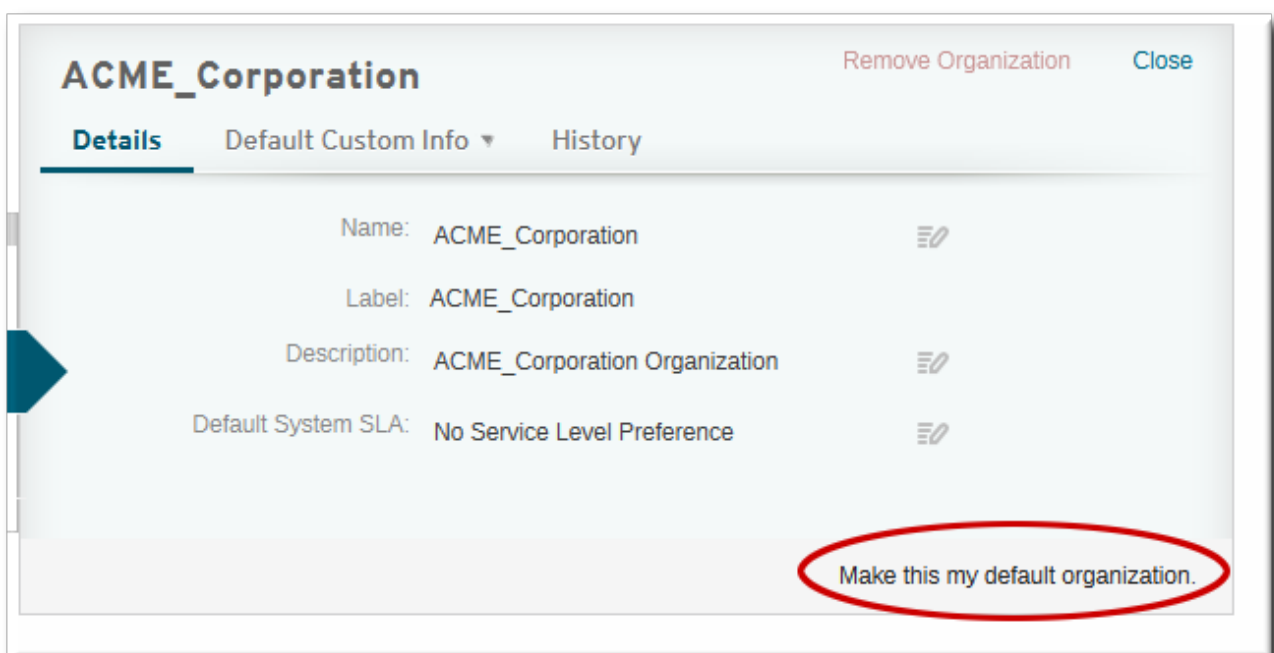


Figure 4.4. Setting a Default Organization

4.2.4. Setting Autoattach Preferences

Autoattaching and updating subscriptions selects what subscriptions to attach to a system based on a variety of criteria, including current installed products, hardware, and architecture. Most of the factors used for selecting the best-matched subscriptions are based on the characteristics of the system, but it is possible to include characteristics of the subscriptions.

Part of a subscription is recognizing the service level for a product on a given system. That service level can be used as a criterion for selecting a subscription to attach to a system.

Red Hat service levels are defined in the contract; a summary of production support levels is available at <https://access.redhat.com/support/offerings/production/sla.html>.

An account can have multiple levels of support available, even for the same product. The support level for a given system can be configured so that the appropriate level of support is available. A production system usually has a premium support level since it is a business critical system, while a development system may have standard support or be self-supported.

Setting a preference for the support level for the organization means that subscriptions of that service level are attached to systems first by default, unless the system-level setting is different or no subscriptions of that level are available.



NOTE

By default, the highest available level of support is selected for the subscription and system.

To set the default service level for systems in the organization, select the appropriate level from the drop-down menu:

- Standard
- None
- Premium
- Self-Support

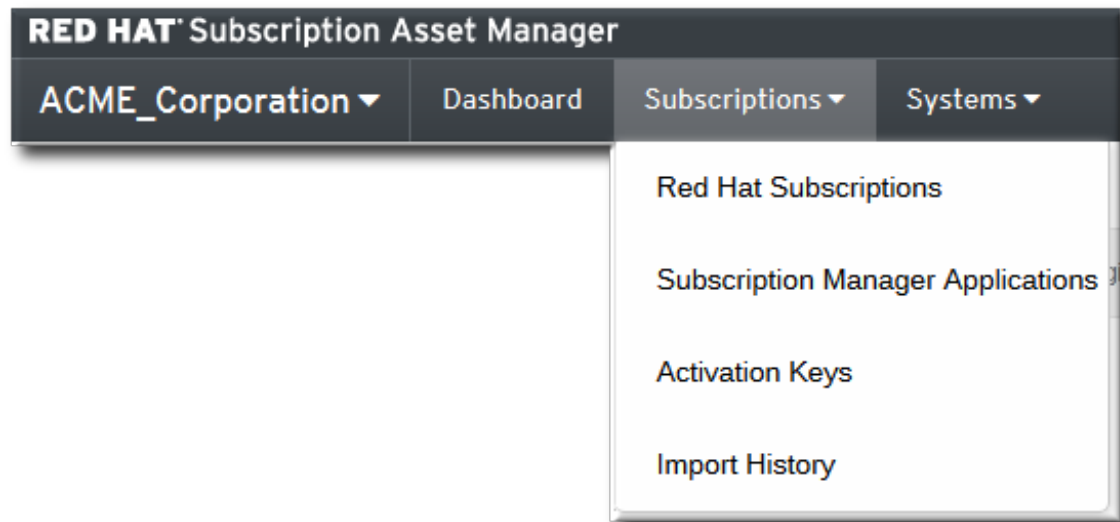
The screenshot shows the 'ACME_Corporation' organization details. The 'Details' tab is selected. The 'Default System SLA' dropdown menu is highlighted with a red box, showing 'No Service Level Preference' as the selected option. The page also shows fields for Name, Label, and Description, all set to 'ACME_Corporation'.

Figure 4.5. Setting Autoattach Preferences

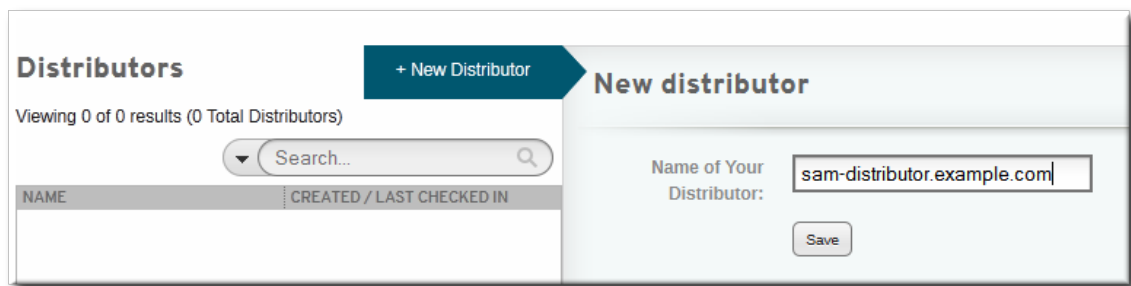
Only available service levels are available in the drop-down menu, based on the subscriptions in the organization's manifest. If no premium subscriptions are available, for example, then the premium service level is not listed in the menu.

4.3. MANAGING DISTRIBUTORS (TECH PREVIEW)

1. Create the distributor entry in the Subscription Asset Manager organization.
 1. Hover over the **Subscriptions** menu, and click the **Subscription Manager Applications** item.



2. In the left column, click the **+ New Distributor** link.
3. Fill in the name of the new distributor.



4. Click the **Save** button.
2. Set up the other Subscription Asset Manager instance and organization.
3. In the original Subscription Asset Manager instance, attach subscriptions to the organization, and download and import the manifest in the other Subscription Asset Manager organization, as described in [Section 4.4.3, “Managing Manifests for Distributors \(TECH PREVIEW\)”](#).

4.4. IMPORTING AND MAINTAINING MANIFESTS

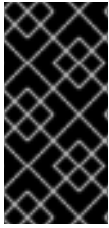
Every registered system has a certain number of software entitlements. These entitlements are listed in a manifest. Manifests are used by Subscription Asset Manager to manage your systems on site, without having to log directly in to your Red Hat Network account.

4.4.1. About Manifests

there is a direct relationship between the subscription management application organization in Customer Portal Subscription Management and the organization definition in an on-premise application like Subscription Asset Manager. This relationship is the method that Customer Portal Subscription Management uses to transfer subscriptions from Red Hat to the on-premise application to administer locally.

This transferred block of subscriptions is listed in the *subscription management application organization manifest*. This manifest is a ZIP archive which is downloaded directly from Customer Portal Subscription Management from the subscription management application organization entry and then is uploaded to

the on-premise application.



IMPORTANT

Any changes to the subscriptions for the organization are made to the subscription attached to the subscription management application organization entry in Customer Portal Subscription Management. The manifest is then regenerated, downloaded, and re-uploaded to the application.

The manifest itself is a collection of directories and JSON files which contain the subscriptions, entitlement certificates, products, and list of rules for the subscription management application organization.

```
manifest.zip
|
|- consumer_export.zip
|
|- export/
|   |
|   |- consumer_types/
|   |
|   |- entitlements/
|   |
|   |- entitlement_certificates/
|   |
|   |- products/
|   |
|   |- rules/
|   |
|   |- consumer.json
|   |
|   |- meta.json
```

consumer.json and meta.json

These JSON files contain a little information about the application organization entry (the UUID) and the manifest itself (version and creation date).

consumer_types/

consumer_types/ contains a list of JSON files, one for each supported application type. The JSON files indicate which type the subscriptions are attached to. For example, for Subscription Asset Manager, the **sam.json** has a **manifest** value of true.

```
{"id": "5", "label": "sam", "manifest": true}
```

entitlements/

entitlements/ contains a JSON file for each subscription attached to the application organization. Each file is named *UUID.json*.

The file contains the complete subscription information, including the contract number, pool ID, contract start/end dates, keys and certificates for the subscription, the product ID for each included product, quantities, and any other information associated with the subscriptions.

For example, this is the information for a single Red Hat Enterprise Linux product in a subscription JSON:

```
...
{"id":"8a878dcd3520d43501353f6f98f911e9","productName":"Red Hat Enterprise
Linux Server","productId":"69","updated":"2012-02-
02T18:59:32.000+0000","created":"2012-02-
02T18:59:32.000+0000"}], "endDate":"2012-10-
13T03:59:59.000+0000","quantity":50,"productName":"Red Hat Enterprise
Linux Server, Premium (4 sockets) (Up to 4
guests)","contractNumber":"2625891","accountNumber":"1506376","productId":
"RH0153936","subscriptionId":"2267347","consumed":31,"exported":30,"source
Entitlement":null,"activeSubscription":true,"restrictedToUsername":null,"p
roductAttributes":
[{"productId":"RH0153936","name":"support_type","value":"L1-
L3","id":"8a878dcd3520d43501353f6f98f811de","updated":"2012-02-
02T18:59:32.000+0000","created":"2012-02-02T18:59:32.000+0000"}
...
```

entitlement_certificates/

entitlement_certificates/ contain PEM files with the base 64-encoded blob of the entitlement certificate for each subscription.

products/

products/ contains JSON file for every product included with the subscriptions. This contains detailed information about supported versions and content sets, dependencies, repositories, and other product-specific (but not necessarily subscription-specific) information.

For example, for part of the JSON file for one version with a basic Red Hat Enterprise Linux product:

```
...
{"name":"Red Hat Enterprise Linux Server","id":"69","attributes":
[{"name":"type","value":"SVC"}, {"name":"arch","value":"i386,ia64,x86_64"},
{"name":"name","value":"Red Hat Enterprise Linux
Server"}], "multiplier":1, "href":"/products/69", "productContent":
[{"content":{"name":"Red Hat Enterprise Linux 5 Server Beta (Source
ISOs)","id":"861","type":"file","vendor":"Red Hat","modifiedProductIds":
[], "contentUrl":"/content/beta/rhel/server/5/$releasever/$basearch/source/
iso","label":"rhel-5-server-beta-source-
isos","gpgUrl":"http://", "metadataExpire":86400, "requiredTags":"rhel-5-
server"}, "enabled":false}
...
```

rules/

rules/ contains a single JavaScript file which sets the functions that the application uses to interact with the backend Red Hat subscription management service.

4.4.2. Managing Manifests for Organizations

4.4.2.1. Attaching Subscriptions to Organizations

Attaching subscriptions to an organization sets the number of that type of subscription which the organization can attach to the systems it manages. (This is in contrast to a system, which attaches the subscription to itself for its own, local installed products.)

The **Attached Subscriptions** tab shows what subscriptions are currently attached to the organization. Clicking the **Attach a subscription** link shows all of the subscriptions that are available to the application organization, based on the overall account subscriptions.

To attach subscriptions to an organization:

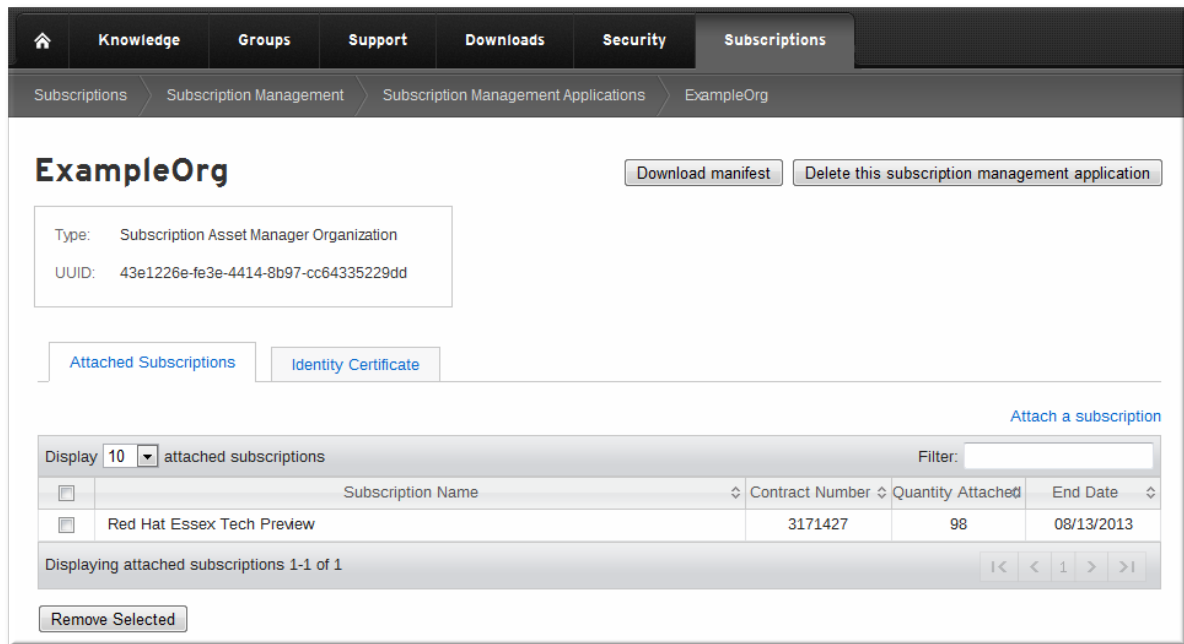
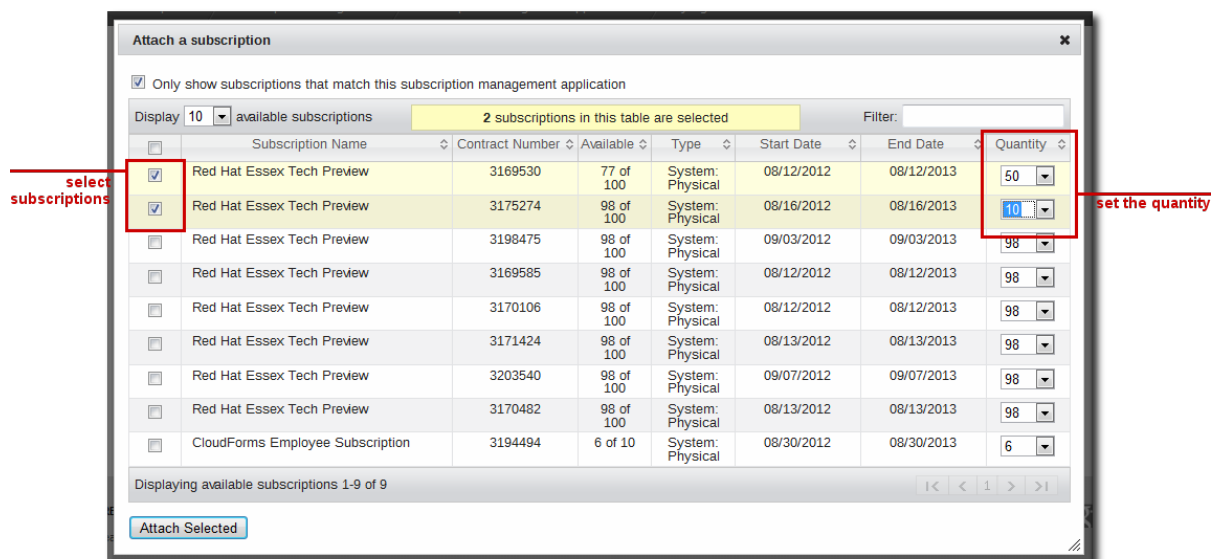
1. Log into the Customer Portal.
2. Open the **Subscriptions** tab, and select the **Overview** item under the **Subscription Management** menu area.
3. In the **Usage** area on the right, click the **Subscription Management** link.

The screenshot shows the 'Subscriptions' tab selected in the top navigation bar. The main content area is titled 'Overview' and is divided into two columns. The left column, 'Subscriptions', displays 'Current subscription totals': 1,645 active subscriptions, 35 subscriptions expiring in the next 120 days, and 17 recently expired subscriptions. Below this are links to 'Activate a subscription', 'Renew a subscription', 'Purchase subscriptions', and 'View web store order history'. The right column, 'Subscription Usage', explains that a subscription can be attached to one or more units and includes a warning that more subscriptions are being utilized than are available, with a 'More info' link. It also features a 'Subscription Utilization' chart and links to 'RHN Classic' and 'Subscription Management'.

4. In the **Subscription Management Applications** column, click the organization type.
5. Click the organization name in the application inventory.

The screenshot shows the 'Subscription Management Applications' page. The top navigation bar is the same as the previous screenshot. Below it, a breadcrumb trail shows 'Subscriptions > Subscription Management > Subscription Management Applications'. The main heading is 'Subscription Management Applications'. Below this are three tabs: 'All Subscription Management Applications', 'Subscription Asset Manager Organizations' (which is selected), and 'CloudForms'. A link 'Register a Subscription Asset Manager Organization' is visible. Below the tabs is a table with columns 'Name', 'Subscriptions Attached', and 'UUID'. The table contains two rows of data for 'ExampleOrg'. At the bottom, there is a status bar indicating 'Displaying Subscription Asset Manager Organizations 1-2 of 2 (filtered from 154 total Subscription Asset Manager Organizations)' and buttons for 'Delete Selected' and 'Export All as CSV'.

	Name	Subscriptions Attached	UUID
<input type="checkbox"/>	ExampleOrg	0	c3415e4d-2bb0-4f4b-bc32-a566af561934
<input type="checkbox"/>	ExampleOrg	98	43e1226e-fe3e-4414-8b97-cc64335229dd

6. Open the **Attached Subscriptions** tab.7. Click the **Attach a subscription** link to open the subscription selection window.8. Select the checkboxes by the subscriptions to attach and set the total quantity for the application organization in the **Quantity** column.

The list of available subscriptions provides three important pieces of information:

- The contract number for the purchase of the subscription, which is important for record keeping and tracking.
- The quantity still available for that subscription. Subscriptions are purchased in quantities; this number tells how many are still left of the total quantity purchased.
- The start and end dates of the subscription. This keeps you from attaching a subscription that may only be valid a few days before it expires or which are not yet active.

There should probably be a mix of subscriptions, with different end dates, attached to the organization to make it easier to renew subscriptions without having to update the manifest.



NOTE

The quantity defaults to be the total number of subscriptions available for that contract. Be aware of how many subscriptions are being attached to a single application organization so that the subscriptions can be attached appropriately among other units and subscription management applications.

- Click the **Attach selected** button in the lower left corner.

4.4.2.2. Downloading the Manifest

Once subscriptions are attached to a application organization, the complete list of subscriptions and products, including product certificates and entitlement certificates, are bundled together in a single *manifest*. The manifest is essentially a master list of everything that the application organization requires to handle local subscription management services.

The manifest can be downloaded from the application organization's details page simply by clicking the **Download manifest** button. This saves the **manifest.zip** archive to the local filesystem, so it can then be uploaded to Subscription Asset Manager or Satellite 6.

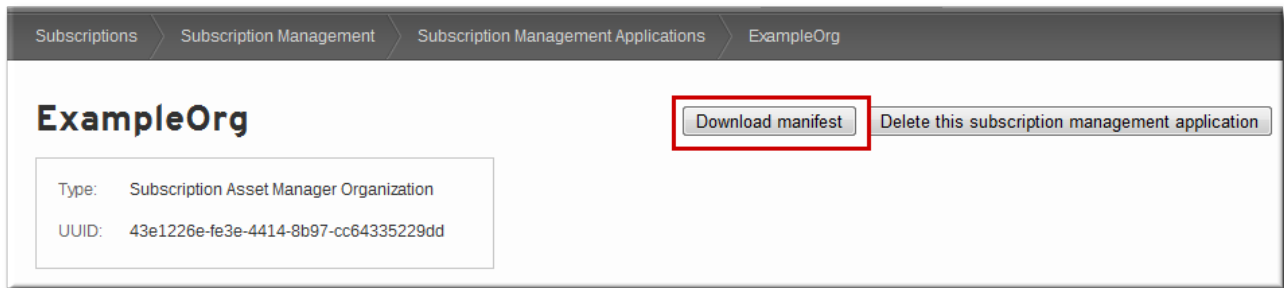
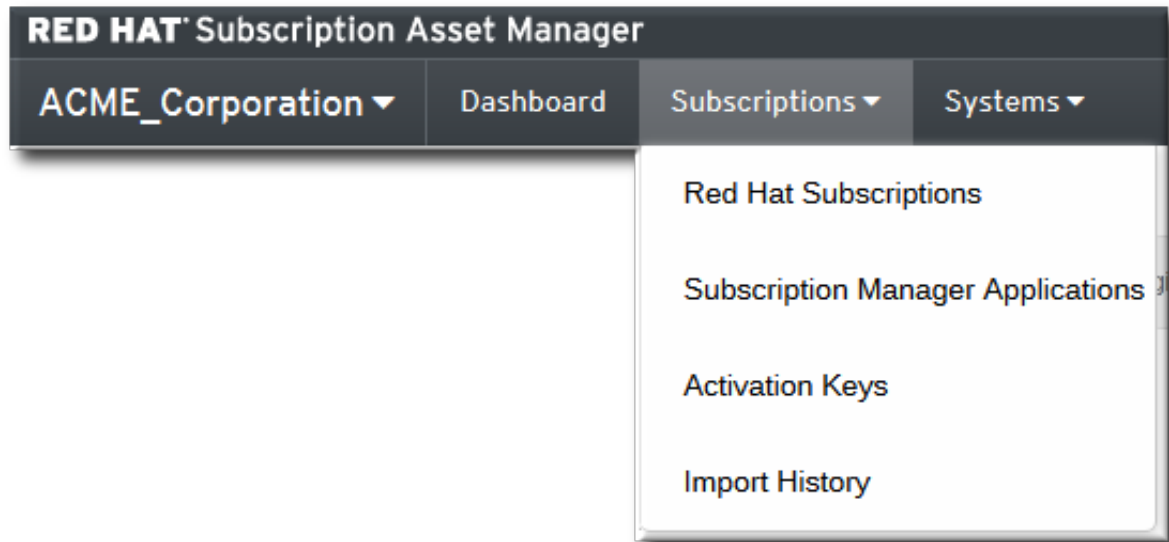


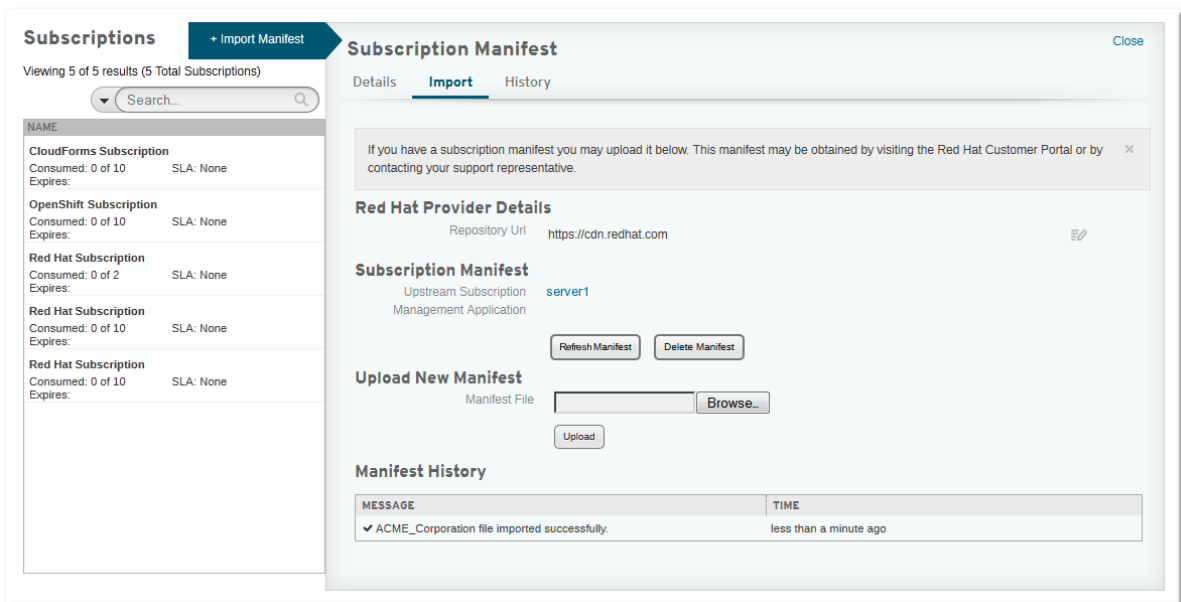
Figure 4.6. Downloading the Application Organization Manifest

4.4.2.3. Uploading a Subscription Manifest

- Open the Subscription Asset Manager UI, and log in as an administrator for the organization associated with the manifest.
- If necessary, expand the organizations menu in the upper left corner, and select the appropriate organization.
- Hover over the **Subscriptions** tab, and click the **Red Hat Subscriptions** item.



4. Click the **+Import Manifest** link.



If a manifest has been loaded previously for the organization, then some details for the existing manifest are filled in the details page.

5. Click the **Browse** button and navigate to the downloaded manifest file.
6. Click the **Upload** button.

4.4.2.4. Refreshing Manifests for Organizations

If the organization needs to change its subscriptions — by altering quantities, adding products, or renewing subscriptions — this is done by editing the subscriptions attached to the organization in Customer Portal Subscription Management.

Manifests are changed in two locations: first, the manifest itself is updated in the Customer Portal and it is then refreshed on the Subscription Asset Manager organization.



IMPORTANT

Do no attempt to update the on-premise organization entry by creating a new organization in Customer Portal Subscription Management. Change the subscriptions attached to the existing organization in Customer Portal Subscription Management, and then have the on-premise organization use the updated manifest.

1. Update the manifest in the Customer Portal.

1. Log into the Customer Portal.
2. Open the **Subscriptions** tab, hover over the **Subscription Management** item, and select the **Subscription Management Applications** item.
3. All registered applications are listed. Search for the specific application organization or switch to the product-specific tab, and search for the system there.

The screenshot shows the Red Hat Customer Portal interface. The top navigation bar includes the Red Hat logo, 'CUSTOMER PORTAL', a search bar, and user information for 'John Smith'. The main navigation menu has tabs for Products, Support, Downloads, Security, and Subscriptions. The Subscriptions tab is active, and the 'Subscription Management Applications' sub-tab is selected. The page title is 'Subscription Management Applications'. Below the title, there are tabs for 'All Subscription Management Applications', 'Subscription Asset Manager Organizations' (which is selected), 'CloudForms', and 'Satellite'. A link 'Register a Subscription Asset Manager Organization' is visible. A table titled 'Subscription Asset Manager Organizations' is displayed, showing a list of organizations with their names, the number of subscriptions attached (all 0), and their UUIDs. The table has a 'Display' dropdown set to 10 and a 'Filter' input field. The footer of the table indicates 'Displaying Subscription Asset Manager Organizations 1-10 of 26'.

Name	Subscriptions Attached	UUID
server-1.example.com	0	8f1631d4-a983-4713-9852-73eaf88450f5
qe.engineering.example.com	0	66cb5b3c-b6ba-4d47-b99c-de6bb48834b0
prep.staging.example.com	0	1c878310-7e64-4c9f-9fb4-4bace4d5c21e
prod-1.example.com	0	19443e63-11e2-42f6-8d1e-8efd745204cd
prod-2.example.com	0	c852e6ad-3d75-4105-b1d6-7a6bae52fc06
support.example.com	0	2ca8dc64-2a5e-405f-84ac-3f552728824f
desktop1	0	2890781f-b304-4956-b9b7-677306600bec
qe-lab-west.engineering.example.com	0	e58c3036-51ba-4755-9e40-d563501409ba
qe-lab-east.engineering.example.com	0	e328fc01-cdc4-447e-8f58-a462766f19bf
engineering.example.com	0	07a3131a-56a6-427d-bef0-bb2f690d4b97

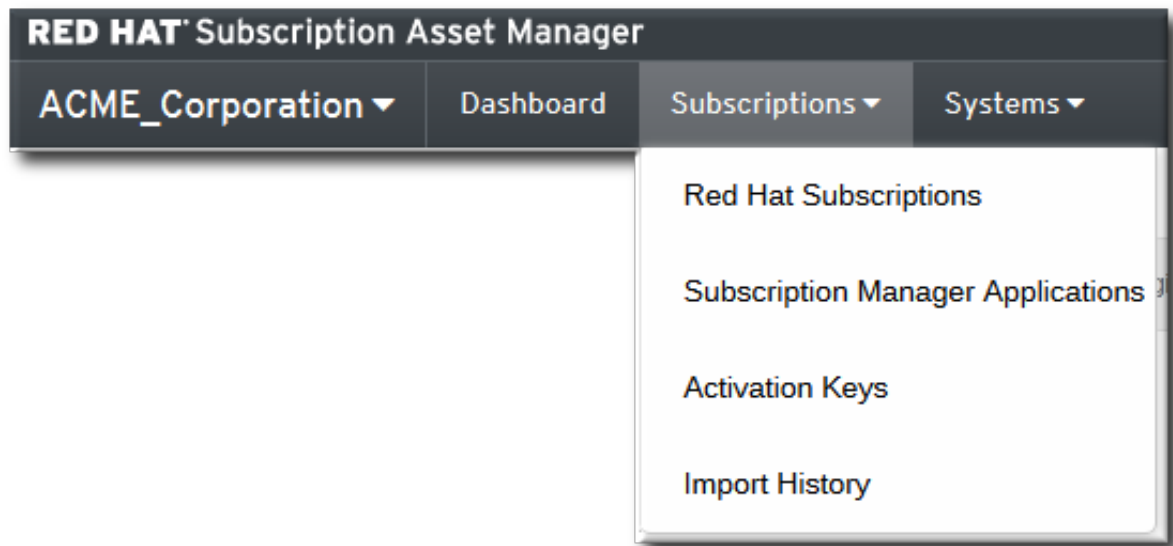
4. Click the organization name in the application inventory.
5. Open the **Attached Subscriptions** tab.
6. Delete any previous subscriptions which need to be updated. Select the checkbox by the subscription, and click the **Remove Selected** button.

A subscription quantity attached to a subscription management application organization cannot be changed directly. If additional numbers need to be added or removed from an attached subscription, delete the original assignment and then attach the subscription with the new quantity.

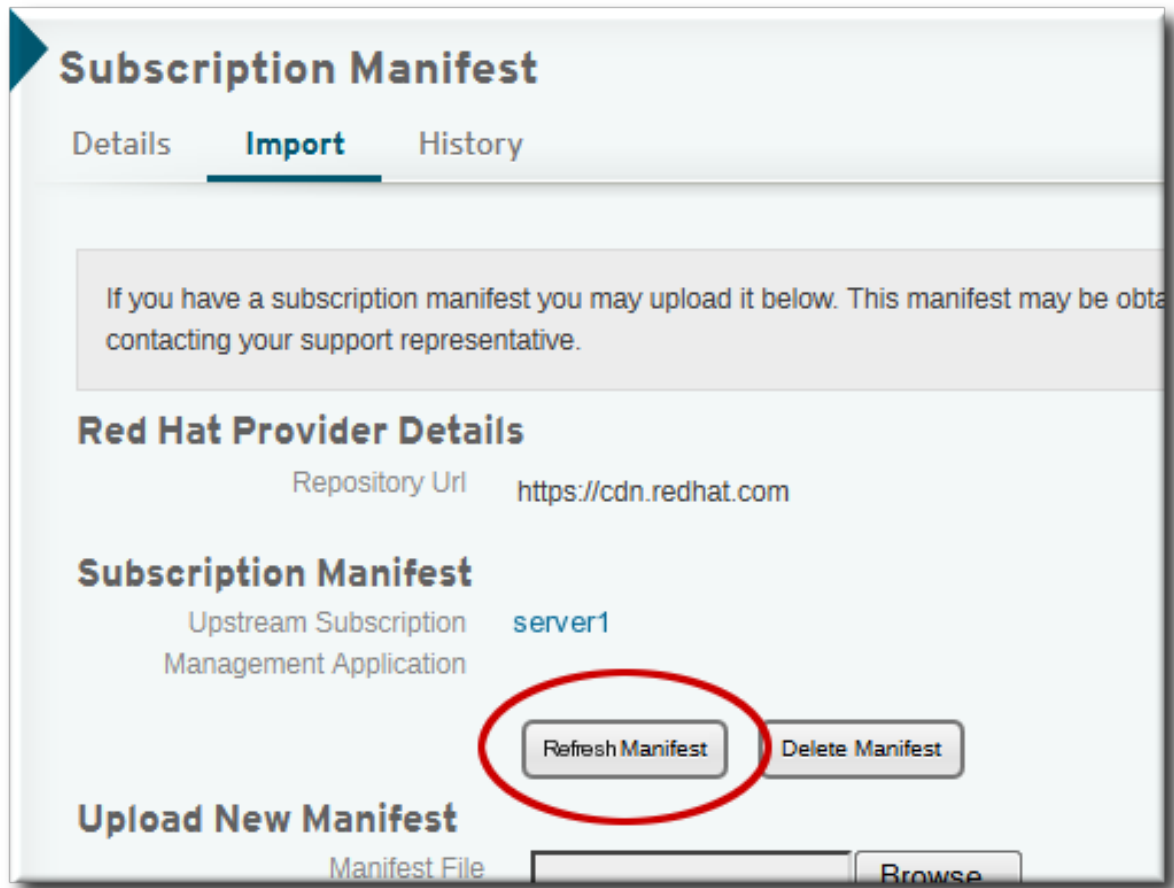
For example, if your subscription bloc has a quantity of 30 and it should increase to 35, you

can delete the old bloc and add a new one with a quantity of 35; that leaves you with one subscription and a quantity of 35. Alternatively, you can simply add a new bloc with a quantity of 5; that results in two separate subscription entries, one with a quantity of 30 and one with a quantity of 5.

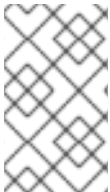
7. Add any new subscriptions.
2. Open the Subscription Asset Manager UI and log in as the organization administrator.
3. If necessary, switch to the appropriate organization.
4. Hover over the **Subscriptions** tab, and click the **Red Hat Subscriptions** item.



5. In the manifest's **Import** area, click the **Refresh Manifest** button.



The updated manifest is pulled directly from the Customer Portal.



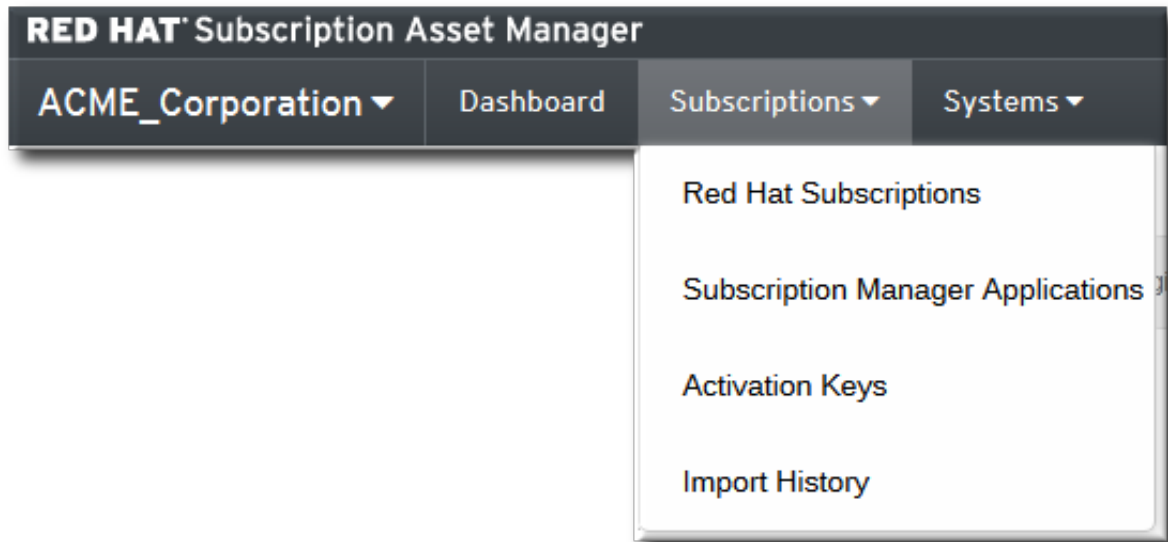
NOTE

In some cases, there may not be a **Refresh Manifest** button. Then, the updated manifest can be uploaded like a new manifest file, as in [Section 4.4.2.3, “Uploading a Subscription Manifest”](#).

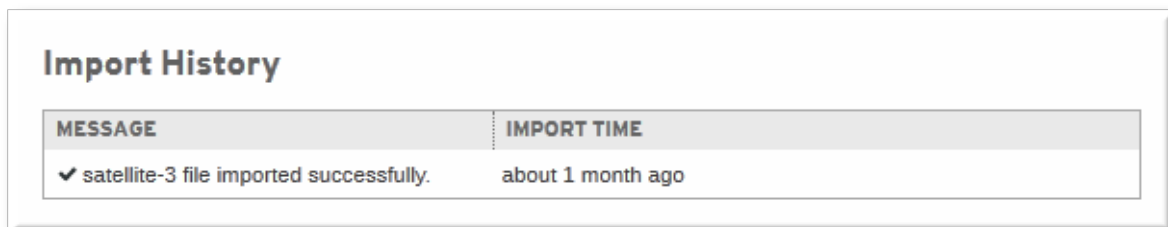
4.4.2.5. Viewing the Manifest Import History

Every time a manifest is imported or refreshed for an organization, the operation is logged. This can help track when updated subscriptions are made available to systems within the organization.

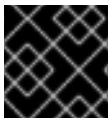
1. Open the Subscription Asset Manager UI and log in as the organization administrator.
2. If necessary, switch to the appropriate organization.
3. Hover over the **Subscriptions** tab, and click the **Import History** item.



4. Every import operation is listed in the history.



4.4.3. Managing Manifests for Distributors (TECH PREVIEW)



IMPORTANT

This is a tech preview feature in Subscription Asset Manager 1.4.

All subscription assignments and inventories originate in the Customer Portal. When a new Subscription Asset Manager organization is created a subset of those subscriptions must be attached to the organization. The relationship is between the Portal (main corporate account) and the Subscription Asset Manager organization (division).

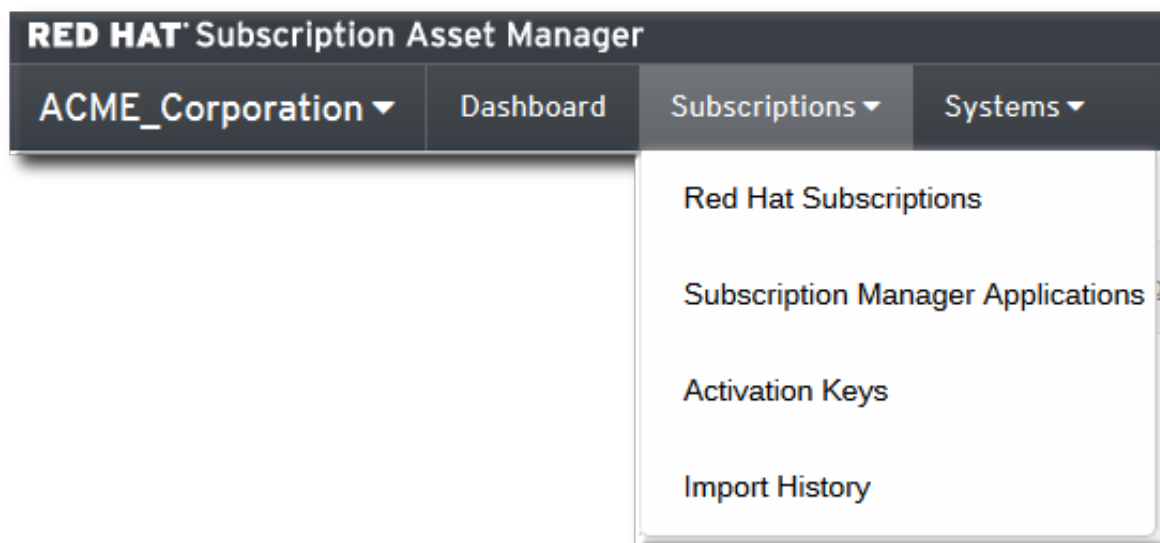
The Portal and the Subscription Asset Manager organization communicate over a secure but external network connection. There may be some instances where a particular Subscription Asset Manager organization cannot or should not be accessible over a public network or a business reason why a Subscription Asset Manager organization should be subdivided into child organizations which are still visible to one another.

Subscription Asset Manager allows *distributors* to be created for a Subscription Asset Manager organization. This is essentially an organization within an organization. The parent organization then attaches subscriptions to the child organization and creates a manifest locally.

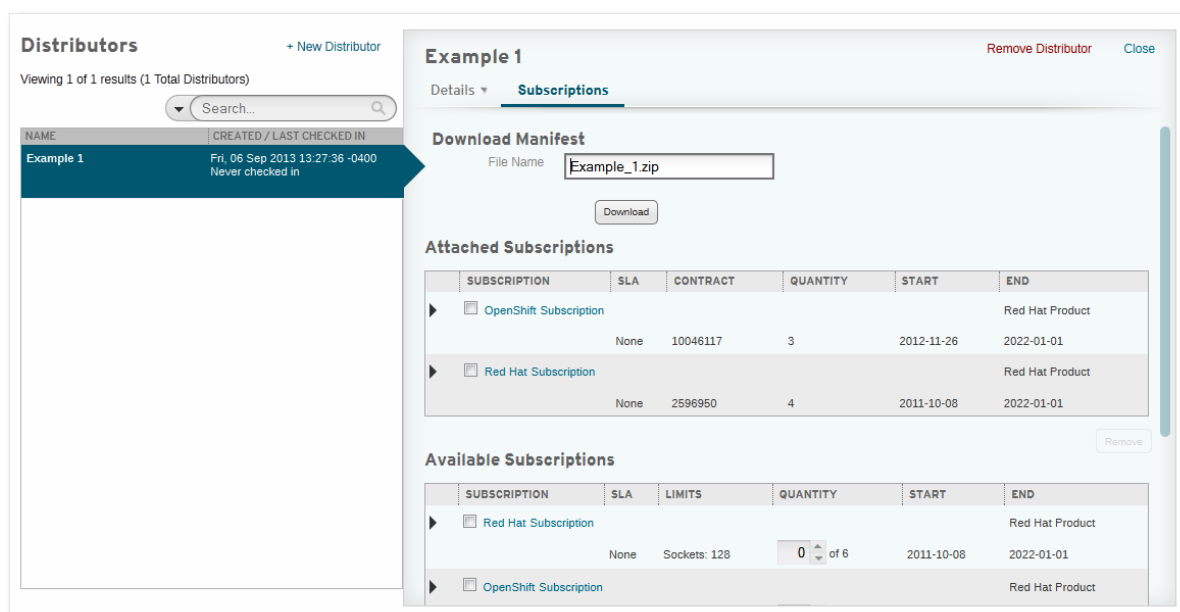
To create manifests for distributors:

1. Open the Subscription Asset Manager UI and log in as the organization administrator.
2. If necessary, switch to the appropriate parent organization.

3. Hover over the **Subscriptions** tab, and click the **Subscription Manager Applications** item.



4. Select the distributor from the list in the left column.
5. In the distributor's **Subscriptions** tab, select the subscriptions to add in the **Available Subscriptions** area.

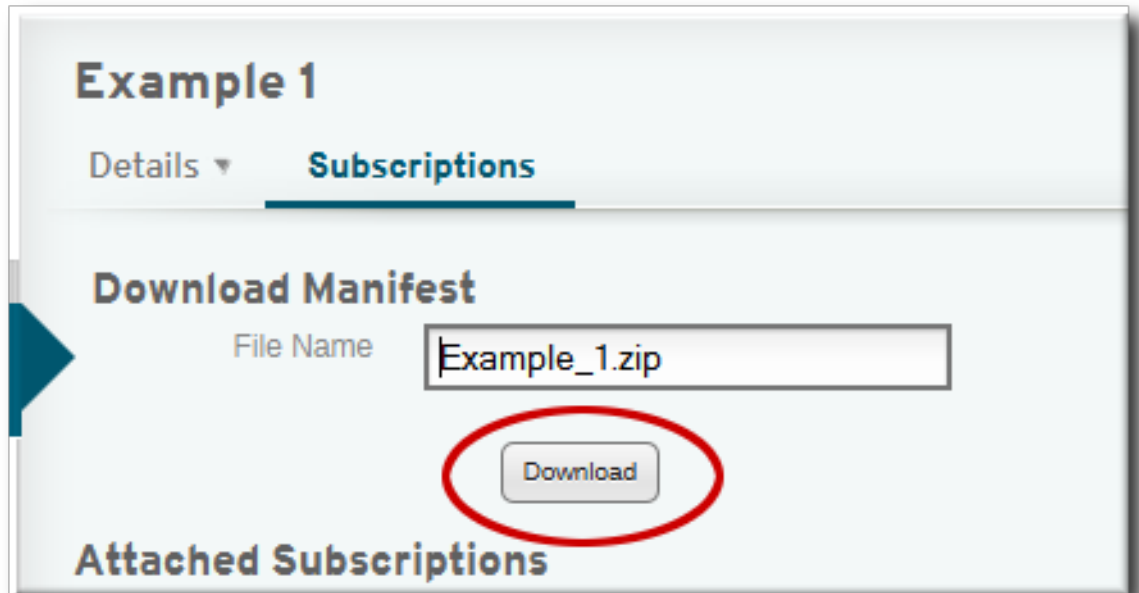


Be sure to set the appropriate quantity of subscriptions for each product selected. The quantity is the total number of subscriptions of that type available to the child organization.

6. Scroll down, and click the **Attach** at the bottom of the window.

Attaching subscriptions automatically updates the child organization's manifest.

7. At the top of the **Subscriptions** tab for the distributor, click the **Download** button to export the manifest.



8. Open the web UI for the child Subscription Asset Manager organization.
9. Upload the manifest to the child organization, as described in [Section 4.4.2.3, “Uploading a Subscription Manifest”](#).

CHAPTER 5. MANAGING SYSTEMS AND SUBSCRIPTIONS

Systems use subscriptions to define the software packages they are able to install. Systems must have a current subscription in order to be able to download updates.

5.1. ABOUT SUBSCRIPTIONS ON SYSTEMS

Assigning a subscription to a system gives the system the ability to install and update any Red Hat product in that subscription. A subscription is a list of all of the products, in all variations, that were purchased at one time, and it defines both the products and the number of times that subscription can be used. When one of those licenses is associated with a system, that subscription is *attached* to the system.

Subscriptions available for an organization are defined through the organization manifests ([Section 4.4, “Importing and Maintaining Manifests”](#)). For systems, subscriptions can be added in several different ways:

- By manually adding and removing subscriptions
- By autoattaching subscriptions based on the installed products and system characteristics
- By registering the system with activation keys to pre-attach subscriptions

5.1.1. About Relationships Between Subscriptions and Systems

5.1.1.1. Interactions with Subscriptions, Products, and Systems

Products on a system have relationships, dependencies, and conflicts between each other. Likewise, subscriptions have relationships that parallel the relationships of the software it represents. Some subscriptions allow virtual guests, some require other subscriptions, some conflict with other subscriptions.

Subscriptions define the relationships between installed products and each other and the systems on which those products are installed. Likewise, subscriptions can also define relationships between systems and how they interact within an environment. This is particularly apparent with virtual environments, where subscriptions can define different relationships for physical hosts and virtual guests, but there are other ways that systems can interact, such as data centers and cloud infrastructures. Subscriptions are a part of those meta relationships.

Using subscriptions to define these relationships introduces a lot of flexibility in how products and systems interact:

- Associate a single quantity of a product with a single system (which is the most common relationship).
- Restrict one product so that it cannot be installed on the same system as a specific, different product.
- Keep a system on a consistent service level. Each subscription includes a definition for what service level (e.g., standard or premium) the product has. Subscription clients first try to assign subscriptions of the same service level (and this can be enforced) so that the system has consistent support levels.
- Allow virtual guests to inherit some subscriptions from their host.

- Allow some hosts to have unlimited guests for a data center deployment.
- Allow a single “subscription” to be broken across multiple systems. This works in something like Red Hat Cloud Infrastructure, where a single purchase actually covers four products — Red Hat Enterprise Linux, Red Hat OpenStack, Red Hat Virtualization, and Satellite 6 — and those products each have their own subscription which can be used on different systems to create the stack.
- Stack or combine subscriptions of the same type to cover a system.

5.1.1.2. Counting Subscriptions

Part of the subscription service inventory is keeping track of subscriptions – not just what subscriptions are purchased but how many of those subscriptions are available.

When a subscription is first purchased, it defines the quantity of times that the subscription can be used. The subscription count is based on a certain element of the underlying system, most commonly its socket count (but it can be something else, such as the number of cores, depending on the specific subscription). The element of a system or software which is directly covered by a subscription is called an instance.

For example, for the subscription for Red Hat Enterprise Linux for 2 sockets, the product is Red Hat Enterprise Linux and the attribute is a physical socket pair. The socket pair is the instance.

A single subscription quantity is usually tied to a single socket pair (or other attribute). A system with eight sockets, then, requires more subscription quantities to cover its socket count than a four socket system. (This is called stacking.)

This simplistic arrangement, however, does not apply to all subscriptions.

Starting in October 2013, Red Hat began introducing other types of subscription relationships, such as:

- Multiple products with a single subscription (Red Hat Cloud Infrastructure)
- Inheritable subscriptions
- Data center subscriptions, which allow unlimited virtual guests (and only the host requires a specific subscription)

Additionally, the 2013 subscription changes altered how virtual guests are handled in subscriptions. There used to be subscriptions for physical systems and then different subscriptions for virtual guests. In the current subscription model, the same subscription is used for both physical and virtual systems – but the quantity used is different, depending on whether it is a physical system or a virtual one.

As stated previously, a single subscription quantity is used per socket pair on a physical system. A virtual guest counts as a single socket, not a socket pair – so it is essentially half of a subscription quantity. When virtual guests are added to the inventory, the total number of available subscriptions is multiplied by two (the instance multiplier). This allows the subscription count to stay in whole numbers, even with virtual guests taking only a “half” quantity.

However, with some subscriptions counts multiplied by two; data center virtual guests not consuming any individual subscriptions; some subscriptions (Cloud Infrastructure) relating to multiple products installed on different systems; *and* older, pre-2013-style subscriptions all in the same environment — the actual counts listed in the subscription utilization pages or subscription management tools may not appear to reflect the quantities purchased in the contract. The fundamental counts are the same; most of the differences reflect changes to keep the count whole or new, more flexible subscription types.

5.2. REGISTERING A SYSTEM

1. Install the configuration RPM to configure Red Hat Subscription Manager to point to the Subscription Asset Manager instance. For example:

```
[root@server ~]# rpm -ivh http://sam.example.com/pub/candlepin-cert-consumer-latest.noarch.rpm
```

2. Register the system using the **subscription-manager** command. The organization name is required; if the username or password are not passed with the command, then the command prompts for them. Using **--auto-attach** to apply subscriptions automatically is not required, but it simplifies setting up the new system.

```
[root@server ~]# subscription-manager register --username=jsmith --password=secret --org="IT Dept" --auto-attach
```

This command must be run as root.

5.3. USING SYSTEM GROUPS

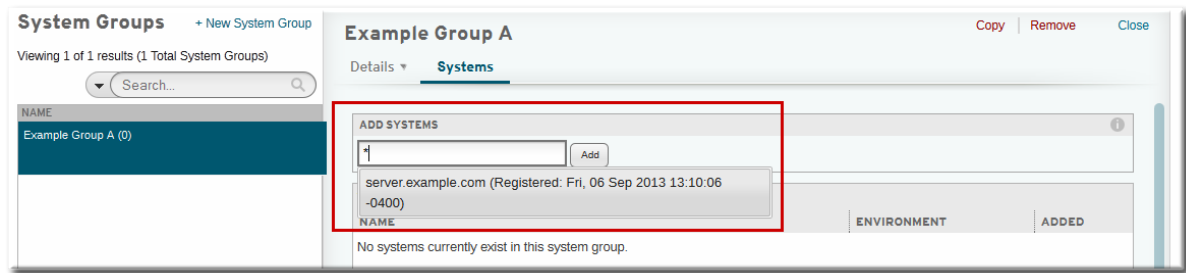
System groups are a way of providing structure for systems within an organization. System groups can be associated with access controls to control what users can access given systems and can be associated with activation keys to help define initial configuration for new systems.

1. Hover over the **Systems** tab, and select the **System Groups** option.
2. In the left column of the groups page, click the **+ New System Group** link.
3. Enter the name for the group and, optionally, a description and system limit. The system limit sets a limit on how many systems can belong to the group; the default is for the group membership to be unlimited.

The screenshot shows the 'System Groups' management page. On the left, there's a sidebar with a search bar and a list of existing groups: 'Example Group A (1)' and 'Example Group B (3)'. The main content area is titled 'New System Group' and features a form for creating a new group. The form includes a 'Name' field, a 'Description' text area, and a 'System Limit' section with a radio button for 'Unlimited' (which is selected). A 'Save' button is located at the bottom right of the form.

4. Click the **Save** button.
5. Clicking the save button opens the **Systems** tab for the new group. Search for the systems to add; using a wild card (an asterisk, *) lists all systems.

Select the systems to add from the drop-down list, and click the **Add** button.

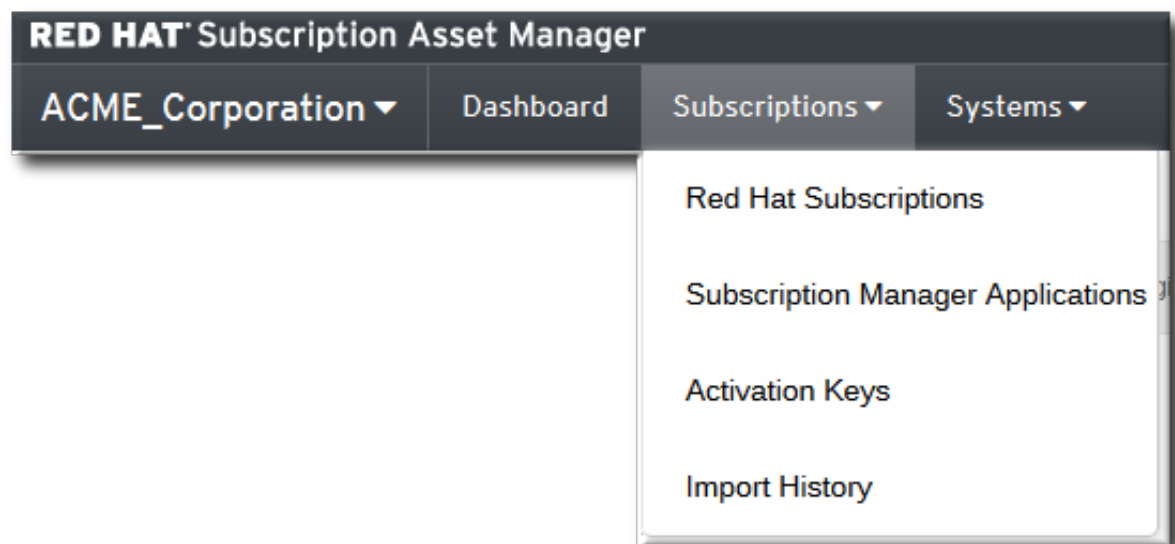


5.4. VIEWING SUBSCRIPTIONS FOR THE ORGANIZATION

Each organization has a single manifest, which is created on the Customer Portal and then uploaded to the organization. The manifest lists all of the subscriptions and quantities (without any adjustments for virtual systems) for that manifest.

Each subscription is listed for the organization, and a subscription may include one or multiple products.

1. Open the Subscription Asset Manager UI and log in as the organization administrator.
2. If necessary, switch to the appropriate organization.
3. Hover over the **Subscriptions** tab, and click the **Red Hat Subscriptions** item.



4. Click the name of the subscription in the left column to view its details.

The details page includes its support level, account number, and system attributes used for the products.

The screenshot shows the 'Subscriptions' tab in the Red Hat Subscription Asset Manager. On the left, a list of subscriptions is displayed, with 'Red Hat Subscription' highlighted. The main panel shows the details for this subscription, including account number, support level, support type, architecture, type, limits, multi-entitlement, and instance-based status. A list of provided products is also shown at the bottom.

Red Hat Subscription	
Details	Units
Account Number	477931
Support Level	None
Support Type	None
Arch	x86_64, ia64, ppc, ppc64, s390x, s390
Type	Physical or Virtual
Limits	Sockets: 128
Multi-entitlement	No
Instance-based	No

Provided Products

- Red Hat Beta
- Red Hat Enterprise Linux Resilient Storage (for RHEL Server)
- Red Hat Enterprise Linux High Availability (for RHEL Server)
- Red Hat Enterprise Linux Server
- Red Hat Enterprise Linux Workstation
- Red Hat Enterprise Linux Load Balancer (for RHEL Server)

The **Products** tab lists every product included in the subscription; expanding the product displays information about its repository location.

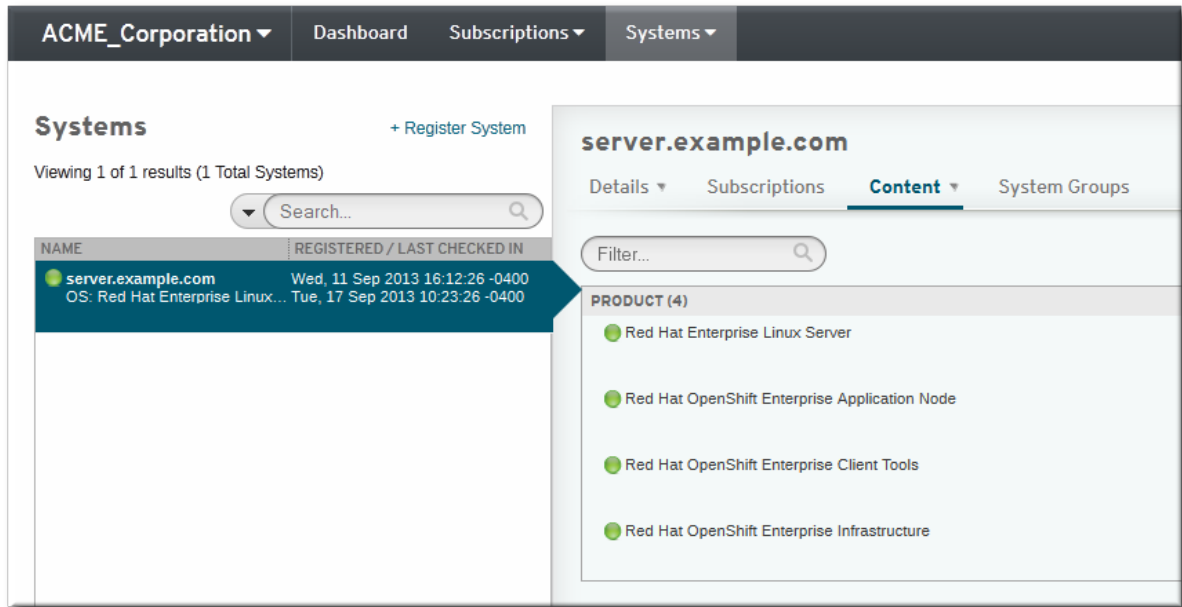
The screenshot shows the 'Products' tab for an 'OpenShift Subscription'. The left sidebar lists subscriptions, with 'OpenShift Subscription' highlighted. The main panel displays the 'PROVIDED PRODUCT DETAILS' for the 'JBoss Enterprise Application Platform'. It lists the content download URL, GPG key URL, repository type, and whether the product is enabled by default.

OpenShift Subscription	
Details	Units
PROVIDED PRODUCT DETAILS	
Red Hat OpenShift Enterprise JBoss EAP add-on	
JBoss Enterprise Application Platform	
Content Download URL:	/content/dist/rhel/server/6/\$releasever/\$basearch/jeap/6/source/SRPMS
GPG Key URL:	file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Repo Type:	yum
Enabled by Default:	false
Content Download URL:	/content/dist/rhel/server/6/\$releasever/\$basearch/jeap/6/os
GPG Key URL:	file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Repo Type:	yum
Enabled by Default:	true
Content Download URL:	/content/dist/rhel/server/5/\$releasever/\$basearch/jeap/4.3.0/os
GPG Key URL:	file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Repo Type:	yum
Enabled by Default:	true
Content Download URL:	/content/dist/rhel/server/5/\$releasever/\$basearch/jeap/5/os
GPG Key URL:	file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Repo Type:	yum
Enabled by Default:	true
Content Download URL:	/content/dist/rhel/server/5/\$releasever/\$basearch/jeap/4.3.0/source/SRPMS
GPG Key URL:	file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
Repo Type:	yum
Enabled by Default:	false

5.5. VIEWING INSTALLED PRODUCTS FOR A SYSTEM

All of the installed products for a system, and the subscription status for each individual product, are listed on the **Content** tab for the system.

1. Hover over the **Systems** tab in the top menu and select the **All** item.
2. Select the name of the system from the column on the left.
3. Open the **Content** tab.
4. Every product currently installed on the system is listed, with the subscription status for each displayed.



5.6. MANUALLY MANAGING SUBSCRIPTIONS

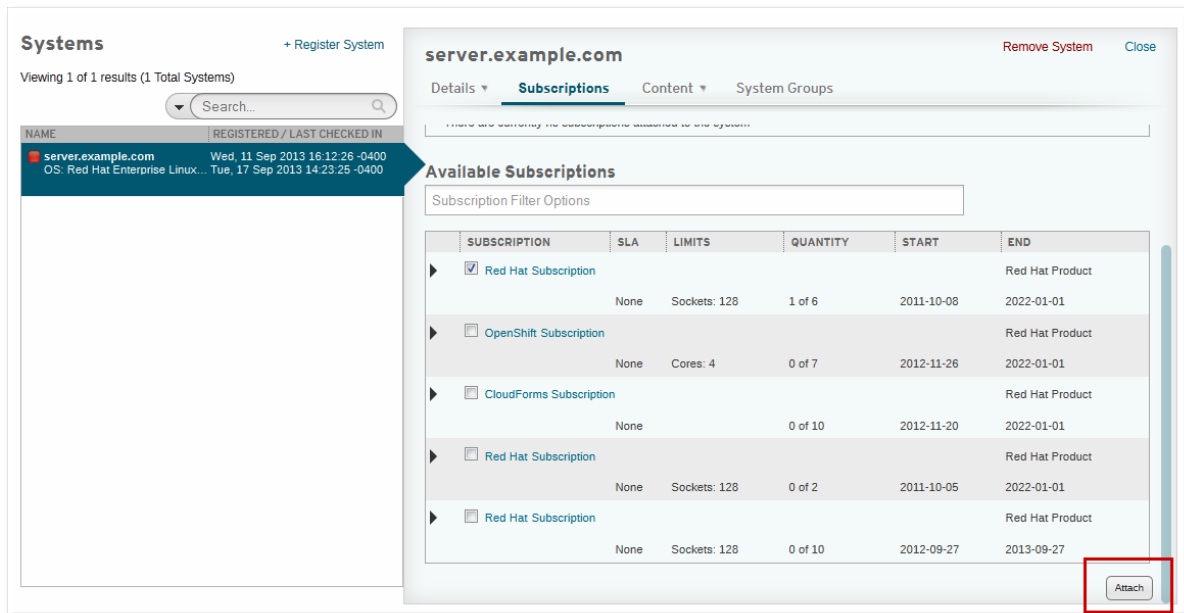
5.6.1. Attaching Subscriptions to a System



NOTE

It takes up to 4 hours for the subscription status on the **RHSM** client system to be updated. If the subscription status needs to be updated immediately, run the **subscription-manager refresh** command from the command line interface.

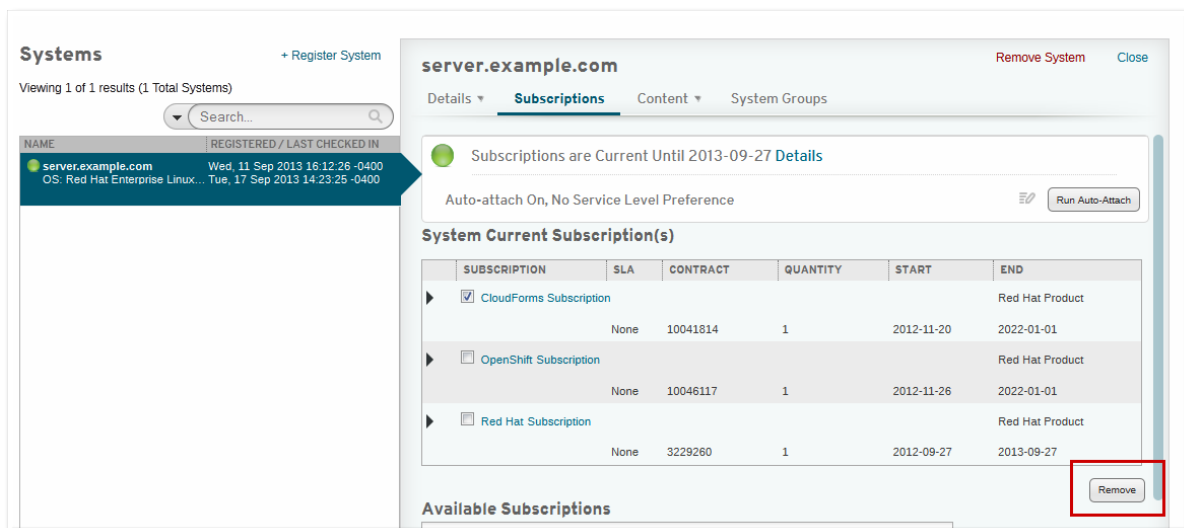
1. Hover over the **Systems** tab in the top menu and select the **All** item.
2. Select the name of the system from the column on the left.
3. Open the **Subscriptions** tab.
4. In the **Available Subscriptions** list, select the checkboxes by the names of the subscriptions to attach to the system.



5. Click the **Attach** button.

5.6.2. Removing Subscriptions

1. Hover over the **Systems** tab in the top menu and select the **All** item.
2. Select the name of the system from the column on the left.
3. Open the **Subscriptions** tab.
4. In the **System Current Subscriptions** list, select the checkbox by the name of the subscription to remove.



5. Click the **Remove** button.

5.7. CONFIGURING AUTOATTACH PREFERENCES FOR A SYSTEM

Autoattaching and updating subscriptions selects what subscriptions to attach to a system based on a variety of criteria, including current installed products, hardware, and architecture. Most of the factors used for selecting the best-matched subscriptions are based on the characteristics of the system, but it is

possible to include characteristics of the subscriptions.

Part of a subscription is recognizing the service level for a product on a given system. That service level can be used as a criterion for selecting a subscription to attach to a system.

Red Hat service levels are defined in the contract; a summary of production support levels is available at <https://access.redhat.com/support/offerings/production/sla.html>.

An account can have multiple levels of support available, even for the same product. The support level for a given system can be configured so that the appropriate level of support is available. A production system usually has a premium support level since it is a business critical system, while a development system may have standard support or be self-supported.

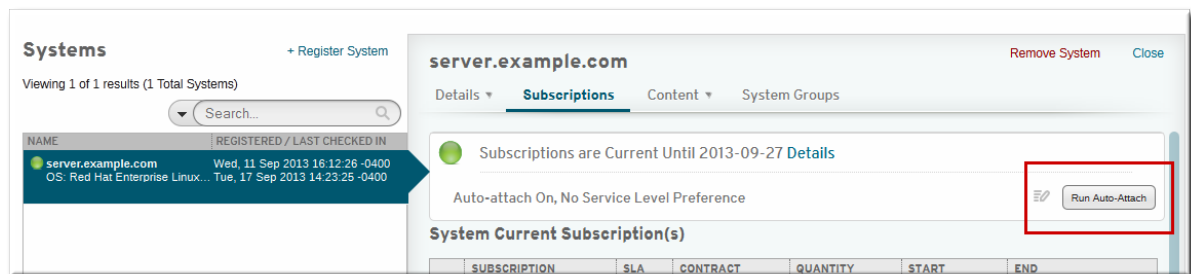
An organization can set a default service level which all systems use when autoattaching subscriptions. A local, system-level preference can be set to override that organizational default, if desired.



NOTE

By default, the highest available level of support is selected for the subscription and system.

1. Hover over the **Systems** tab in the top menu and select the **All** item.
2. Select the name of the system from the column on the left.
3. Open the **Subscriptions** tab.
4. Click the edit icon in the top box to change the autoattach settings.



5. Select the appropriate autoattach setting.

The options in the list depend on the available support levels in the subscriptions for the organization. From a high level, the options are:

- Enable autoattach and use a specific system-level preference for the support level.
- Enable autoattach and use the default support level preference for the organization.
- Disable autoattach and set the support level preference to either a system-level setting or the organizational default. (In either case, the preference is not used since autoattach is disabled.)

6. Click the **Save** button.

5.8. RUNNING AUTOATTACH OPERATIONS MANUALLY

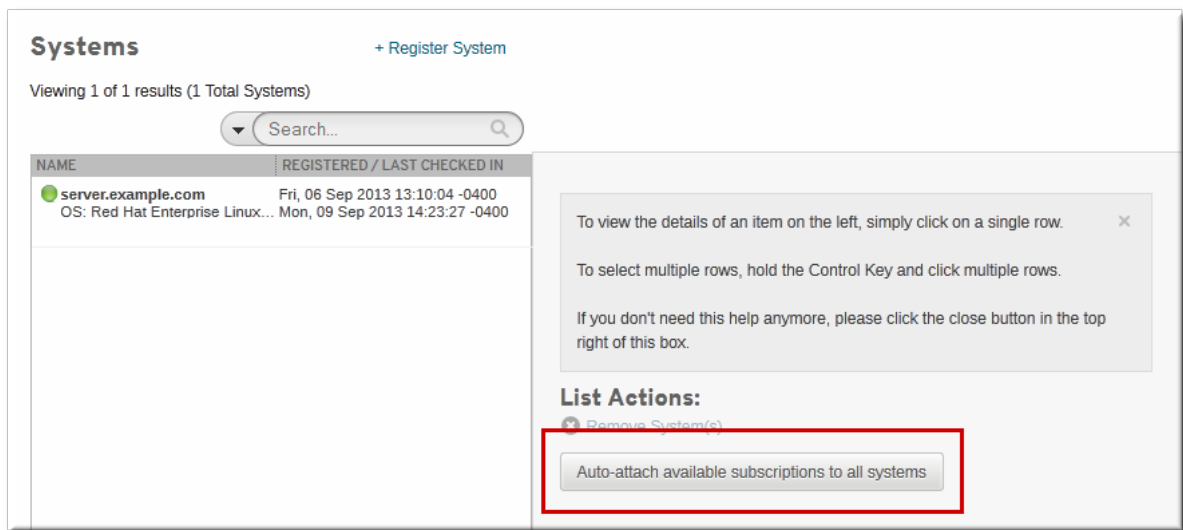
The local system can run a job every four hours to update subscriptions automatically, and attach or remove any subscriptions depending on the installed products and subscription status.

It is possible to run an asynchronous autoattach operation on a single system or for every system, to update all subscriptions on the system immediately.

5.8.1. Running Autoattach on All Systems

To update subscriptions for all systems:

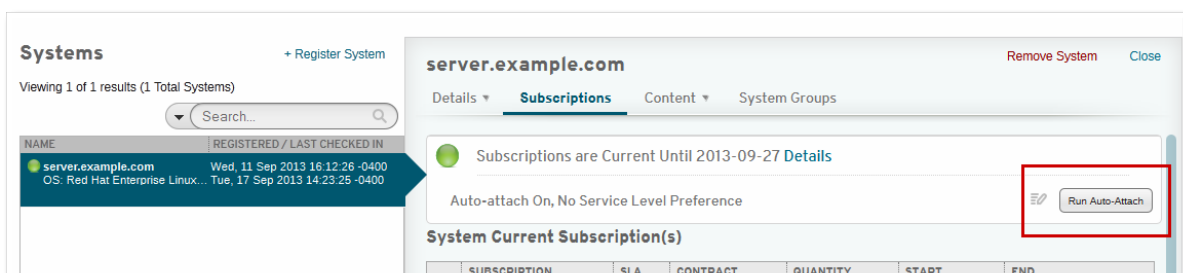
1. Hover over the **Systems** tab in the top menu and select the **All** item.
2. In the main page for systems, click the **Auto-attach available subscriptions to all systems** button.



5.8.2. Running Autoattach on a Single System

To update subscriptions for all systems:

1. Hover over the **Systems** tab in the top menu and select the **All** item.
2. In the search box on the left of the systems column, search for the specific system.
3. Click the name of the system in the column on the left.
4. Click the **Subscriptions** tab for the system.
5. In the upper right of the subscriptions area, click the **Run Auto-Attach** button.



CHAPTER 6. USING ACTIVATION KEYS

Activation keys are a way to preconfigure subscriptions for a system before it is registered (or, when used with provisioning systems like kickstart, before the system is even created). This grants a great deal of flexibility and control to administrators over what subscriptions to attach to new systems, while simplifying the registration process for users.

6.1. ABOUT ACTIVATION KEYS

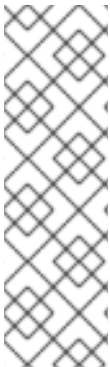
Activation keys provide a way to apply predetermined subscription configuration to a system, specifically in two areas:

- Attaching predefined subscriptions
- Assigning the system to identified system groups

Activation keys provide more control over the initial setup of a system over autoattaching when a system is registered because autoattaching only evaluates *what is currently installed* on a system. An activation key can contain and attach a subscription for a product which is not yet installed.

Even more, using activation keys allows administrators to plan what products will be (initially) installed and subscribed for systems *which do not yet exist*.

The same activation key can be applied to multiple systems, as long as there are sufficient quantities on the subscriptions themselves.



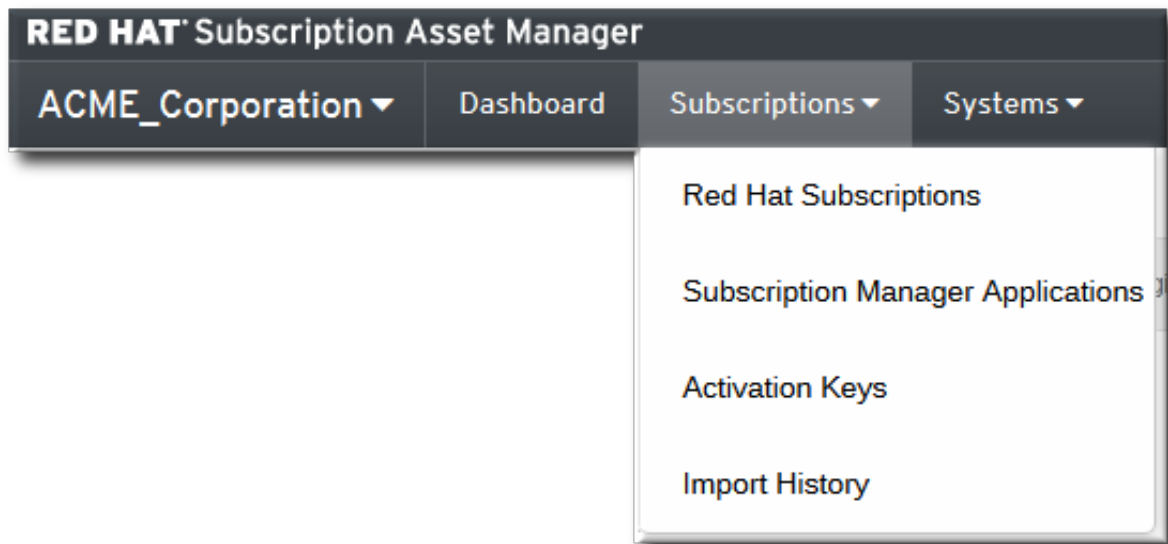
NOTE

Activation keys only set the *initial* configuration for a system. Once it is registered to an organization, all of the products and subscriptions which that organization possesses can be attached to the system.

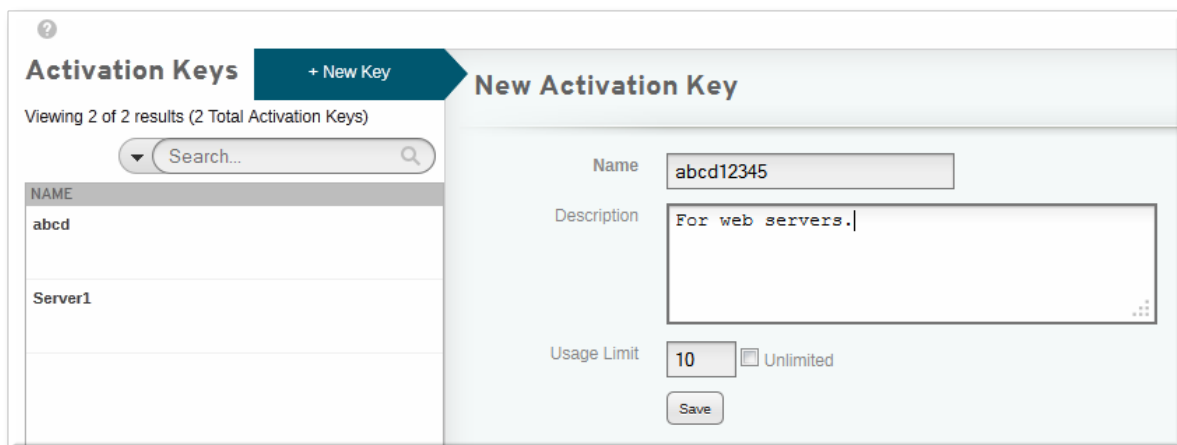
For example, if an activation key contains a subscription to Red Hat Enterprise Linux, and then the organization contains an additional subscription for JBoss Enterprise Application Platform (EAP), the system can install EAP and have a subscription attached for it, even though its initial activation key did not cover it.

6.2. CREATING AN ACTIVATION KEY

1. Log into the Subscription Asset Manager UI.
2. Hover over the **Subscriptions** item in the top menu, and click the **Activation Keys** item.



3. Click the **+ New Key** button in the upper left column.
4. Enter the string for the key in the **Name** field. This is the string that will be used by clients to redeem the subscriptions at registration.



Optionally, enter a description for the key.

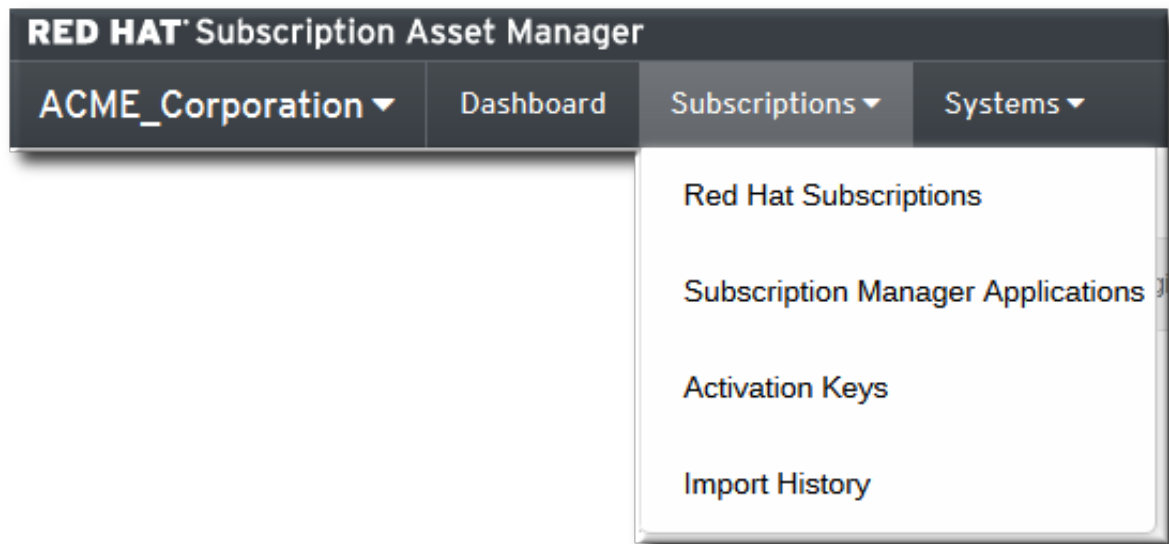
5. Click the **Save** button.

Once the key is created, assign subscriptions and, optionally, system groups to it.

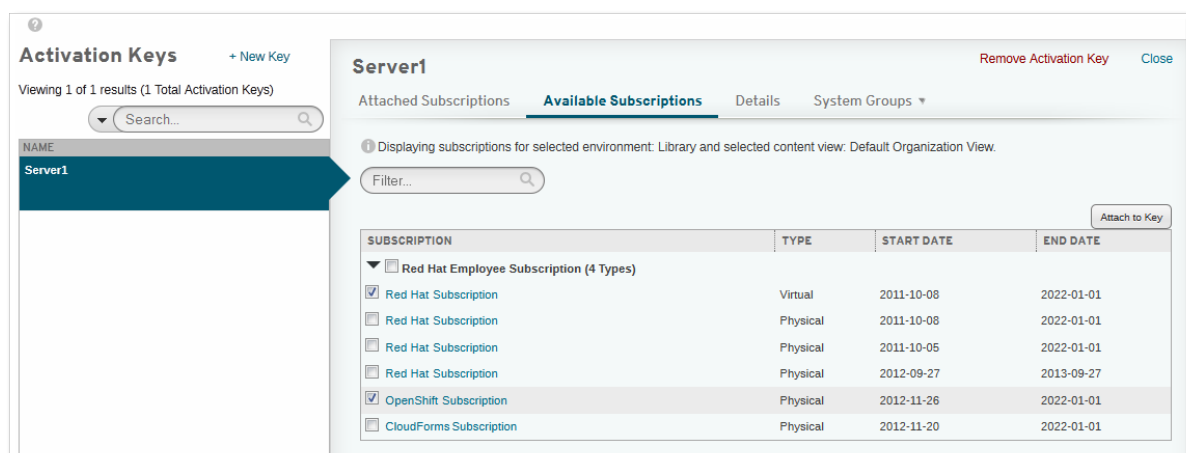
6.3. ATTACHING AND UPDATING SUBSCRIPTIONS FOR AN ACTIVATION KEY

The subscriptions attached to an activation key are the subscriptions that will be automatically attached to a system when it is registered using that activation key. It does not limit a system to only those subscriptions (whatever subscriptions are available to the organization as a whole are available to the system on subsequent attach operations).

1. Log into the Subscription Asset Manager UI.
2. Hover over the **Subscriptions** item in the top menu, and click the **Activation Keys** item.



3. Select the activation key from the column on the left.
4. Click the **Available Subscriptions** tab. A list of available subscriptions is displayed. If more than one subscription is available for the same product, then the product is listed and must be expanded (by clicking the arrow by its name) to select which specific subscription to use.

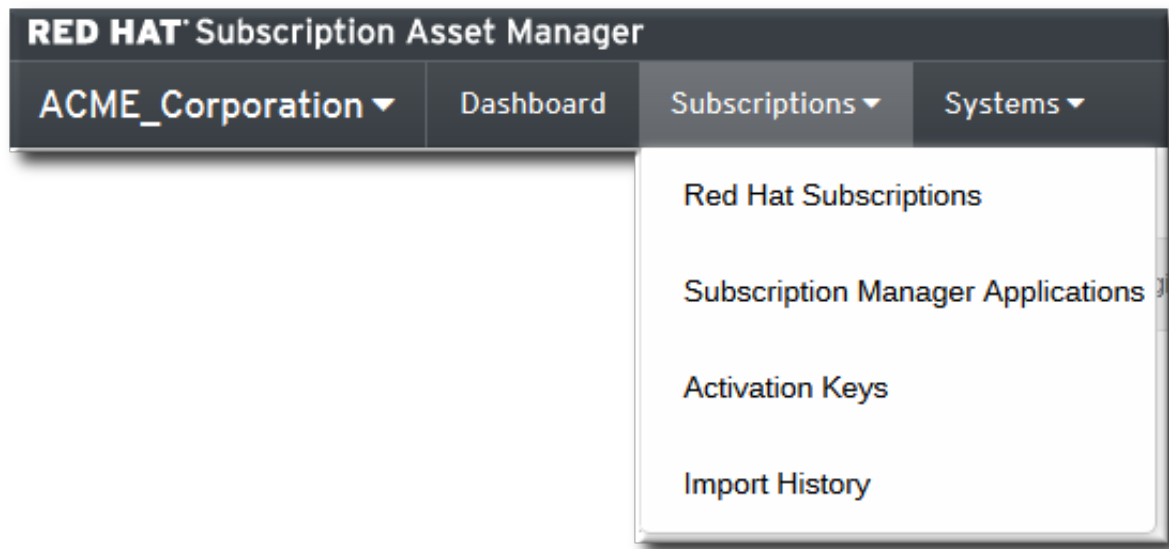


5. Select the checkbox by the subscriptions to add to the key.
6. Click the **Attach to Key** button.

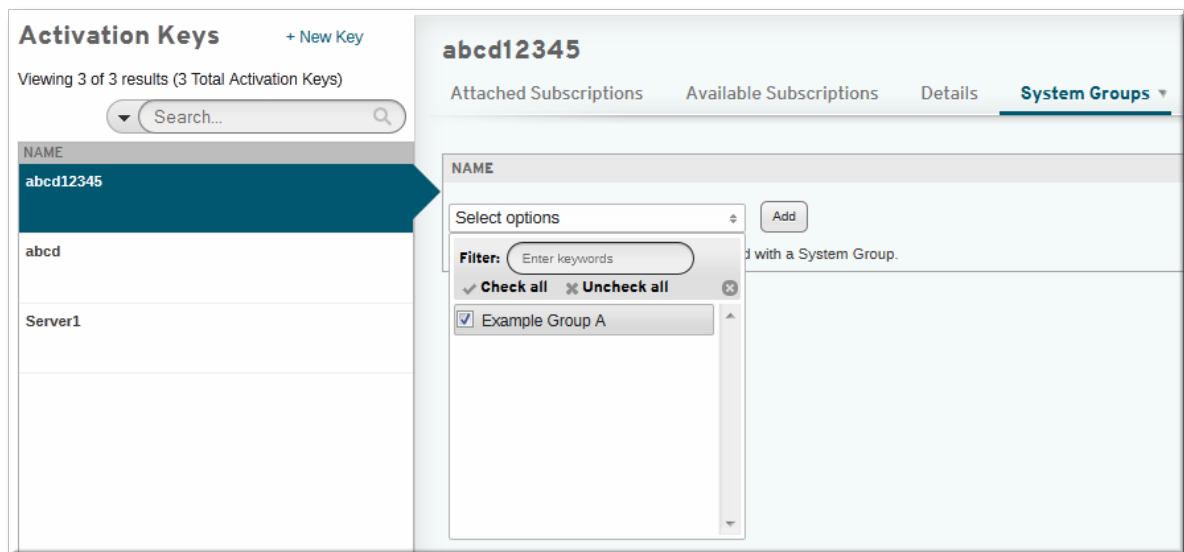
6.4. ASSIGNING SYSTEM GROUPS TO A KEY

When a system group is associated with an activation key, any system which registered using the key is automatically assigned to the system group. This can allow for some access controls to be immediately applied to the system.

1. Log into the Subscription Asset Manager UI.
2. Hover over the **Subscriptions** item in the top menu, and click the **Activation Keys** item.



3. Select the activation key from the column on the left.
4. Click the **System Groups** tab.
5. Click the **Select options** dropdown menu to expose the list of available system groups, and select the checkboxes by the system groups to include with the key.



6. Click the **Add** button.

6.5. REGISTERING A SYSTEM USING AN ACTIVATION KEY

An activation key must be used when a system is registered, not applied after a system is registered.

6.5.1. Using Activation Keys from the GUI

Activation keys for Subscription Asset Manager are configured before the system is ever created or added to the inventory, and the activation keys are passed *as part of* registering the system.

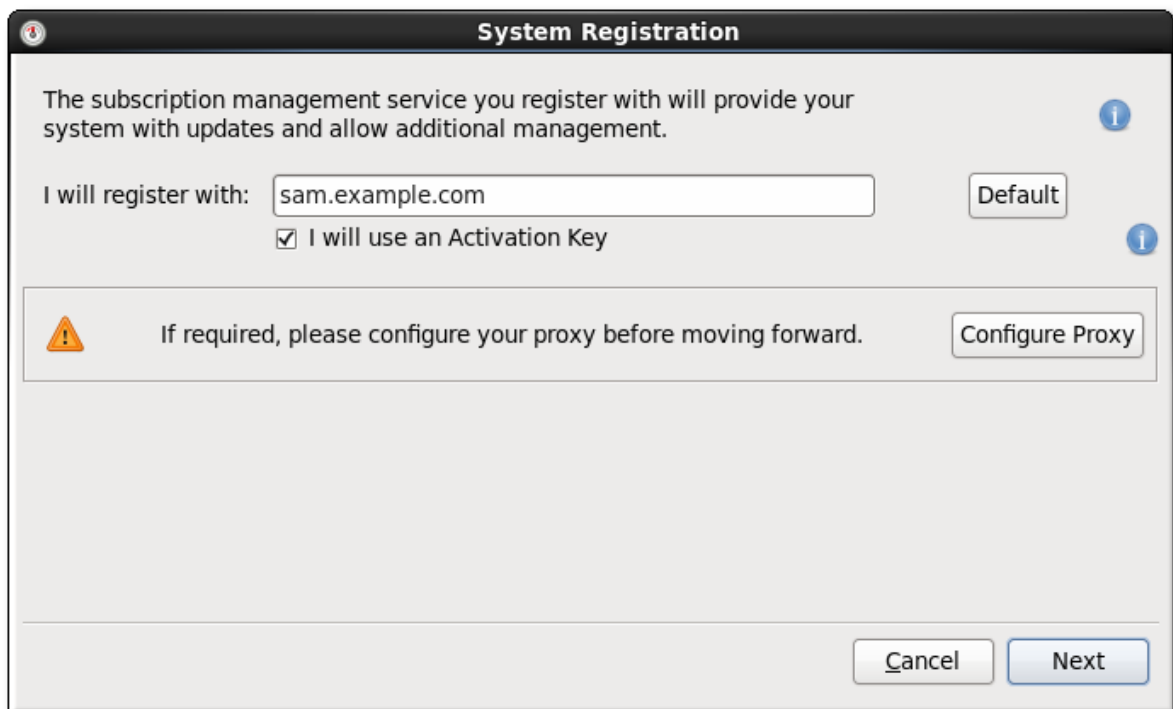
1. Install the configuration RPM to configure Red Hat Subscription Manager to point to the Subscription Asset Manager instance. For example:

```
[root@server ~]# rpm -ivh http://sam.example.com/pub/candlepin-cert-consumer-latest.noarch.rpm
```

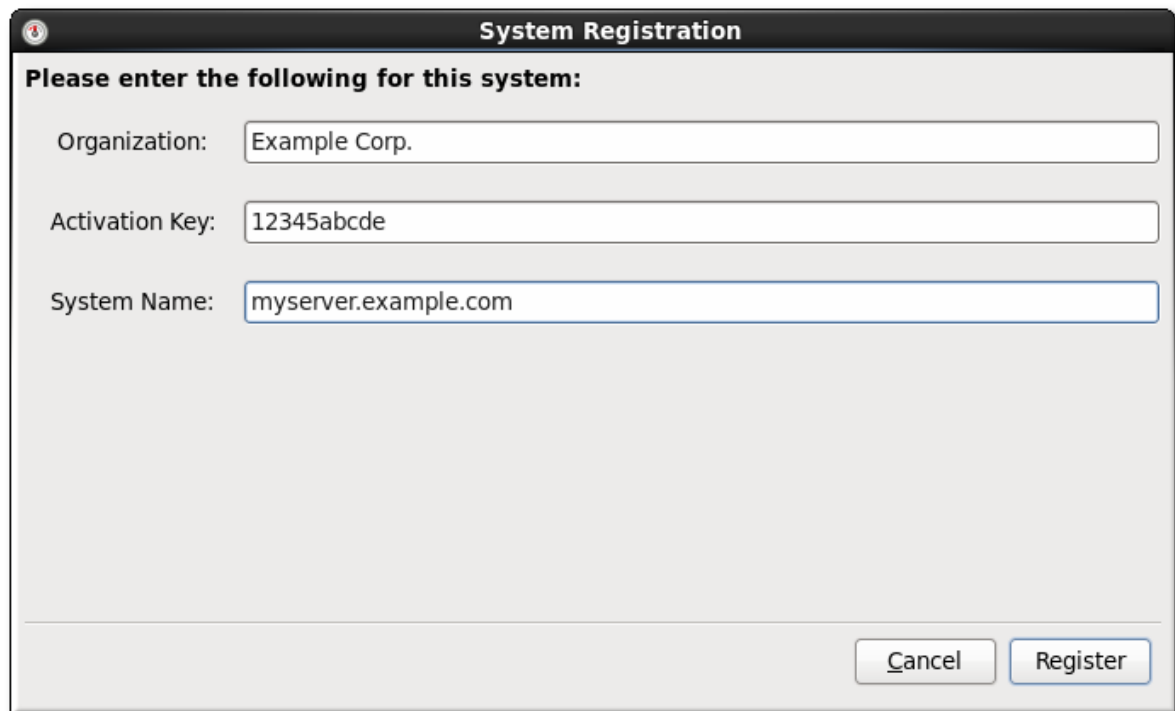
2. Launch Subscription Manager with the **--register** option to open the registration screens immediately.

```
[root@server ~]# subscription-manager-gui --register
```

3. Check the **I will use an Activation Key** checkbox and click the **Next** button.



4. Enter the name of the organization to which the system will belong, the activation key value (an alphanumeric string), and the system name to use for the entry in Subscription Asset Manager.

A screenshot of a 'System Registration' dialog box. The title bar says 'System Registration'. Below the title bar, it says 'Please enter the following for this system:'. There are three text input fields: 'Organization:' with 'Example Corp.', 'Activation Key:' with '12345abcde', and 'System Name:' with 'myserver.example.com'. At the bottom right, there are two buttons: 'Cancel' and 'Register'.

5. Click the **Register** button.

After the registration completes, all of the pre-configured subscriptions are attached to the system.

6.5.2. Using Activation Keys from the Command Line

Activation keys for Subscription Asset Manager are configured before the system is ever created or added to the inventory, and the activation keys are passed *as part of* registering the system.

1. Install the configuration RPM to configure Red Hat Subscription Manager to point to the Subscription Asset Manager instance. For example:

```
[root@server ~]# rpm -ivh http://sam.example.com/pub/candlepin-cert-consumer-latest.noarch.rpm
```

2. Then, run the **register** command with the **--activationkey** parameter to attach the configured subscriptions.

```
[root@server ~]# subscription-manager register --username=jsmith --password=secret --org="IT Dept" --activationkey=abcd1234
```

If there are multiple organizations — or even if there is only a single organization but it is possible for there to be multiple ones — it is still necessary to specify the organization for the system. That information is not defined in the activation key.

CHAPTER 7. MANAGING VIRTUAL HOSTS AND GUESTS

Several services can automatically detect guests on a virtual host system and register them as virtual systems. This allows subscriptions which are specific to virtual systems to be available to the guest and for subscriptions which are inherited from the host to be applied to the guest.

7.1. SUPPORTED HYPERVISORS

The **virt-who** process can detect and associate guests on several different types of hypervisors:

- Red Hat Enterprise Virtualization (KVM)
- Xen
- HyperV
- VMware ESX

7.2. ABOUT HOST/GUEST ASSOCIATIONS

Subscription relationships have a lot of potential flexibility. Some subscriptions can be applied to a physical machine *or* to a certain number of virtual machines, while others can be applied to a physical host and then inherited by guests. This is covered in detail in [Section 5.1, “About Subscriptions on Systems”](#).

For subscriptions to be managed effectively — particularly with inheritable subscriptions or interactions between subscriptions — there has to be an internal awareness in the subscription service of the relationships between hosts and guests. This is a *host/guest mapping*, which is literally a list of all of the guest identifiers for a given hypervisor.

Hypervisors are registered as a special type of consumer in Subscription Asset Manager or Customer Portal Subscription Management. Hypervisors themselves are managed as regular physical systems, but the hypervisor type indicates that that particular system will have guests mapped to it, and that subscriptions may be inheritable or applied differently to those guests.

With a host/guest mapping to associate every guest with a specific host, a subscription service can properly attach a single subscription to a virtual host and then apply an included and inheritable subscription to its guest (for example), rather than consuming two separate subscriptions for each instance.

This association is done by extracting a *universally unique identifier* for each guest and associating it with its hypervisor. These UUIDs are part of the system facts for each virtual system.

The hypervisor is registered first with Subscription Asset Manager, and then a related process on the system scans for any guests and submits the discovered UUIDs to the subscription service. This is done by the **virt-who** process on the hypervisor.

There are three factors that must be true for Subscription Asset Manager to recognize the host/guest association and properly attach subscriptions:

- The appropriate virtual detection process must be run periodically to detect new guest instances.
- The hypervisor and the guest systems must be registered to the same Subscription Asset Manager instance.

- The hypervisor must have a subscription attached to it that includes virtual subscriptions or inheritable subscriptions.

7.3. SETTING UP A RHEV (KVM) OR XEN HYPERVISOR

1. Configure the Subscription Manager application to use the Subscription Asset Manager service and CA certificate.

```
[root@rhel-server ~]# rpm -ivh http://sam.example.com/pub/candlepin-
cert-consumer-latest.noarch.rpm
```

2. Then register the system as a hypervisor and attach any required subscriptions.

```
[root@rhel-server ~]# subscription-manager register --
type=hypervisor --username=admin --password=secret --org=1234 --
auto-attach
```

The organization ID should be available in the Subscription Asset Manager UI or in the Portal entry for the organization. If another system is already registered to that organization, then the ID is available using the **subscription-manager orgs** command.

3. Install the **virt-who** packages on the hypervisor.

```
[root@server ~]# yum install virt-who
```

4. Open the **virt-who** configuration file (**/etc/sysconfig/virt-who**) and set it to use the **libvirt** service with the hypervisor.

```
VIRTWHO_LIBVIRT=1
```

5. Start the **virt-who** service.

```
[root@server ~]# service virt-who start
```

6. Configure every host in the RHEV/Xen environment.
7. Create and register virtual machines as normal.

7.4. SETTING UP A VMWARE HYPERVISOR



NOTE

The **virt-who** packages that create the host/guest mapping are available for Red Hat Enterprise Linux. In a VMware environment, there must be a Red Hat Enterprise Linux system available to run the **virt-who** process which connects to the VMware hypervisor.

1. Configure the Subscription Manager application on the virtual system to use the Subscription Asset Manager service and CA certificate.

```
[root@rhel-server ~]# rpm -ivh http://sam.example.com/pub/candlepin-
cert-consumer-latest.noarch.rpm
```


2. Then register the Red Hat Enterprise Linux system (which communicates with the VMware server) as a hypervisor.

```
[root@rhel-server ~]# subscription-manager register --
type=hypervisor --username=admin --password=secret --org=1234-56789
--auto-attach
```

The organization ID should be available in the Subscription Asset Manager UI or in the Portal entry for the organization. If another system is already registered to that organization, then the ID is available using the **subscription-manager orgs** command.

By default, the hypervisor name is *esx hypervisor UUID*. This name can be changed in the Subscription Asset Manager UI by editing the system entry.

3. Install the **virt-who** packages.

```
[root@server ~]# yum install virt-who
```

4. Open the **virt-who** configuration file (**/etc/sysconfig/virt-who**) and set it to use the Subscription Asset Manager instance and the appropriate vCenter server.

1. Set that the VMware server will interact with a Subscription Asset Manager instance.

```
VIRTWHO_BACKGROUND=1
VIRTWHO_SAM=1
```

2. Enable ESX mode, and set the environment to **Library**:

```
VIRTWHO_ESX=1
VIRTWHO_ESX_ENV=Library
```

3. Specify the owner of the subscriptions. This must be the ID or name of an organization that is created within the Subscription Asset Manager instance. For example:

```
VIRTWHO_ESX_OWNER=6340056
```

The organization ID should be available in the Subscription Asset Manager UI or in the Portal entry for the organization. If another system is already registered to that organization, then the ID is available using the **subscription-manager orgs** command.

4. Set the hostname or IP address of the vCenter server:

```
VIRTWHO_ESX_SERVER=vcenter.example.com
```

5. Specify the username and password to use when connecting to the vCenter server:

```
VIRTWHO_ESX_USERNAME=admin
VIRTWHO_ESX_PASSWORD=secret
```

6. Save the changes to the configuration file.

5. Start the **virt-who** service; this begins gathering all of the host/guest data.

```
[root@rhel-server ~]# service virt-who start
```

The data are added to the **/var/lib/virt-who/hypervisor-systemid-UUID** file.

6. Use **chkconfig** to configure the **virt-who** service so that it starts automatically when the system starts.

```
[root@rhel-server ~]# chkconfig virt-who on
```

7.5. REGISTERING GUEST INSTANCES

Register a virtual system the same as a physical system.



NOTE

The **virt-who** process must be running on the virtual host or on a hypervisor in the environment (for VMware) to ensure that **virt-who** process maps the guest to a physical host, so the system is properly registered as a virtual system. Otherwise, the virtual instance will be treated as a physical instance.

1. Configure the Subscription Manager application on the virtual system to use the Subscription Asset Manager service and CA certificate.

```
[root@virt-server ~]# rpm -ivh http://sam.example.com/pub/candlepin-  
cert-consumer-latest.noarch.rpm
```

2. Then register the system to the same organization as its host.

```
[root@virt-server ~]# subscription-manager register --username=admin  
--password=secret --org=12345-67-8901 --auto-attach
```

The organization ID should be available in the Subscription Asset Manager UI or in the Portal entry for the organization. If another system is already registered to that organization, then the ID is available using the **subscription-manager orgs** command.

7.6. CREATING A DATA CENTER

There is a specific subscription available for data centers which registers a physical system as a hypervisor and then allows an unlimited number of virtual guests to be installed and registered on that system. That physical system can be a Red Hat Enterprise Linux system running RHEV or Xen, or it can be a non-Linux system, running VMware or HyperV. The configuration does not matter; as with running any virtualized environment, there simply must be one Red Hat Enterprise Linux system to run the **virt-who** process to create the host/guest mapping.

For each physical host in the environment:

1. Set up the host or hypervisor, as described in [Section 7.3, “Setting up a RHEV \(KVM\) or Xen Hypervisor”](#) or [Section 7.4, “Setting up a VMware Hypervisor”](#).

2. Attach the data center subscription to the hypervisor entry. The name of the subscription is *Red Hat Enterprise Linux for Virtual Datacenters ... System:Physical*.
3. Register all guests for that host/hypervisor, as described in [Section 7.4, “Setting up a VMware Hypervisor”](#).



NOTE

If a virtual instance is migrated from one hypervisor to another, the Red Hat Enterprise Linux subscription is preserved, but any subscriptions for additional products, such as JBoss Enterprise Application Platform, must be released and then re-attached.

7.7. REMOVING AND RESTORING HOST AND GUEST ACCOUNTS

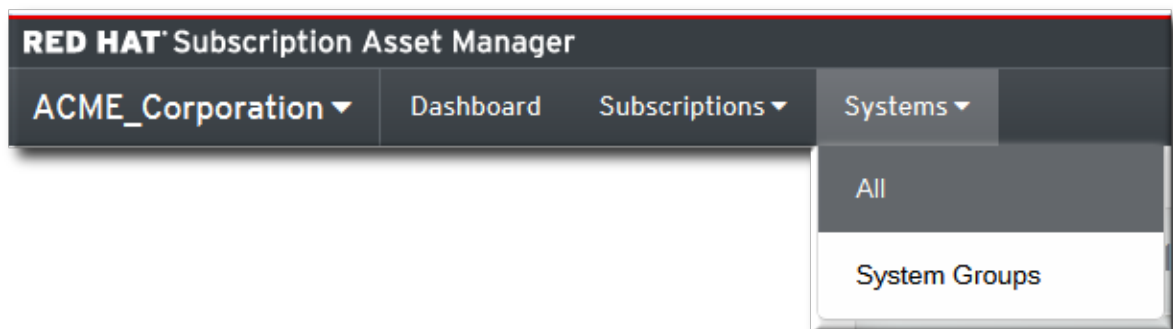
7.7.1. Removing a Guest Entry

If a guest system is running, then simply unregister the system

```
[root@virt-guest ~]# subscription-manager unregister
```

If the system has been deleted, however, the virtual service (like **virt-who** cannot tell whether the service is deleted or paused. In that case, the system must be removed manually from Subscription Asset Manager.

1. Log into the Subscription Asset Manager UI.
2. Hover over the **Systems** item in the top menu, and click the **All** item.



3. Click the name of the system in the column on the left.
4. At the top of the system's details page, click the **Remove System** link.

7.7.2. Removing a Hypervisor Entry

1. Unregister the hypervisor.

```
[root@rhel-server ~]# subscription-manager unregister
```

2. For VMware, delete the UUID file, `/var/lib/virt-who/hypervisor-systemid-UUID`, to remove the host/guest mapping records.

7.7.3. Reregistering a Hypervisor

When a hypervisor is deleted, the deletion itself is recorded in Subscription Asset Manager as a marker. The hypervisor is prevented from re-registering with the same subscription inventory ID. If the hypervisor is added back later, it is registered with a new ID. This is why the UUID file must be deleted when the hypervisor entry is removed; if it is ever re-registered, then new host/guest mappings must be created.

If for some reason the hypervisor must be re-registered with the same UUID, then the deletion record can be removed.

```
[root@sam-server ~]# headpin -u admin -p admin system remove_deletion --  
uuid=<system uuid>
```

Then, the system can be re-registered using its original ID:

```
[root@rhel-server ~]# subscription-manager register --username admin-  
example --password secret --org=12345678-90 --consumerid=7d133d55-876f-  
4f47-83eb-0ee931cb0a97
```

The organization ID should be available in the Subscription Asset Manager UI or in the Portal entry for the organization. If another system is already registered to that organization, then the ID is available using the **subscription-manager orgs** command.

CHAPTER 8. VIEWING SYSTEM-LEVEL SUBSCRIPTION INFORMATION

The ultimate goal of subscription management is to allow administrators to identify the relationship between their systems and the subscriptions used by those systems. This can be done from two different perspectives: from the perspective of the local system looking externally to potential subscriptions and from the perspective of the organization (top-level account), looking down at the total infrastructure of systems and all subscriptions.

Subscription Asset Manager has several different ways of conveying subscription and system information. This includes information about insufficient or expiring subscriptions, which is invaluable to administrators for maintaining current subscriptions.

8.1. VIEWING THE HIGH-LEVEL INFORMATION IN THE DASHBOARD

The Subscription Asset Manager dashboard shows a count of all *systems* registered with that organization and their overall subscription status.

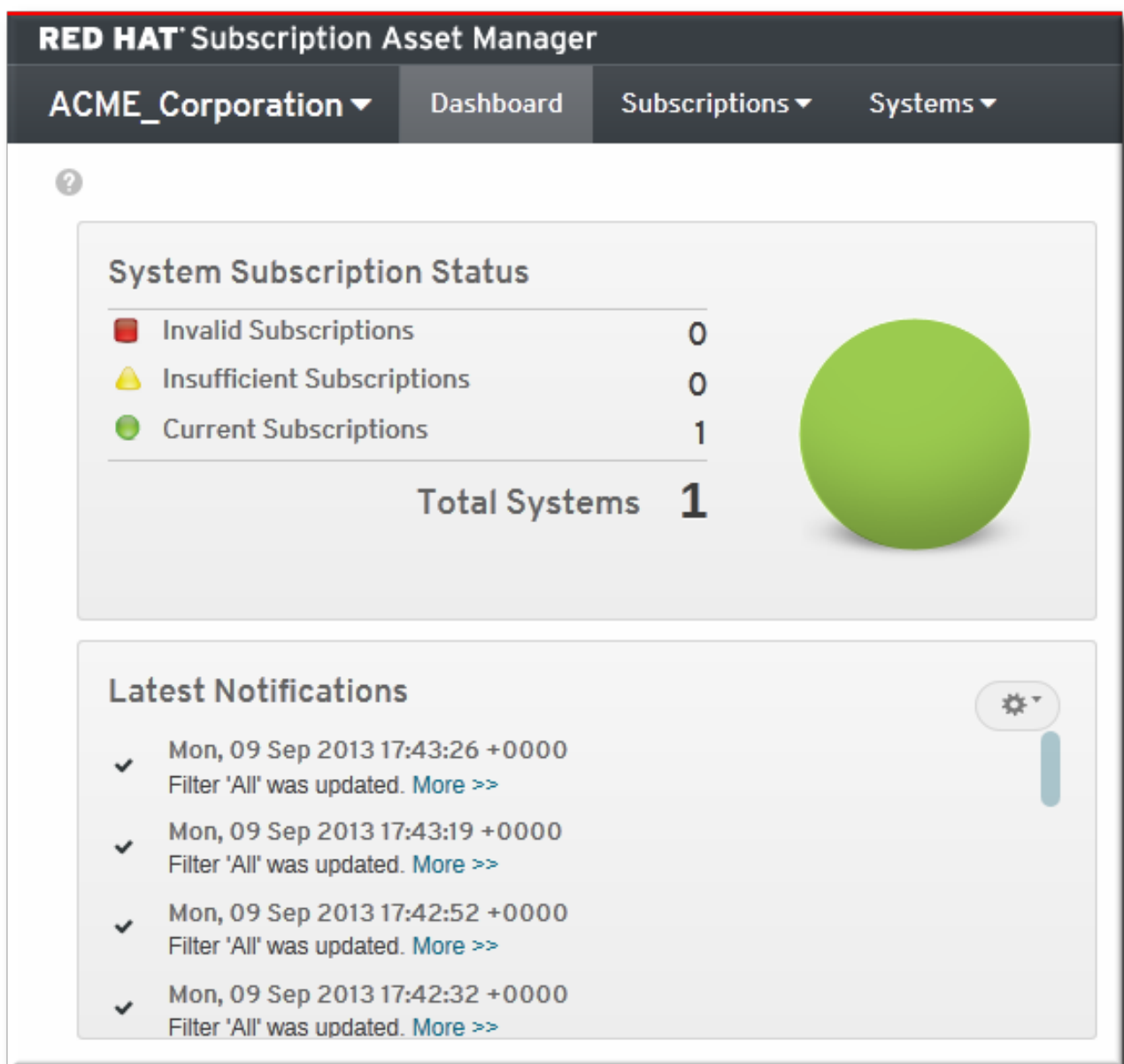


Figure 8.1. The Subscription Asset Manager Dashboard

Subscription status refers to the status of all subscriptions for all products which are installed on a system. For example, if a system has Red Hat Enterprise Linux, OpenShift, and Directory Server all installed, then that system must have subscriptions for Red Hat Enterprise Linux, OpenShift, and Directory Server attached to it so that it is current.

There are three categories of subscription status:

- *Current subscriptions* mean that a system has a subscription for every install product, in the appropriate quantity.
- *Invalid subscriptions* mean that a system has installed products but at least one of those products has no corresponding subscription for it.
- *Insufficient subscriptions* is a slightly more complex state. It means that at least one installed product has some subscriptions for it, but not enough. Each subscription states some attribute that applies to it. For example, an operating system subscription may specify a certain number of cores or a certain amount of RAM. If a system has four cores and the subscription specifies that it covers two sockets, then the system requires two subscriptions. If only one subscription is attached, then the system is in an *insufficient* state.

8.2. VIEWING SERVER NOTIFICATIONS

Every time any action is taken in Subscription Asset Manager — such as adding a new user, editing an organization's configuration, importing a manifest, or running a report — a system notification is recorded. Notifications include both success and error messages.

The most recent notifications are shown in the **Latest Notifications** area of the **Dashboard** tab.

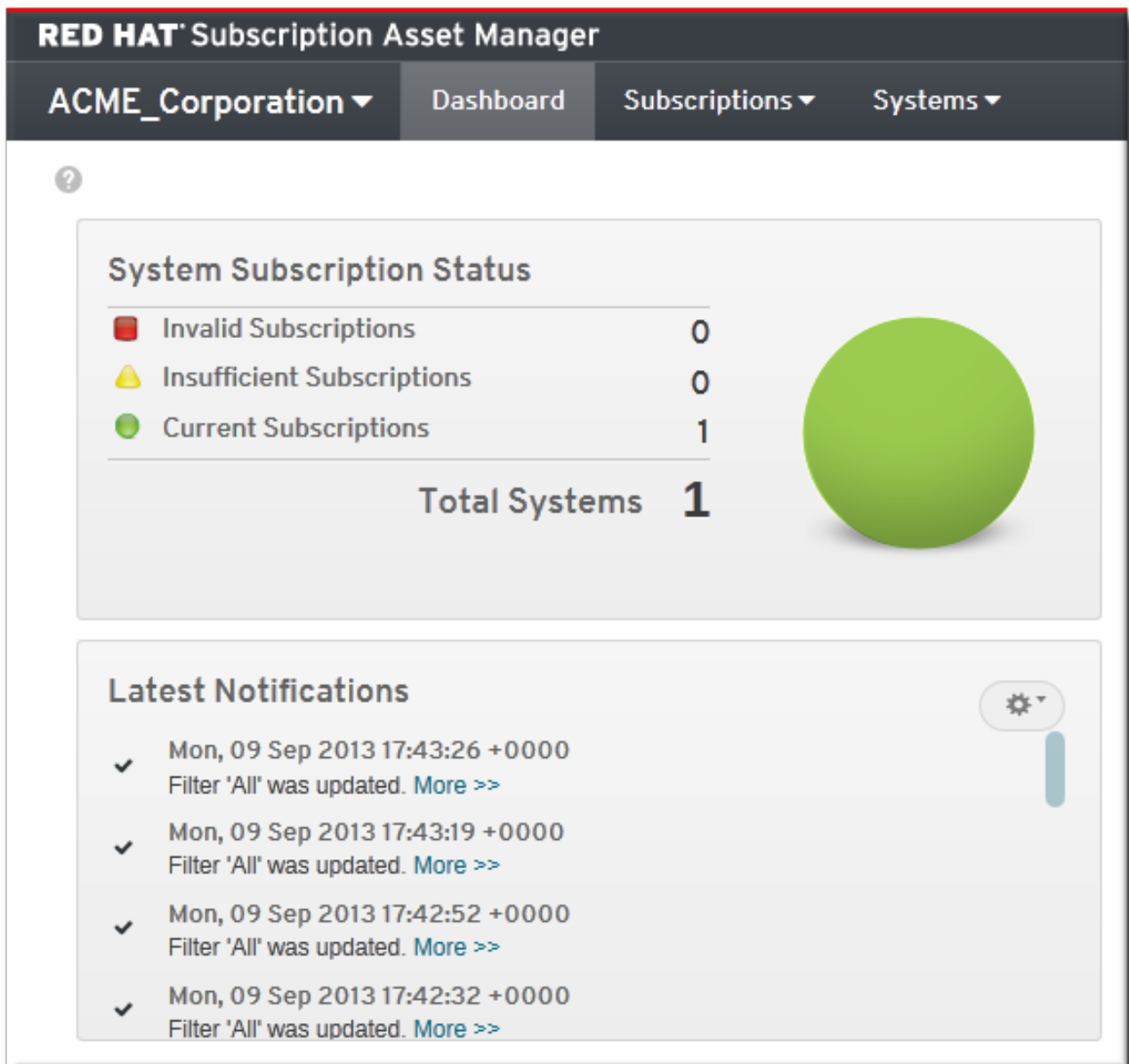


Figure 8.2. Notifications in the Dashboard

By default, the most recent five notifications are displayed, but that can be changed to 15 or 30 notifications by clicking the gear icon.

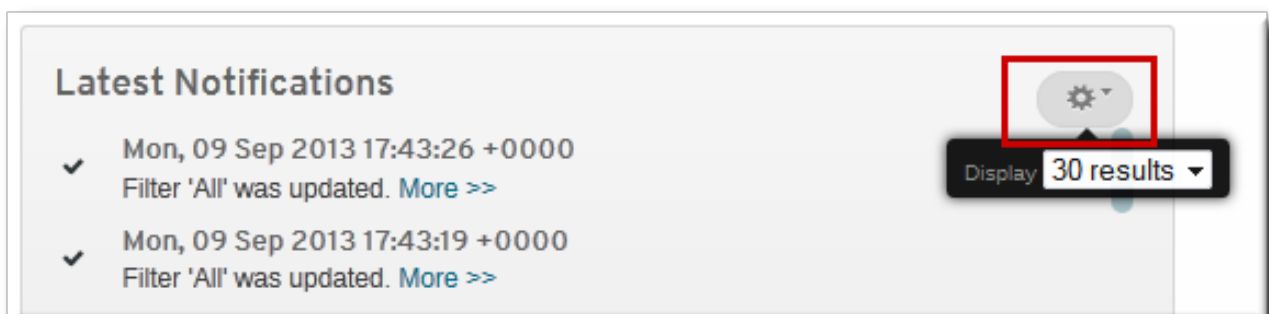


Figure 8.3. Changing the Number of Notifications Displayed

The complete list of notifications can be opened by clicking the **More** link by any notification in the **Dashboard** or by clicking the notification icon in the upper right of the Subscription Asset Manager UI.

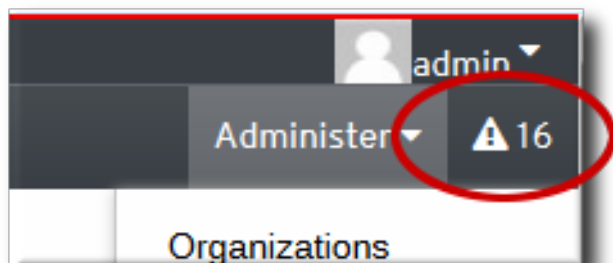


Figure 8.4. Notification Link in the Admin Menu

The **User Notifications** page lists the time and date of each notification, the type of notification (either **success** or **error**), and a description of the notification.

User Notifications Delete All Filter...			
CREATED (UTC) ▼	LEVEL	ORGANIZATION	DESCRIPTION
Tue, 06 Aug 2013 15:31:15 +0000	success	Red Hat SatTeam	Successfully added system[s] to group 'Example Group B'.
Tue, 06 Aug 2013 15:30:52 +0000	success	Red Hat SatTeam	Successfully added system[s] to group 'Example Group B'.
Tue, 06 Aug 2013 15:30:38 +0000	success	Red Hat SatTeam	Successfully added system[s] to group 'Example Group B'.
Tue, 06 Aug 2013 15:30:30 +0000	success	Red Hat SatTeam	System Group Example Group B created successfully.
Tue, 06 Aug 2013 15:30:05 +0000	success	Red Hat SatTeam	Successfully added system[s] to group 'Example Group A'.
Tue, 06 Aug 2013 15:29:51 +0000	success	Red Hat SatTeam	System Group Example Group A created successfully.
Fri, 02 Aug 2013 12:29:07 +0000	success		User updated successfully.

Figure 8.5. List of Notifications

Individual notifications cannot be deleted, but it is possible to clear the entire notifications queue by clicking the **Delete All** link.

8.3. VIEWING SYSTEM ADMINISTRATIVE NOTIFICATIONS

When certain administrative events are performed on a system, they are recorded within an event history for the system, as well as a notification in Subscription Asset Manager.

System-level administrative notifications are viewed in the system's entry page, in the **Details > Event History** tab.

Every administrative event for that system, alone, is listed with a timestamp.

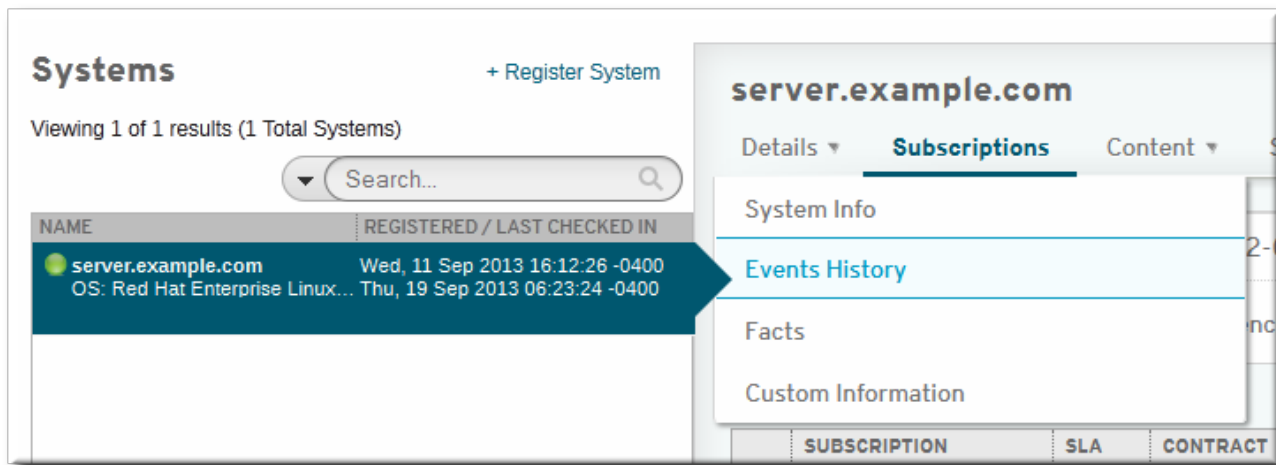


Figure 8.6. Event History in the Menu

Every administrative event for that system, alone, is listed with a timestamp.

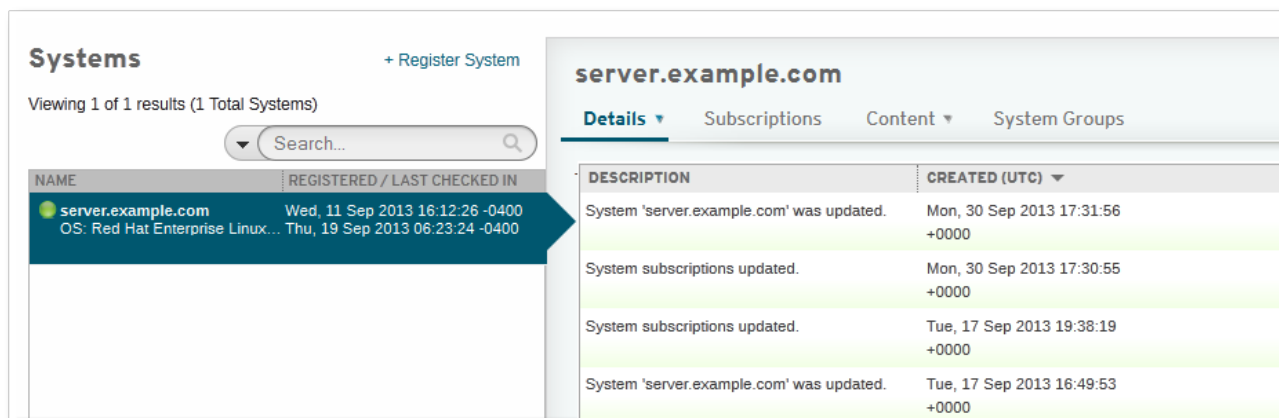


Figure 8.7. List of Notifications

8.4. CHECKING INDIVIDUAL SYSTEM STATUS

The **Dashboard** shows the cumulative subscription status counts for all systems within the Subscription Asset Manager organization. However, it is possible to view the information for each individual system, including subscription expiration dates and installed products.

First, open the system entry:

1. Hover over the **Systems** tab in the top menu and select the **All** item.
2. In the search box on the left of the systems column, search for the specific system.
3. Click the name of the system in the column on the left.

The server list entry itself shows the overall subscription status for the system by using a red (invalid), yellow (insufficient), or green (current) icon.

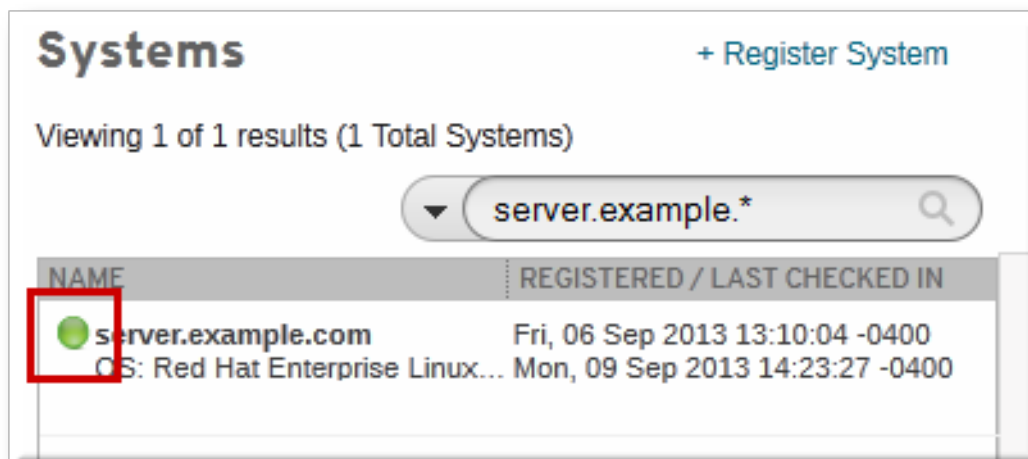


Figure 8.8. Status Icon in the System List

There are a couple of different places in the system page that show the details of the subscription status.

- The **Details** tab shows the status and (if current) the expiration date for the subscriptions.

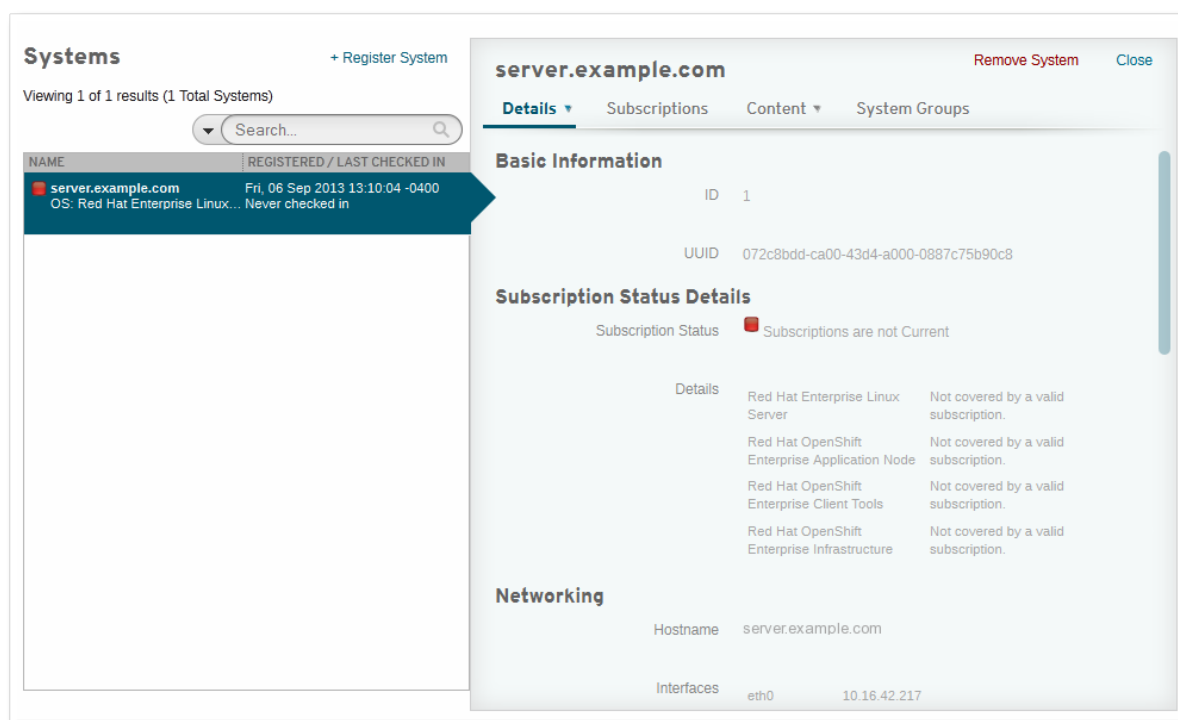


Figure 8.9. Status Details

- The **Subscriptions** tab also shows the status and the expiration date (if subscriptions are current). Additionally, the **Subscriptions** tab has a list of available and attached subscriptions, so that the subscriptions for the system can be reassigned as necessary.

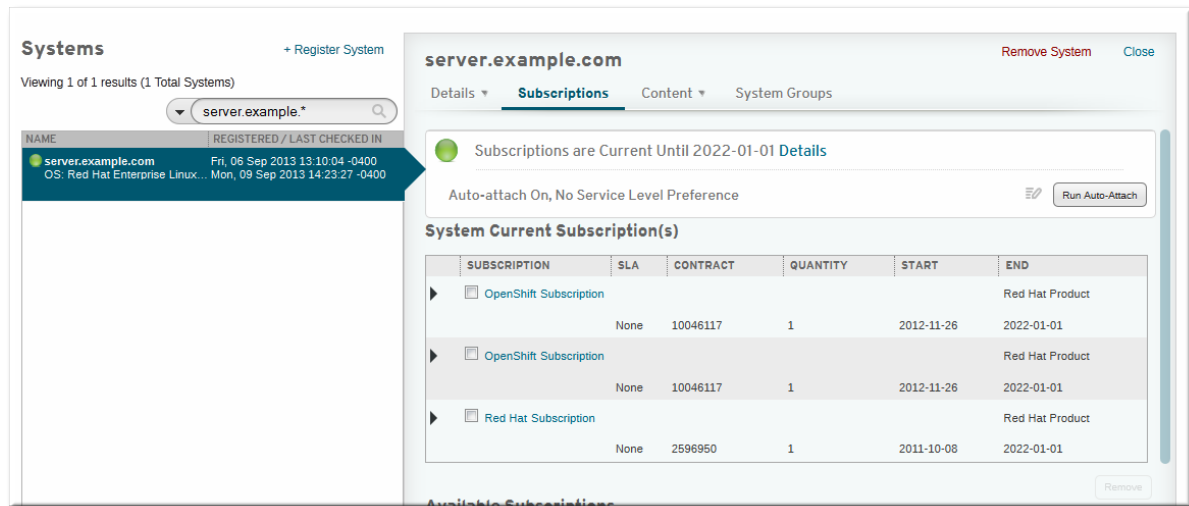


Figure 8.10. Status and Subscription Lists

8.5. REMEDIATING PROBLEM SUBSCRIPTION STATUS

The local system can run a job every four hours to update subscriptions automatically. When products are newly-installed on a system or when the subscription status changes, then the system may not be fully-covered, which is reflected in the system status.

Subscriptions can be added to systems individually ([Section 5.6.1, “Attaching Subscriptions to a System”](#)), but the easiest way to remediate an insufficient or invalid system status is to run an autoattach operation. Autoattach operations can be run on a single system or all systems within an organization ([Section 5.8, “Running Autoattach Operations Manually”](#)).

8.6. CREATING SUBSCRIPTION ASSET MANAGER USAGE REPORTS

As [Section 4.1, “About the Structure of Organizations and Distributors”](#) covers, organizations in Subscription Asset Manager are discrete and separate. Cumulative information in areas such as the **Dashboard** are for the organization being viewed only. Even if multiple organizations are configured on a single Subscription Asset Manager server, only one organization is viewable at a time.

It can be useful or even necessary to have a cross-organization or cross-distributor view of subscription allocation and status in order to maintain regulatory and policy compliance. Subscription Asset Manager reports can be configured to return information for multiple organizations, which allows that kind of cross-organizational data to be compiled.

8.6.1. Prerequisites

When using enhanced reporting, there are some additional system requirements:

- All of the prerequisites in [Section 2.1, “Prerequisites”](#).
- The **crond** service must be running.
- An additional 4 GB of disk space must be available for the reporting database journal.
- Additional packages for the reporting server
 - splice
 - ruby193-rubygem-splice_reports

- spacewalk-splice-tool

8.6.2. Setting up Reporting After Installation

Subscription Asset Manager reporting is an additional module that requires additional packages to be installed. This is covered in the installation procedures in [Section 2.3, “Installing and Configuring Enhanced Reporting”](#).

If a Subscription Asset Manager instance is already configured, then the additional packages can be pulled in. For example, using **yum**:

```
[root@server ~]# yum install splice ruby193-rubygem-splice_reports  
spacewalk-splice-tool
```

When the reporting packages are installed, there is an additional item in the administration menu to create and run reports.

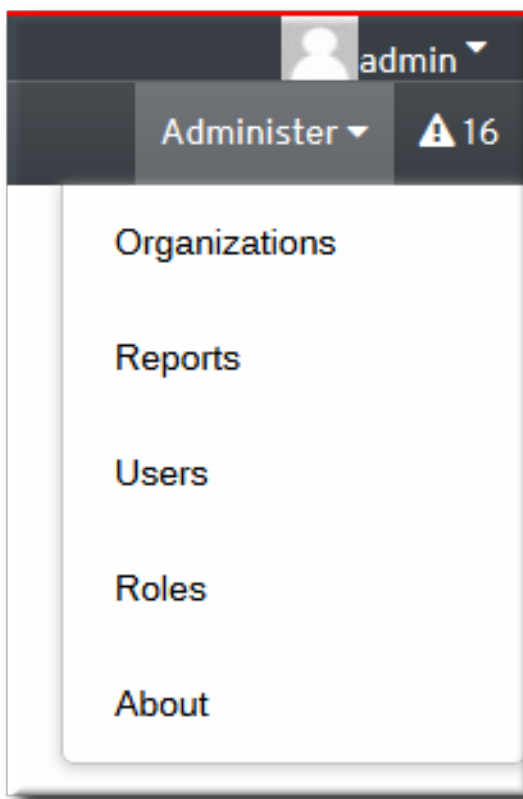


Figure 8.11. The Reports Menu Item

8.6.3. Creating Report Filters

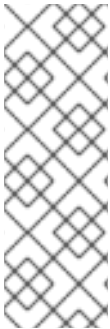
The Subscription Asset Manager report is actually a collection of filters that collect and structure data for different organizations.

There is one default report that checks for subscriptions for all statuses, in all organizations, in all Satellite servers (if configured), over the past 24 hours.

There is a lot more flexibility possible with the report form, however. In particular, there are three versatile settings:

- The organizations to check for the report

- The subscription statuses to include
- The date range to check; this looks for systems which had the status *within the given range*, which may not necessarily be the current status for the system



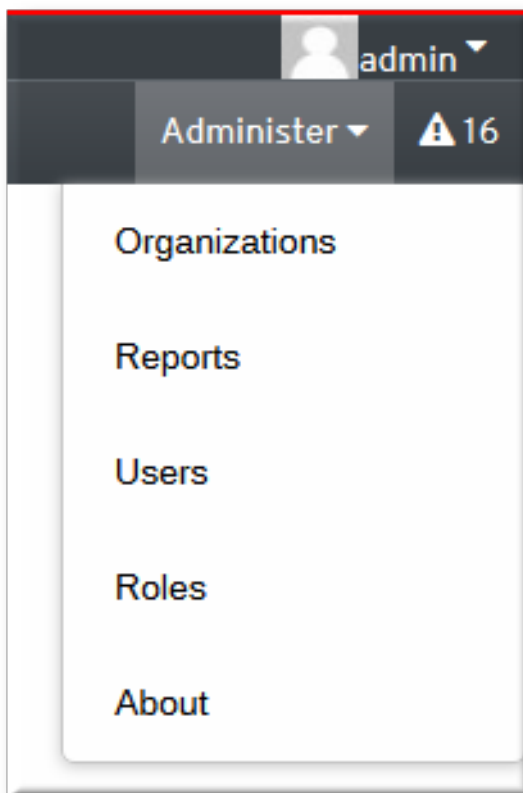
NOTE

Here are some good reports for tracking infrastructure status and for compliance audits:

- All systems that have changed to invalid or insufficient (status) in the past 24 hours.
- All systems that will have invalid or insufficient subscriptions (meaning, the existing subscriptions will expire) within the next three months.

To create a new report filter:

1. Click the **Reports** item in the administration menu.



2. In the left column, click the **New Filter** link.
3. Fill in the required information for the report, including the organizations, statuses, date range, and active states.

Report Filters + New Filter

Viewing 2 of 2 results (2 Total Report Filters)

Search...

NAME

Red Hat Default Report

All

New Filter

Name: Example report

Description: for 2013

Subscription Status: Current Invalid

Satellite Server: sam13-dlackey-demo

Organizations: ACME_Corporation

LifeCycle State: Active Inactive

Search Previous Number of Hours: ☐ ☒

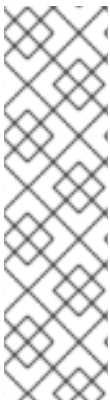
Search by Date Range: ☐ ☒

Start Date: 01/01/2013

End Date: 12/31/2013

Save Filter

- Click the **Save Filter** button.



NOTE

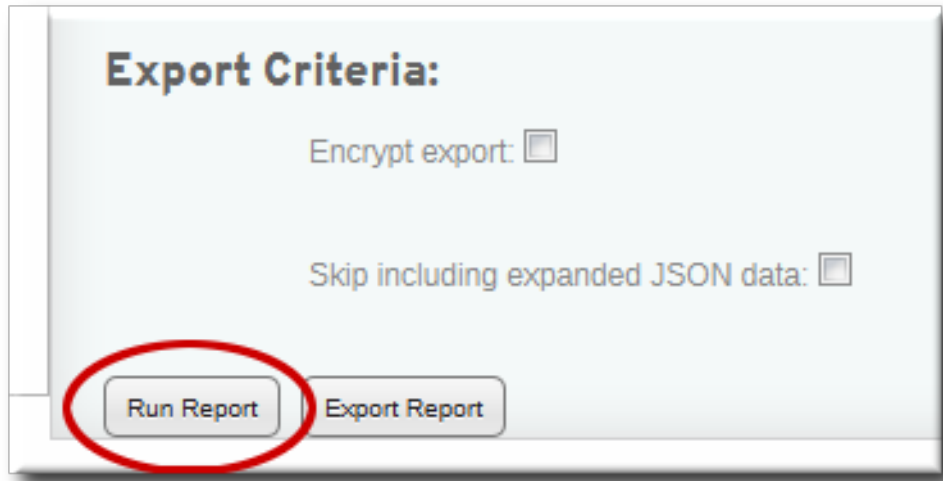
The data within the Subscription Asset Manager database includes historic subscription statuses. This allows reports to be generated to track subscriptions at a given point in time, not just the current date.

For example, if purchasing occurs in July, then a report can be configured to search for insufficient or invalid systems from April through June, to influence purchasing decisions.

The time filter also allows very short windows of time — the previous 24 or 48 hours — to be able to identify and remediate suscription issues immediately.

8.6.4. Running Reports

- Click the **Reports** item in the administration menu.
- In the left column, click the name of the report filter to run.
- Scroll to the bottom of the report page, and click the **Run Report** button.



Alternatively, the report results can be exported to a CSV file instead of being rendered in the Subscription Asset Manager UI. To export the data, click the **Export Report** button.

The data are exported to a CSV file and, optionally, a JSON file which contains the system details. These files are contained in a ZIP archive named **report - YEAR-MONTH-DAY-TIMESTAMPZ.zip**.



NOTE

Selecting the **Encrypt export** checkbox means that the exported CSV and JSON files are encrypted and can only be accessed by a private key used by Red Hat support.

8.6.5. Subscription Asset Manager Reports Results and Data

The Subscription Asset Manager report returns a chart of all registered systems for all selected organizations, similar to the **Dashboard**.

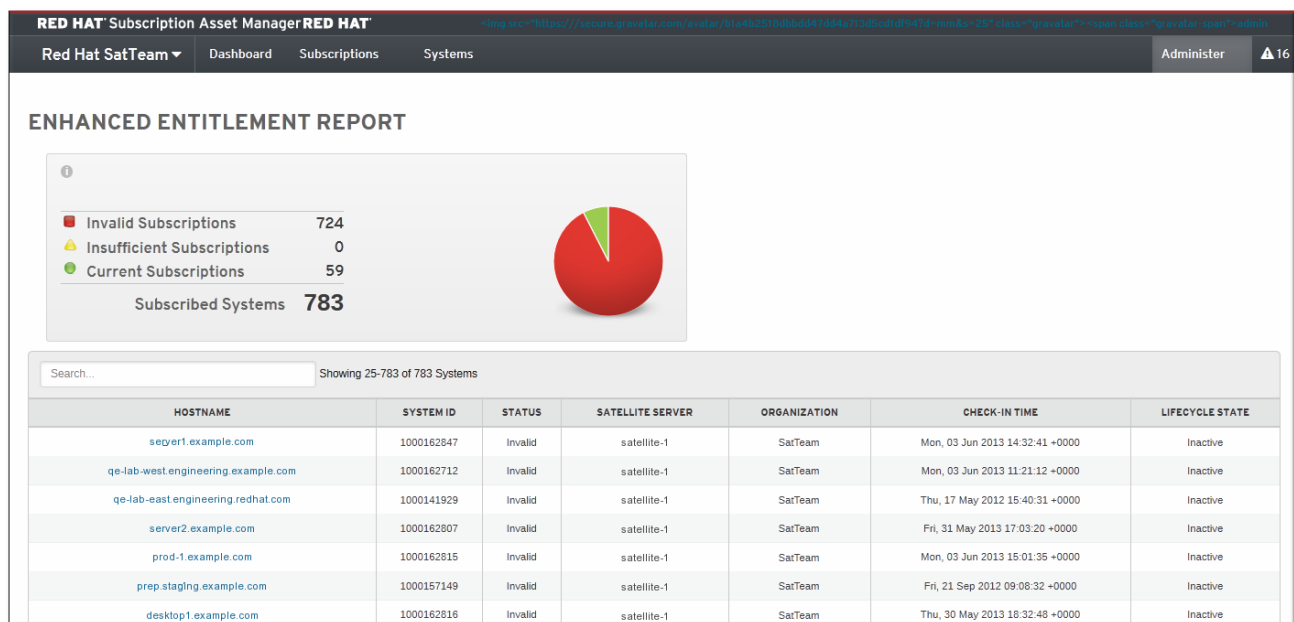


Figure 8.12. Report Results

There is also a list of all included systems. The list itself contains summary information such as the Subscription Asset Manager or Satellite server to which the system is registered, its status, its organization, and its most recent checkin time.

Clicking the name of any system pulls up more details subscription data for the system. The details page includes a history of subscription status changes for the given report period, the list of installed products and subscription information for each, and system facts (attributes about the physical machine and operating system, such as CPU, socket count, RAM, and cores).

The screenshot displays the 'Systems' page in the Subscription Asset Manager. At the top, a navigation bar shows 'ACME_Corporation' and tabs for 'Dashboard', 'Subscriptions', 'Systems', and 'Administer'. Below the navigation bar, a summary box shows subscription status: 0 Invalid Subscriptions, 0 Insufficient Subscriptions, and 1 Current Subscriptions, totaling 1 Subscribed System. A large green circle represents the system. Below this, a search bar and 'Showing 1-1 of 1 Systems' are visible. A table lists the system with columns 'HOSTNAME' and 'SYSTEM ID'. The system 'server.example.com' is selected, and its details are shown in a panel. The details panel includes tabs for 'Subscription Details', 'Facts', 'Products', 'System Check-Ins', and 'Close'. The 'Subscription Details' tab is active, showing three 'OpenShift Subscription' entries with their respective start and end dates and consumed counts.

HOSTNAME	SYSTEM ID
server.example.com	

Subscription Details

Compliant Product Info:

OpenShift Subscription

Start Date: 2012-11-26T05:00:00.000+0000

End Date: 2022-01-01T04:59:59.000+0000

Consumed: 5 out of 10

OpenShift Subscription

Start Date: 2012-11-26T05:00:00.000+0000

End Date: 2022-01-01T04:59:59.000+0000

Consumed: 5 out of 10

OpenShift Subscription

Start Date: 2012-11-26T05:00:00.000+0000

End Date: 2022-01-01T04:59:59.000+0000

Figure 8.13. Report Results: System Details

When the report results are exported, the same information is included in the export files.

The CSV report contains the summary information in the initial reports page: organization, registered subscription server, hostname, and subscription state, among others.

```
_id, record, CHECK-IN TIME, STATUS, DB ID, SATELLITE SERVER, HOSTNAME,
ORGANIZATION, LIFECYCLE STATE,
{"ident"=>"072c8bdd-ca00-43d4-a000-0887c75b90c8"},
522e0970af5d242094000002, 2013-09-09T14:23:27Z, "Current", "072c8bdd-ca00-
43d4-a000-0887c75b90c8", "sam-server.example.com", "server.example.com",
"ACME_Corporation", "Active",
```


The (optional) JSON file contains the same summary information, as well as the complete list of system facts and product information that is available on the details page.

```
[{"_id":{"_id":"522e0970af5d242094000002"},"_types":
["MarketingProductUsage"],"instance_identifier":"072c8bdd-ca00-43d4-a000-
0887c75b90c8","updated":"2013-09-09T17:46:24Z","splice_server":"sam13-
dlackey-demo","name":"server.example.com","facts":
{"memory_dot_memtotal":"3780964", ...
```

8.6.6. Enhanced Reporting Logs

Reporting Log Sizes

By default, enhanced reporting takes up to 200 MB of additional log space on a system. Logs grow at roughly 750 KB per system per month.

If the logging level needs to be changed, it can be edited in the log configuration file in `/etc/splice/logging/basic.cfg`.

Synchronization Log

All of the errors, messages, and operations for the sync tool are recorded in a specific tool log at `/var/log/splice/spacewalk_splice_tool.log`

8.7. USING SUBSCRIPTION ASSET MANAGER TO GENERATE SATELLITE USAGE REPORTS

Red Hat Satellite 5.6 has a utility (**spacewalk-reports**) to export information on the system inventory, organizations and associated subscriptions, errata, and users. Subscription Asset Manager provides enhanced reporting for Satellite 5.6 through the **spacewalk-splice-tool** reports. The **spacewalk-splice-tool** utility draws on the same Satellite 5.6 data, but parses and presents it in an updated, more detailed way.

Subscription management establishes relationships between systems, organizations, and subscriptions. The types of relationships, and the ways those relationships are described, are slightly different in Satellite when compared to the new subscription services at Red Hat. (This is described more in the *Subscriptions Concepts and Workflows* document.) The new subscription services create a direct relationship between a subscription and the system to which it is applied. In Satellite 5.x, the concept of channels meant that a system was granted access to a content stream and the overall subscription allowed a certain number of systems to have access — but there was no direct association between a subscription and a system.

Satellite enhanced reporting allows Satellite 5.6 servers to sync their system, subscription, and channel data over to a Subscription Asset Manager server. That Subscription Asset Manager server can then take the underlying subscription information and generate a report using Red Hat Subscription Management rules, revealing the relationships between systems and subscriptions in Satellite.

This gives Satellite administrators a greater level of detail and control over the systems in their Satellite inventory.

8.7.1. About Satellite Consolidated Reports

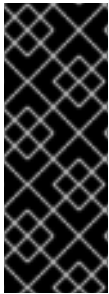
8.7.1.1. The Advantages of Enhanced Reporting

Subscriptions are frequently predicated on attributes of the underlying physical system, such as socket count, RAM, CPU, and cores. For virtual systems, subscriptions can be based on the host/guest relationship and inherited or restricted. Subscriptions can also be related to *other* products installed on the system.

The reporting in Satellite is more limited; it measures the overall counts of systems and subscriptions, but it does not associate subscriptions and systems or identify required subscriptions based on system attributes.

The enhanced reporting provides more detailed reports in two ways:

- Determining actual subscription usage based on system attributes, host/guest relationships, and installed products.
- Tracking *historical* subscription usage based on subscription statuses at different points in time.



IMPORTANT

The enhanced reporting in Subscription Asset Manager only displays data about the Satellite 5.6 systems and organizations. It does not alter, update, or manage any systems, subscription assignments, or organizations in Satellite.

Both system management and subscription management for Satellite 5.6 must be performed in Satellite.

Like the Satellite reports, all of the Subscription Asset Manager report data can be exported to a CSV, so any additional data analysis can be performed. Additionally, a Subscription Asset Manager report can be exported to a JSON file and can be rendered visually in the Subscription Asset Manager UI (including system-level details) so it can be quickly read and interpreted.

8.7.1.2. Differences in Subscription Statuses from Satellite

Because enhanced reporting uses a different set of criteria — system attributes instead of channel access — to calculate the required subscriptions for a system, there are potentially differences between how enhanced reporting reports subscription status for a system and how the same information is reported in Satellite.

For example, many Red Hat Enterprise Linux subscriptions are for two sockets. A four-socket system, then, requires two subscriptions to cover all its socket pairs. If only one subscription is attached to that system, then the Subscription Asset Manager report shows that system as having an insufficient status. However, in Satellite, the same system only consumes a single channel entitlement, regardless of the number of sockets.

Additionally, Satellite has two different types of subscriptions: *system entitlements* (which are required to register a system) and *software channel entitlements* (which is what actually provides access to content, updates, and support). In the new subscription structure (and, thus, in Subscription Asset Manager), the system and channel entitlements are merged into a single product subscription which most closely correlates to channel entitlements.

Lastly, Satellite allows channels to be cloned. If a system is registered with a cloned channel, then no channel entitlements are used, so no entitlements are decremented from the available entitlements pool. However, when the channel information is synchronized, the cloned channel is associated with its original Red Hat channel, and subscriptions are then properly attached to the system (or its status is shown to be invalid or insufficient).

8.7.1.3. Syncing Data from Satellite 5.6 to Subscription Asset Manager

A script is run periodically to sync data from the Satellite 5.6 server to the Subscription Asset Manager database, so the data are accessible for Subscription Asset Manager reports. Only certain Satellite information is synced:

- System information (called system facts in Subscription Asset Manager) including the hostname, socket count, any host/guest relationships, and other relevant attributes
- Satellite organizations and associated subscriptions
- User information, including roles and administrator accounts such as Satellite Administrator and Organization Administrator
- Satellite cloned channels and their associated, originating channel.

Synchronization occurs in two phases. First, the inventory information is pulled out of Satellite as a **spacewalk-reports** report, using the **spacewalk-splice-checkin** process. The information then is sent to the Subscription Asset Manager server. This synchronization step is run every four hours, by default.

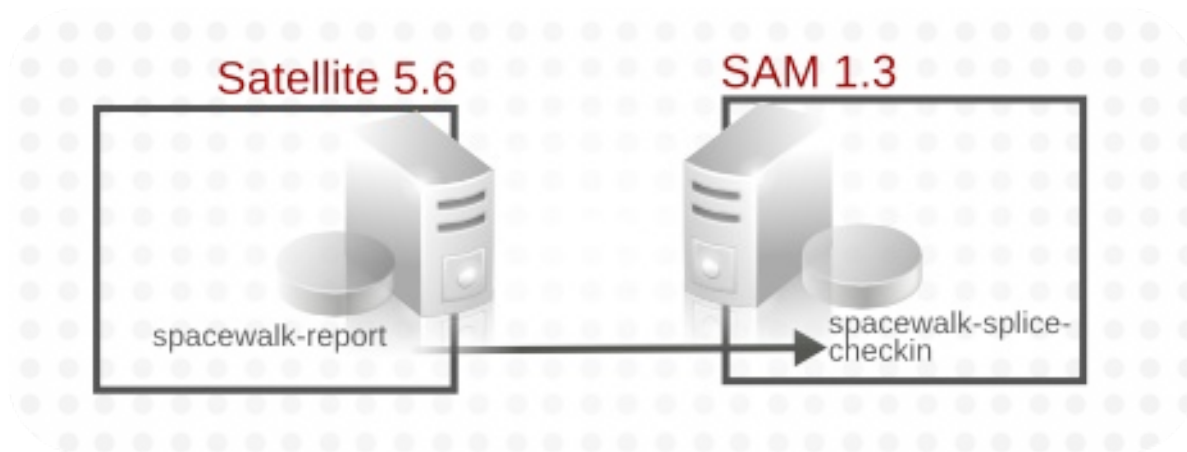


Figure 8.14. Satellite 5.6 to Subscription Asset Manager Sync

From there, the information is collected from Subscription Asset Manager and sent to a dedicated backend database (separate from the normal Subscription Asset Manager database) and stored in the separate reporting server. This step is performed every ten minutes.

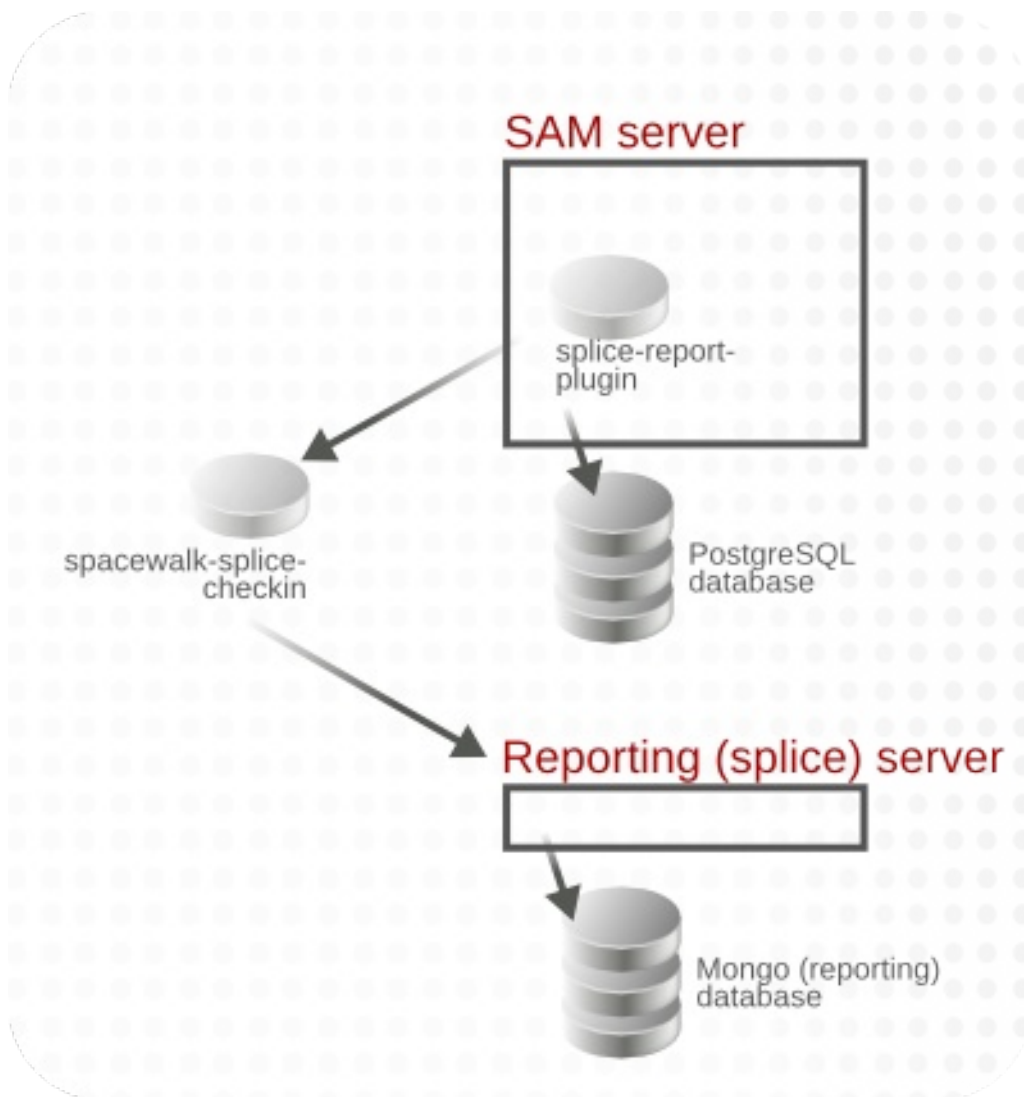


Figure 8.15. Subscription Asset Manager to Reporting Server Sync

After the second phase of synchronization, once the data are stored in the reporting database, then enhanced reports can be run, using the populated system data.

8.7.1.4. Users in Satellite 5.6 and Subscription Asset Manager

As mentioned in [Section 8.7.1.3, “Syncing Data from Satellite 5.6 to Subscription Asset Manager”](#), Satellite are synced over to Subscription Asset Manager and added as Subscription Asset Manager users. All of their organization and role assignments are preserved.

Satellite 5.6 passwords are not synced over to Subscription Asset Manager. Satellite users must log in with their Satellite username and a default password of *CHANGEME*.



NOTE

Satellite users are added to Subscription Asset Manager if they are added to the Satellite server, but they are not deleted in Subscription Asset Manager if they are deleted on the Satellite server. If a Satellite user is deleted, then that account must be manually deleted on the Subscription Asset Manager side.

8.7.2. Prerequisites

When using enhanced reporting, there are some additional system requirements:

- A dedicated Subscription Asset Manager instance specifically for Satellite reporting.



WARNING

A Subscription Asset Manager instance used for enhanced reporting can *only* be used as a reporting server for Satellite. It cannot be used as a regular Subscription Asset Manager instance to manage systems or data could be lost.

- All of the prerequisites in [Section 2.1, “Prerequisites”](#).
- The **crond** service must be running.
- An additional 4 GB of disk space must be available for the reporting database journal.
- Additional packages for the reporting server
 - splice
 - ruby193-rubygem-splice_reports
 - spacewalk-splice-tool

8.7.3. Configuring Reporting

1. Install Subscription Asset Manager as described in [Section 2.2, “Basic Installation and Setup for Subscription Asset Manager”](#), using the additional packages covered in [Section 2.3, “Installing and Configuring Enhanced Reporting”](#).

1. Register the host system. Use the **--autoattach** option to attach the required subscriptions for the operating system immediately.

```
[root@server ~]# subscription-manager register --autoattach
Username: jsmith@example.com
Password:
```

2. Wait several minutes for the updated content repositories to be added to the system configuration.
3. Enable the **[rhel-6-server-sam-rpms]** repository.

```
[root@server ~]# yum-config-manager --enable rhel-6-server-sam-rpms
Loaded plugins: product-id, refresh-packagekit
===== repo: rhel-6-server-sam-rpms
=====
[rhel-6-server-sam-rpms]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl =
```

```
https://cdn.redhat.com/content/dist/rhel/server/6/6Server/x86_64/
subscription-asset-manager/1/os
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/rhel-6-server-sam-rpms
cost = 1000
enabled = 1
enablegroups = True
exclude =
failovermethod = priority
...
```

4. Install the **katello-headpin-all** package using **yum install**:

```
[root@server ~]# yum install -y katello-headpin-all splice
ruby193-rubygem-splice_reports spacewalk-splice-tool
```

This can also be done when installing from an ISO image ([Section 2.2.2, “Installing Through an ISO Image”](#)) by using the **--enhanced_reporting** option.

```
[root@server cdrom]# ./install_packages --enhanced_reporting
```

2. The reporting database is a MongoDB database. Set up the Mongo service on the system to start automatically, and then start the service.

```
[root@sam-server ~]# chkconfig mongod on
[root@sam-server ~]# service mongod start
```

3. Run the configuration script to set up the Subscription Asset Manager server, the default admin user, and the initial organization.

```
[root@server ~]# katello-configure --deployment=sam --
org=Example_Org --user-name=samadmin --user-pass=secret
```

The Subscription Asset Manager admin user is not the same as the Satellite 5.6 admin user.

4. Still on the Subscription Asset Manager machine, create an SSH key to use to authenticate to the Satellite 5.6 machine.

```
[root@sam-server ~]# su - splice -s /bin/sh -c 'ssh-keygen -t rsa -f
/var/lib/splice/id_rsa-sat -N ""'
```

```
Generating public/private rsa key pair.
Your identification has been saved in /var/lib/splice/id_rsa-sat.
Your public key has been saved in /var/lib/splice/id_rsa-sat.pub.
The key fingerprint is:
78:fa:c9:68:71:a2:a7:c1:ec:35:e3:43:ce:27:b7:d8 splice@dhcp129-
162.rdu.redhat.com
```

```
The key's randomart image is:
```

```
+-[ RSA 1024]-----+
|                      |
|                      |
|                      |
|                      |
```

```

|      . S      |
|      o  +o.   |
|      +=+=+    |
|      ..+B0o.  |
|      o++=E.   |
+-----+

```

5. Switch to the Satellite 5.6 machine.
6. Create a new user which can run the required Satellite reports to sent to the Subscription Asset Manager server.

```
[root@sat-server ~]# useradd swreport
```

7. Add the key file that was created on the Subscription Asset Manager machine to the **authorized_keys** file for the **swreport** user on the Satellite 5.6 machine. Include the **command=** option to restrict the **swreport** user to only running Satellite reports on the system.

```
[root@sat-server ~]# vim /home/swreport/.ssh/authorized_keys

command="/usr/bin/spacewalk-report $SSH_ORIGINAL_COMMAND" \
ssh-rsa key_hash swreport@sat-server
```

The **command** directive should be all on one line in the keys file.

8. Set the proper permissions on the **.ssh** directory and the **authorized_keys** file:

```
[root@sat-server ~]# chown -R swreport:swreport /home/swreport/.ssh
[root@sat-server ~]# chmod 700 /home/swreport/.ssh
[root@sat-server ~]# chmod 600 /home/swreport/.ssh/authorized_keys
```

9. Add the **swreports** user to the **apache** system group so that it can connect to the database.

```
[root@sat-server ~]# gpasswd -a swreport apache
```

10. Switch back to the Subscription Asset Manager machine.
11. Switch to the reporting service user (**splice**), and test that the user can SSH into the Satellite machine using the **swreport** key.

```
[root@sam-server ~]# su - splice -s /bin/bash
[splice@sam-server ~]$ ssh -i /var/lib/splice/id_rsa-sat
swreport@sat-server.example.com splice-export
```

Accept the key fingerprint if prompted.

12. Edit the reporting configuration to recognize the Satellite 5.6 server.

```
[root@sam-server ~]# vim /etc/splice/checkin.conf

[spacewalk]
```

```
host=sat-server.example.com
ssh_key_path=/var/lib/splice/id_rsa-sat
login=swreport
```

13. Edit the reporting configuration to use the Subscription Asset Manager administrator password that was set during the Subscription Asset Manager setup.

```
admin-pass=secret
```

14. On the Subscription Asset Manager server, run the sync utility to populate the Subscription Asset Manager database with the Satellite 5.6 data.

```
[root@sam-server ~]# su - splice -s /bin/bash
[splice@sam-server ~]$ spacewalk-splice-checkin
```



NOTE

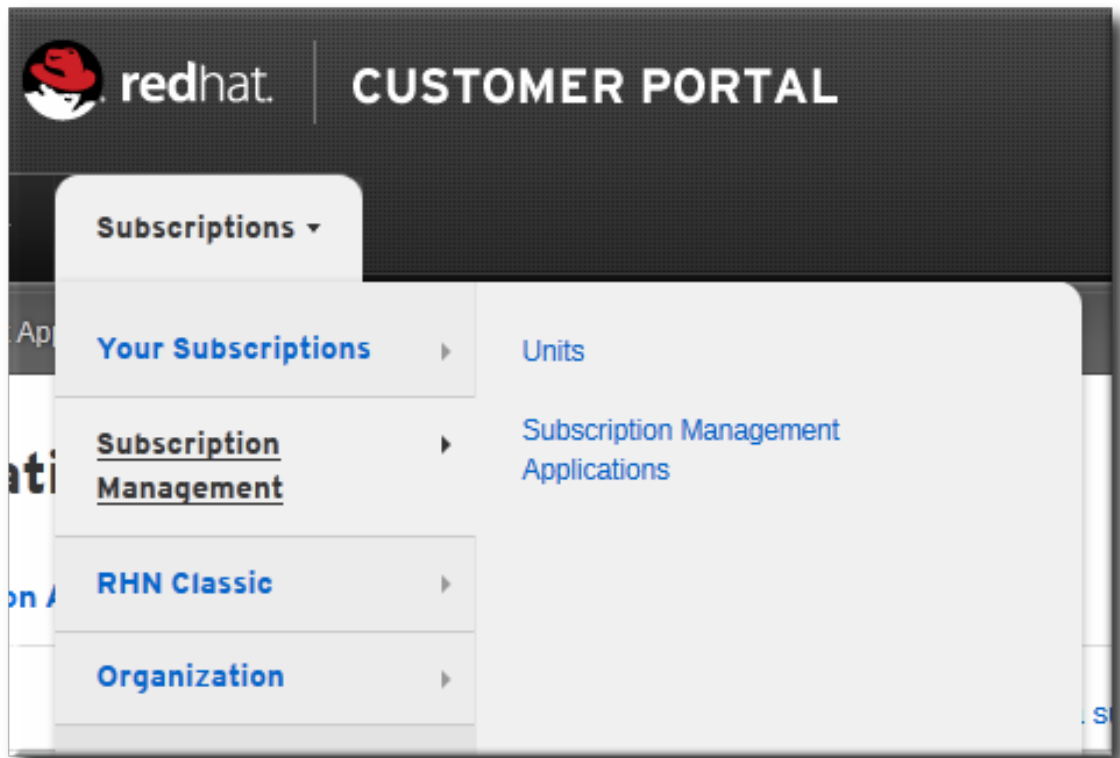
This can a long time to run on the initial sync operation.

To improve the tool performance, set the number of threads for the **spacewalk-splice-tool** process to use. This should be one thread for every two cores on a low-utilization system or one thread for every three cores on a high-utilization system.

For example:

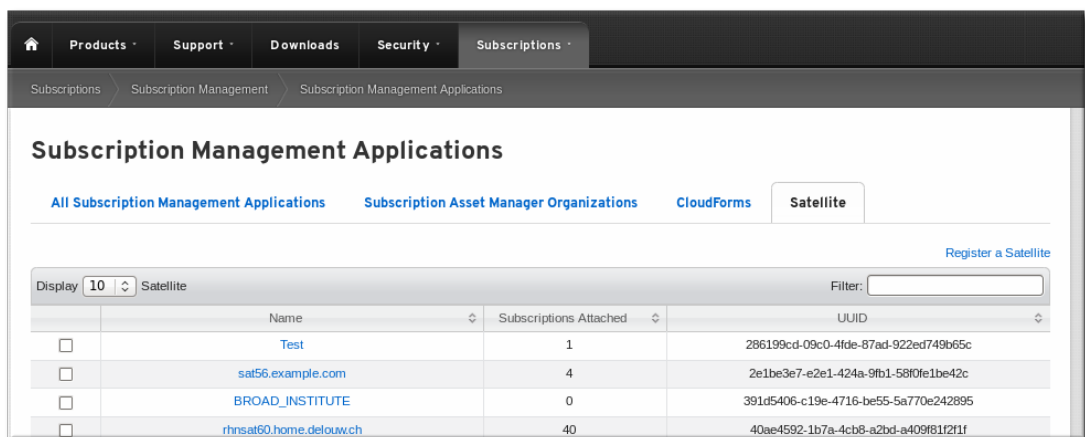
```
[root@sam-server]# /etc/splice/checkin.conf
num-threads=3
```

15. Get the Satellite 5.6 manifest from the Customer Portal.
 1. Log into the Customer Portal.
 2. Expand the **Subscriptions** tab, and select the **Subscription Management > Subscription Management Applications** item.



3. Open the **Satellite** tab.
4. If the Portal entry does not already exist, create the Satellite 5.6 entry and attach the required subscriptions.

1. In the **Satellite** tab, click the **Register a Satellite** link.



2. Fill in the required information for the Satellite 5.6 instance:

- The name for the Satellite server entry.
- The version of the Satellite instance; this should be 5.6.

Subscriptions Subscription Management Register a New Satellite

Register a New Satellite

Name:

Satellite version:

or [Cancel](#)

- Click the **Register** button.
- In the Satellite 5.6 server's **Subscriptions** tab, select the subscriptions to add in the **Available Subscriptions** area.

Subscriptions Subscription Management Subscription Management Applications sat56.example.com

sat56.example.com

Note: download both the manifest and satellite certificate

UUID: 2e1be3e7-e2e1-424a-9fb1-58f0e1be42c
Created: 04/23/2014
Type: Satellite
Version:

Attached Subscriptions [Identity Certificate](#)

[Attach a subscription](#)

Display attached subscriptions Filter:

<input type="checkbox"/>	Subscription Name	Contract Number	Quantity Attached	End Date
<input type="checkbox"/>	90 Day Supported Red Hat JBoss Data Grid Evaluation	10375937	1	05/03/2014
<input type="checkbox"/>	60 Day Self-Supported Red Hat Enterprise Linux OpenStack Platform Preview	10407707	1	05/16/2014
<input type="checkbox"/>	90-day Unsupported Red Hat JBoss BPM Suite Evaluation	10375941	1	05/03/2014
<input type="checkbox"/>	JBoss Business Partner Self-Supported NFR	10283972	1	09/25/2014

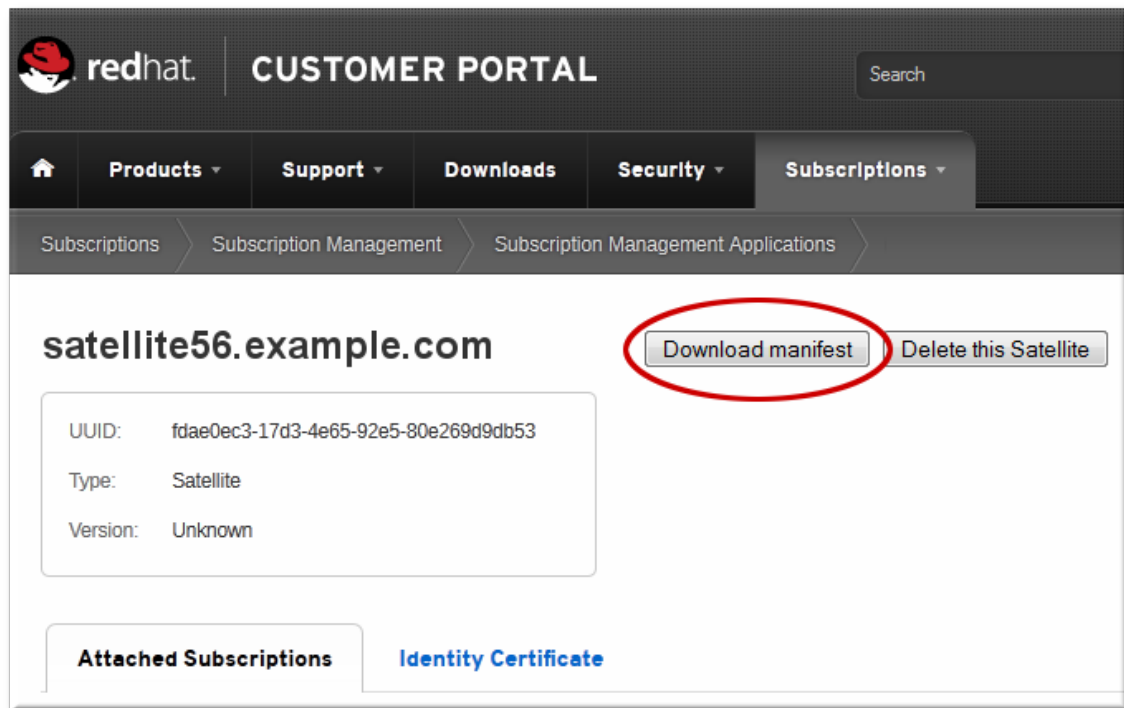
Displaying attached subscriptions 1-4 of 4

Be sure to set the appropriate quantity of subscriptions for each product selected. The quantity is the total number of subscriptions of that type available to the child organization.

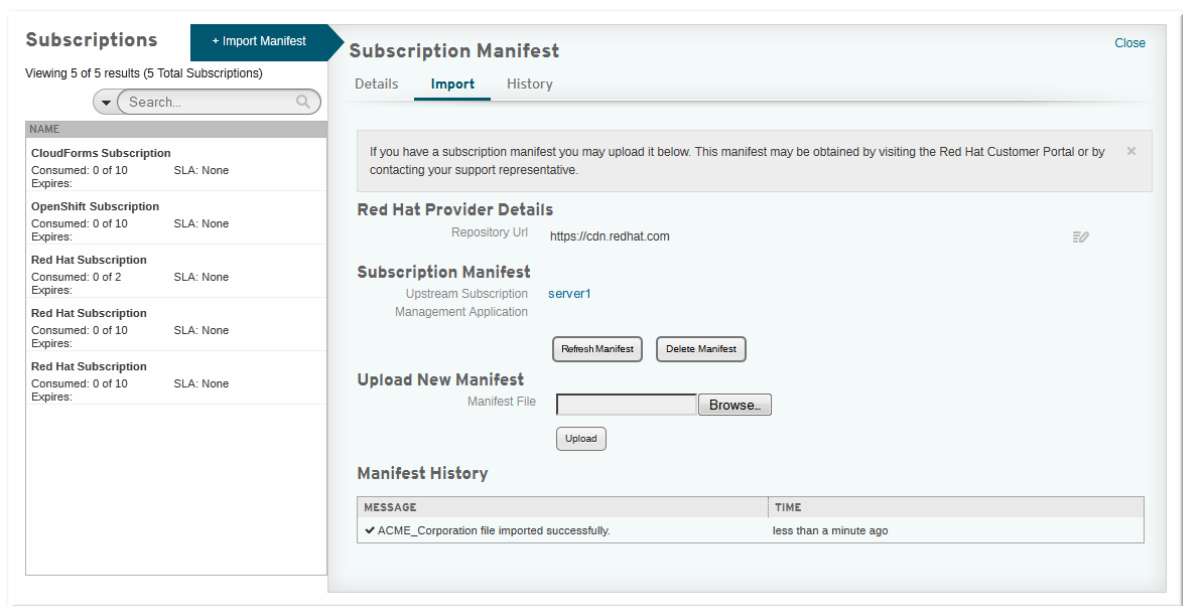
- Scroll down, and click the **Attach** at the bottom of the window.

Attaching subscriptions automatically updates the child organization's manifest.

- In the Satellite 5.6 server's entry page, click the **Download manifest** button, and save the archive file.



16. Log into the Subscription Asset Manager UI (<https://sam-hostname/sam>) as a Satellite administrator, and switch to the appropriate Satellite 5.6 organization.
17. Open the **Subscriptions** > **Subscriptions** tab, and click the **Import Manifest** link.



18. In the middle of the import tab, click browse to navigate to the saved manifest file.
19. Click the **Upload** button.

8.7.4. Running Reports and Getting Results

There is a default report configured which returns information for every organization and every system in every state that is being managed by Subscription Asset Manager.

It is possible to create additional report filters that return specific subsets of information or information for a given period of time. These custom reports are very useful for analyzing usage and compliance trends.

There is a lot more flexibility possible with the report form, however. In particular, there are three versatile settings:

- The organizations to check for the report
- The subscription statuses to include
- The date range to check; this looks for systems which had the status *within the given range*, which may not necessarily be the current status for the system



NOTE

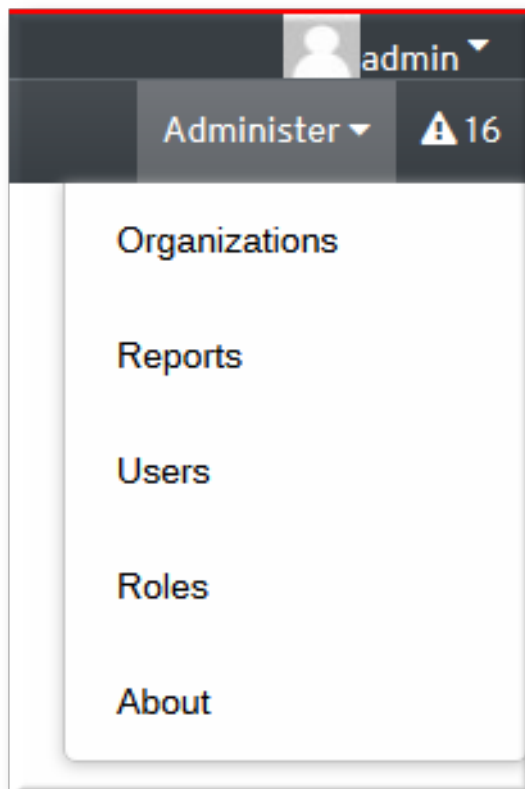
The data within the Subscription Asset Manager database includes historic subscription statuses. This allows reports to be generated to track subscriptions at a given point in time, not just the current date.

For example, if purchasing occurs in July, then a report can be configured to search for insufficient or invalid systems from April through June, to influence purchasing decisions.

The time filter also allows very short windows of time — the previous 24 or 48 hours — to be able to identify and remediate suscription issues immediately.

8.7.4.1. Creating Report Filters

1. Click the **Reports** item in the administration menu.



2. In the left column, click the **New Filter** link.
3. Fill in the required information for the report, including the organizations, statuses, date range, and active states.

Report Filters + New Filter

Viewing 2 of 2 results (2 Total Report Filters)

Search...

NAME

Red Hat Default Report

All

New Filter

Name: Example report

Description: for 2013

Subscription Status: Current Invalid

Satellite Server: sam13-dlackey-demo

Organizations: ACME_Corporation

LifeCycle State: Active Inactive

Search Previous Number of Hours: Search by Date Range:

Start Date: 01/01/2013

End Date: 12/31/2013

Save Filter

4. Click the **Save Filter** button.

8.7.4.2. Running Reports

1. Click the **Reports** item in the administration menu.
2. In the left column, click the name of the report filter to run.
3. Scroll to the bottom of the report page, and click the **Run Report** button.

Export Criteria:

Encrypt export: ☐

Skip including expanded JSON data: ☐

Run Report Export Report

Alternatively, the report results can be exported to a CSV file. To export the data, click the **Export Report** button.

The data are exported to a CSV file and, optionally, a JSON file which contains the system details. These files are contained in a ZIP archive named **report - YEAR-MONTH-DAY-TIMESTAMPZ.zip**.



NOTE

Selecting the **Encrypt export** checkbox means that the exported CSV and JSON files are encrypted and can only be accessed by a private key used by Red Hat support.

8.7.4.3. Subscription Asset Manager Reports Results and Data

The Subscription Asset Manager report returns a chart of all registered systems for all selected organizations, similar to the **Dashboard**.

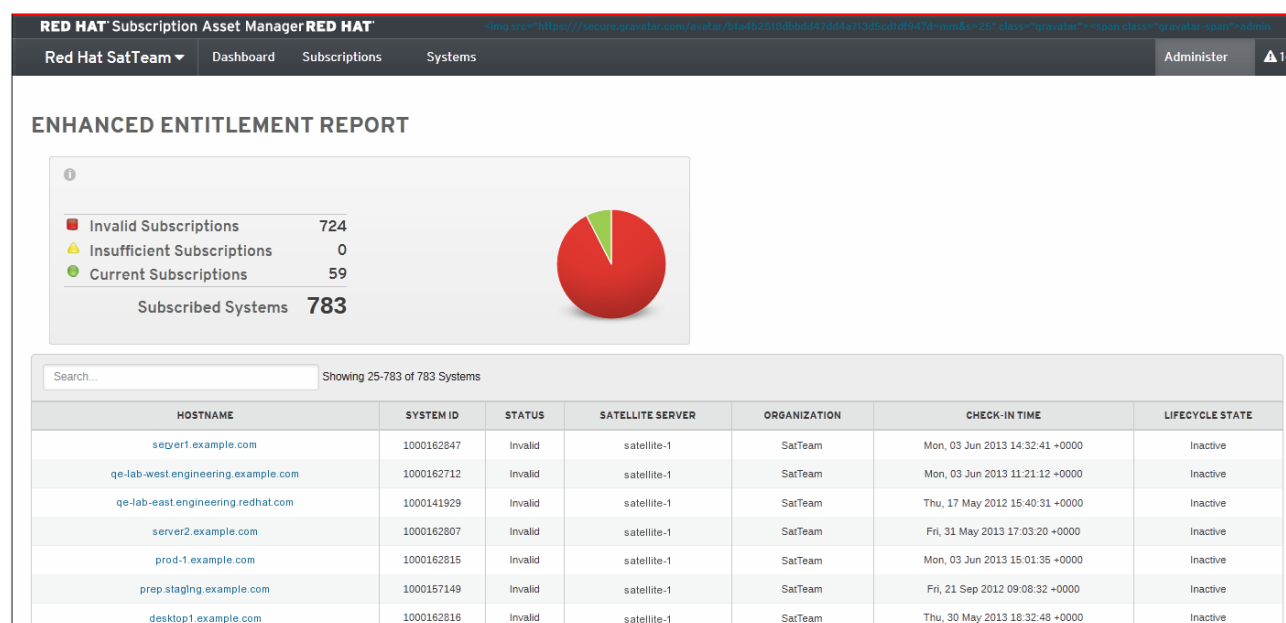


Figure 8.16. The Reports Results

There is also a list of all included systems. The list itself contains summary information such as the Subscription Asset Manager or Satellite server to which the system is registered, its status, its organization, and its most recent checkin time.

Clicking the name of any system pulls up more details subscription data for the system. The details page includes a history of subscription status changes for the given report period, the list of installed products and subscription information for each, and system facts (attributes about the physical machine and operating system, such as CPU, socket count, RAM, and cores).

The screenshot displays the Subscription Asset Manager interface. At the top, a navigation bar includes 'ACME_Corporation', 'Dashboard', 'Subscriptions', 'Systems', and 'Administer'. Below this, a summary box shows subscription status: 0 Invalid Subscriptions, 0 Insufficient Subscriptions, 1 Current Subscriptions, and 1 Subscribed Systems. A large green circle is also present. The main section features a search bar and a table with columns 'HOSTNAME' and 'SYSTEM ID'. The table lists 'server.example.com'. To the right, a detailed view for this system is shown, including tabs for 'Subscription Details', 'Facts', 'Products', 'System Check-Ins', and 'Close'. The 'Subscription Details' tab is active, displaying 'Compliant Product Info: OpenShift Subscription' with start and end dates, and 'Consumed 5 out of 10'.

Figure 8.17. The Reports Results: System Details



NOTE

Only the last 250 check-ins are shown on the system's details page when viewing the enhanced report in the Subscription Asset Manager UI.

When the report results are exported, the same information is included in the export files.

The CSV report contains the summary information in the initial reports page: organization, registered subscription server, hostname, and subscription state, among others.

```
_id, record, CHECK-IN TIME, STATUS, DB ID, SATELLITE SERVER, HOSTNAME,
ORGANIZATION, LIFECYCLE STATE,
{"ident"=>"072c8bdd-ca00-43d4-a000-0887c75b90c8"},
522e0970af5d242094000002, 2013-09-09T14:23:27Z, "Current", "072c8bdd-ca00-
43d4-a000-0887c75b90c8", "sam-server.example.com", "server.example.com",
"ACME_Corporation", "Active",
```

The (optional) JSON file contains the same summary information, as well as the complete list of system facts and product information that is available on the details page.

```
[{"_id":{"$oid":"522e0970af5d242094000002"},"_types":
["MarketingProductUsage"],"instance_identifier":"072c8bdd-ca00-43d4-a000-
0887c75b90c8","updated":"2013-09-09T17:46:24Z","splice_server":"sam13-
dlackey-demo","name":"server.example.com","facts":
{"memory_dot_memtotal":"3780964", ...
```

8.7.5. Troubleshooting Enhanced Reports

8.7.5.1. Enhanced Reporting Logs

Report Log Sizes

By default, enhanced reporting takes up to 200 MB of additional log space on a system. Logs grow at roughly 750 KB per system per month.

If the logging level needs to be changed, it can be edited in the log configuration file in `/etc/splice/logging/basic.cfg`.

Sync Log

All of the errors, messages, and operations for the sync tool are recorded in a specific tool log at `/var/log/splice/spacewalk_splice_tool.log`

8.7.5.2. Common Problems

Q: Why are no systems displayed in the report?

A: First, make sure that there are systems which match the given report filters.

If the filters should return some systems and there are still no systems displayed, this means that the information is not being pulled into the reporting database. There are several potential points of failure:

The information isn't being pulled from the Satellite server.

The information is not being properly transmitted from Subscription Asset Manager into the reporting database.

The information is not being properly stored in the database.

The information stored in Subscription Asset Manager is outdated.

First, make sure that the sync script is running by checking the history in the sync tool log, `/var/log/splice/spacewalk_splice_tool.log`.

Then, make sure that the Mongo service is running and listening on port 27017. If the Mongo service is not running, then the Subscription Asset Manager services cannot start.

```
[root@sam-server ~]# service mongod status
[root@sam-server ~]# telnet localhost 27017
```

If the service is running, check the Mongo database to look for sync entries. For example:


```
[root@sam-server ~]# mongo checkin_service --eval
"printjson(db.marketing_product_usage.count())"
```

If neither of those reveal a problem, or if they do not have relevant entries, then run the reporting debug script:

```
[root@sam-server ~]# /usr/bin/splice-debug
```

This collects all relevant configuration and log files for the reporting server and exports the data to a file in the `/tmp` directory name **splice-debug-YYYY-MM-DD-TIME**. For example, **/tmp/splice-debug-2013-06-14-T15-22-19**.

That directory can be zipped and sent to support if necessary.

Q: Why are all systems marked as invalid?

A: Check that a manifest has been imported into Subscription Asset Manager for the Satellite server. The manifest tells Subscription Asset Manager what subscriptions the Satellite server has attached to it; without the manifest, reporting assumes that no subscriptions are available.

Q: I updated subscriptions for a system or my Satellite server in Subscription Asset Manager, but those changes are not being reflected in the report.

A: The sync script runs every four hours, so it may not have synchronized the changes yet. Either wait for the next scheduled run or run the script by hand (which may take several minutes to finish):

```
[root@sam-server ~]# su - splice -s /bin/bash
[splice@sam-server ~]$ spacewalk-splice-checkin
```

Q: The link to the Satellite 5.6 UI in the report results is returning an HTTP 404 error.

A: Verify that the **rhn-search** process is running on the Satellite 5.6 machine.

8.7.5.3. Other Known Issues

These are some other issues that are recognized with the enhanced reporting and Satellite, but do not have a workaround.

Having Organizations Not Related to Satellite Reporting

If a Subscription Asset Manager instance used in enhanced reporting has non-Satellite organizations added to it, those organizations may be overwritten and removed in the Subscription Asset Manager database as part of the sync process.



WARNING

A Subscription Asset Manager instance used for enhanced reporting can *only* be used as a reporting server for Satellite. It cannot be used a regular Subscription Asset Manager instance to manage systems or data could be lost.

CHAPTER 9. MANAGING SUBSCRIPTION ASSET MANAGER INSTANCES

This chapter covers basic maintenance tasks for Subscription Asset Manager instances, such as backing up and restoring data.

9.1. BACKING UP SUBSCRIPTION ASSET MANAGER

There is no utility to back up all Subscription Asset Manager information. The configuration files must be saved manually, and the backend database data must be dumped to a backup file.

All backup operations must be performed as **root**.

1. Create the desired backup directory. In this example, **umask** is set so that the directory is created with the proper permissions. Then, the directory is added to the **postgres** system group because Subscription Asset Manager uses a PostgreSQL database as its backend.

```
[root@server]# umask 0017
[root@server]# mkdir /backup
[root@server]# chgrp postgres /backup
```

2. Open the backup directory.

```
[root@server]# cd /backup
```

3. Use **tar** or **zip** to create an archive of all of the Subscription Asset Manager configuration files. For example:

```
[root@server]# tar --selinux -czvf config_files.tar.gz \
/etc/katello \
/etc/elasticsearch \
/etc/candlepin \
/etc/gofer \
/etc/grinder \
/etc/pki/katello \
/etc/pki/pulp \
/etc/qpid.conf \
/etc/sysconfig/katello \
/etc/sysconfig/elasticsearch \
/root/ssl-build \
/var/www/html/pub/*
```

4. Create a separate archive for the Elastic Search directory.

```
[root@server]# tar --selinux -czvf elastic_data.tar.gz
/var/lib/elasticsearch
```

5. Back up all of the PostgreSQL databases. The default database names are **katello-schema** and **candlepin**.

If the Subscription Asset Manager instance is not using the default names, then the custom values will be in the **db_name** parameters in the **katello-config.conf** file.

```
[root@server]# grep db_name /etc/katello/katello-configure.conf
```

Run the **pg_dump** command for each database to create a backup. This can take several minutes, depending on the sizes of the databases.

```
[root@server]# su postgres -c "pg_dump -Fc katello-schema > /backup/katello.dump"
[root@server]# su postgres -c "pg_dump -Fc candlepin > /backup/candlepin.dump"
```

The **postgres** service must be running for the **pg_dump** command to work. If the service is not running, then the databases can be backed up by zipping or tarring the PostgreSQL data directory. For example:

```
[root@server]# tar --selinux -czvf postgresql_data.tar.gz /var/lib/pgsql/data/
```

Simply archiving the entire directory backs up all databases. Since all of the databases are shut down, the data directory should only be archived during a maintenance period.

For more information about PostgreSQL backups, consult the **pg_dump** man page or the [PostgreSQL documentation](#).

6. After running **pg_dump**, check that the appropriate **.dump** files have been created in the specified backup directory. For example:

```
# ls /backup
candlepin.dump      config_files.tar.gz  elastic_data.tar.gz
katello.dump
```

9.2. RESTORING SUBSCRIPTION ASSET MANAGER



WARNING

Restoring Subscription Asset Manager overwrites all existing data for the Subscription Asset Manager instance.

Be certain to restore the proper Subscription Asset Manager instance.



IMPORTANT

This procedure assumes that the Subscription Asset Manager server is being restored on the same machine where it was previously installed and backed up.

If the Subscription Asset Manager instance is being restored on a different host, then that system must have the same configuration as the original system, including the same hostname and IP address.

All restore operations must be performed as **root**.

1. Create a configuration file with the same options as the original instance using the file from the configuration backup, then run the **katello-configure** script.

```
[root@server ~]# katello-configure --answer-  
file=/etc/katello/katello-configure.conf
```

2. Open the directory containing the backup files. For example:

```
[root@server ~]# cd backup/
```

3. Check the database names. The default database names are **katello-schema** and **katello-pin**. If the Subscription Asset Manager instance is not using the default names, then the custom values will be in the **db_name** parameters in the **katello-configure.conf** file.

```
[root@server backup]# grep db_name /etc/katello/katello-  
configure.conf
```

4. Stop all services prior to restoring the databases:

```
[root@server backup]# katello-service stop
```

5. Restore the system files by extracting the archived files and directories. For example:

```
[root@server backup]# tar --selinux -xzf config_files.tar.gz -C /  
[root@server backup]# tar --selinux -xzf elastic_data.tar.gz -C /
```

6. Drop the existing Subscription Asset Manager PostgreSQL databases if any exist:

```
[root@server backup]# service postgresql start  
[root@server backup]# su postgres -c "dropdb katello-schema"  
[root@server backup]# su postgres -c "dropdb katello-pin"
```

If all the processes are not stopped first, then the database drop operation can return this error:

```
database xxx is being accessed by other users
```

7. As the **postgres** user, run the **pg_restore** command to restore the databases.

```
[root@server backup]# su postgres -c "pg_restore -C -d postgres  
/backup/katello.dump"  
[root@server backup]# su postgres -c "pg_restore -C -d postgres  
/backup/katello-pin.dump"
```

For more info about PostgreSQL restore operations, see the **pg_dump** man page or the [PostgreSQL documentation](#).

8. Restart all Subscription Asset Manager processes:

```
[root@server backup]# katello-service restart
```

9. Verify that all services are responding properly by checking the `/var/log/katello/production.log` log file for errors and the `/var/log/candlepin/audit.log` file for denials.

Attempt to ping the Subscription Asset Manager instance:

```
[root@server backup]# katello -u admin -p admin ping
-----
-----
Katello Status
-----
Status Service      Result Duration Message
-----
OK
candlepin           OK       74ms
candlepin_auth      OK       38ms
elasticsearch        OK       37ms
katello_jobs         OK       39ms
```

9.3. REGENERATING AND REPLACING CA CERTIFICATES

There may be a situation when it is necessary to replace the existing CA certificate used for the subscription service. In that case, every system which uses Subscription Asset Manager as its subscription service must be updated to use that new certificate.

This is done by generating a new RPM containing the certificate file and then distributing that RPM to the client systems.

1. *On the Subscription Asset Manager server*, generate a new certificate RPM to install on the clients.

This should all be on a single line; each argument is broken out to show the required options.

```
[root@sam-server ~]# /usr/share/katello/certs/gen-rpm.sh
--name "candlepin-cert-consumer-${hostname}"
--version 1.4
--release 2
--packager None
--vendor None
--group 'Applications/System'
--summary "Subscription-manager consumer certificate for Katello
instance ${hostname}"
--description 'Consumer certificate and post installation script
that configures rhsm.'
--requires subscription-manager
--post /root/ssl-build/rhsm-katello-reconfigure
/etc/rhsm/ca/candlepin-local.pem:644=/root/ssl-build/candlepin-
cert.crt 2>>/var/log/katello/katello-configure/certificates.log &&
/sbin/restorecon ./*rpm
```

This script generates a new RPM containing the new Subscription Asset Manager certificate file and sets up the RPM to install that certificate and properly configuration the Red Hat Subscription Manager client on each system. The arguments for the bash script define the RPM settings:

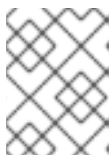
- **--name**, **--version**, and **--release** set the name of the RPM, in the form *name.version-release.rpm*.
 - **--vendor** and **--package** are required for the RPM information, but the values do not matter.
 - **--group** specifies the type of application or package which the RPM installs.
 - **--summary** and **--description** just set information about the RPM.
 - **--requires** sets packages that must be available or installed before this RPM can be installed. Since this RPM configures the local Red Hat Subscription Manager client, the **subscription-manager** package is required.
 - **--post** runs a given command, script, or series of command once the RPM package is installed. In this case, it configures the local Red Hat Subscription Manager client to use the specified Subscription Asset Manager server as its subscription service and sets the required connection and certificate properties in the Red Hat Subscription Manager configuration file.
2. Delete any existing certificate RPMs in the **/var/www/html/pub** directory on the server machine.
 3. Copy the newly-generated RPM to the **/var/www/html/pub** directory so it can be downloaded via HTTP.
 4. *On each Subscription Asset Manager client system*, download and install the new RPM. For example:

```
[root@server1 ~]# rpm -ivh http://SAM_server_hostname/pub/candlepin-  
cert-consumer-SAM_server_hostname.noarch.rpm
```

9.4. SETTING UP SUBSCRIPTION ASSET MANAGER ON A MULTIHOMED SYSTEM

SSL certificates rely on the hostname of the server to identify and authenticate the server. When Subscription Asset Manager is running on a multi-homed instance, each interface requires its own certificate for secure connections to succeed.

1. Configure a self-signed certificate authority (CA) to use to generate the server certificates.



NOTE

If a CA is already configured or a third-party CA is available to issue certificates, then this step can be skipped.

1. Create the appropriate directory and subdirectories in **/etc/pki**. This example creates **myCA**.

This command is split onto multiple lines for clarity; this should be run on the same line.

```
[root@server1 ~]# mkdir -m 0755 \  
/etc/pki/myCA \
```

```
/etc/pki/myCA/private \  
/etc/pki/myCA/certs \  
/etc/pki/myCA/newcerts \  
/etc/pki/myCA/crl
```

2. Copy the OpenSSL configuration file to the new PKI directory, and set the appropriate permissions.

```
[root@server1 ~]# cp /etc/pki/tls/openssl.cnf  
/etc/pki/myCA/openssl.my.cnf  
[root@server1 ~]# chmod 0600 /etc/pki/myCA/openssl.my.cnf
```

3. Create an index file.

```
[root@server1 ~]# touch /etc/pki/myCA/index.txt
```

4. Create a serial number file and seed it with the starting serial number.

```
[root@server1 ~]# echo '01' > /etc/pki/myCA/serial
```

5. Open the CA directory.

```
[root@server1 ~]# cd /etc/pki/myCA/
```

6. Generate the self-signed CA certificate. This prompts for information such as the location and company to use in the certificate's subject name.

```
[root@server1 myCA]# openssl req -config openssl.my.cnf -new -  
x509 -extensions v3_ca -keyout private/myca.key -out  
certs/myca.crt -days 1825
```

```
-----
```

```
Country Name (2 letter code) [XX]:US  
State or Province Name (full name) []:North Carolina  
Locality Name (eg, city) [Default City]:Raleigh  
Organization Name (eg, company) [Default Company Ltd]:Example  
Organizational Unit Name (eg, section) []:Engineering  
Common Name (eg, your name or your server's hostname) []:server1  
Email Address []:admin@example.com
```

```
-----
```

7. Edit the OpenSSL configuration file to use the location of the new CA directory (**/etc/pki/myCA/**).

```
[root@server1 myCA]# sed -i "s/\etc\/pki\/CA\/\etc\/pki\/myCA\/"  
openssl.my.cnf
```

8. Create a multi-homed CA configuration file.

```
[root@server1 myCA]# cp /etc/pki/tls/openssl.cnf  
/etc/pki/myCA/openssl.my_multihome.cnf
```


9. Edit the multi-homed configuration file.

- Change the CA directory to **/etc/pki/myCA**.

```
dir = /etc/pki/myCA
```

- Uncomment the **copy_extensions** line and set the value to **copy**.

```
copy_extensions = copy
```

- Uncomment the required extensions line and enable version 3 extensions for requests.

```
req_extensions = v3_req
```

- Add the subjectAltNames extension, and add the list of other interfaces on the system.

```
subjectAltName = @alt_names
```

```
[alt_names]
DNS.1 = server2
DNS.2 = server3
```

2. Create and sign the server certificate for Subscription Asset Manager.

1. Create the certificate request for Subscription Asset Manager, using the multi-homed configuration file. As with creating the CA certificate, it prompts for information (such as locality and organizational units) to use to build the subject name of the certificate

```
[root@server1 myCA]# openssl req -config openssl.my_multihome.cnf
-new -nodes -keyout private/server.key -out server.csr -days 365
```

```
-----
```

```
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:North Carolina
Locality Name (eg, city) [Default City]:Raleigh
Organization Name (eg, company) [Default Company Ltd]:Example
Organizational Unit Name (eg, section) []:Engineering
Common Name (eg, your name or your server's hostname) []:server1
Email Address []:admin@example.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
```

```
A challenge password []:
An optional company name []:
```

```
-----
```

2. Set the owner and permissions for the generated keys.

```
[root@server1 myCA]# chown root.apache
/etc/pki/myCA/private/server.key
```

```
[root@server1 myCA]# chmod 0440 /etc/pki/myCA/private/server.key
```

3. Sign the certificate, using the multi-homed configuration file.

```
[root@server1 myCA]# openssl ca -config openssl.my_multihome.cnf  
-keyfile ./private/myca.key -cert ./certs/myca.crt -policy  
policy_anything -out certs/server.crt -infiles server.csr
```

4. Verify that the certificate is valid.

```
[root@server1 myCA]# openssl verify -purpose sslserver -CAfile  
/etc/pki/myCA/certs/myca.crt /etc/pki/myCA/certs/server.crt
```

3. Set up the Subscription Asset Manager web server with the new certificates.

1. Copy the certificates into the Subscription Asset Manager certificate directory.

```
[root@server1 myCA]# /etc/pki/myCA/certs/server.crt  
/etc/pki/katello/  
[root@server1 myCA]# /etc/pki/myCA/certs/myca.crt  
/etc/pki/katello/  
[root@server1 myCA]# /etc/pki/myCA/certs/server.key  
/etc/pki/katello/
```

2. Edit web server configuration file to point to the new certificates.

```
[root@server1 myCA]# vim /etc/httpd/conf.d/katello.conf  
  
SSLCaCertificateFile /etc/candlepin/certs/candlepin-ca.crt  
SSLCertificateFile /etc/pki/katello/server.crt  
SSLCertificateKeyFile /etc/pki/katello/server.key
```

3. Restart the Subscription Asset Manager services to load the new certificate information.

```
[root@server1 myCA]# service tomcat6 restart && service pulp-  
server restart && service katello restart && service katello-jobs  
restart
```

4. Copy the CA certificate into the **pub/** directory for the web server so that clients can download it.

```
[root@server1 myCA]# cp /etc/pki/myCA/certs/myca.crt  
/var/www/html/pub/
```

5. Import the CA certificate into the browser used to access the SAM web UI.
4. Set up new certificate RPMs to use to install the proper configuration and certificates on clients.
 1. Open the **pub** directory.

```
[root@server1 myCA]# cd /var/www/html/pub
```

2. Edit the Red Hat Subscription Manager configuration files to use the newly-created CA.

```
[root@server1 pub]# sed 's/scandlepin-local/smyca/' ~/ssl-build/rhsm-katello-reconfigure > ~/ssl-build/rhsm-katello-reconfigure-myca
```

3. Generate new client RPMs with the new certificate and the updated **rhsm.conf** file.

```
[root@server1 pub]# SERVER_NAMES="server1 server2 server3"
[root@server1 pub]# for KATELLO_SERVER in $SERVER_NAMES; do sed
"s/KATELLO_SERVER=.* /KATELLO_SERVER=${KATELLO_SERVER}/" ~/ssl-build/rhsm-katello-reconfigure-myca > ~/ssl-build/rhsm-katello-reconfigure-myca-${KATELLO_SERVER};/usr/share/katello/certs/gen-rpm.sh --name "candlepin-cert-consumer-${KATELLO_SERVER}" --version 1.0 --release 2 --packager None --vendor None --group 'Applications/System' --summary "Subscription-manager consumer certificate for Katello instance ${KATELLO_SERVER}" --description 'Consumer certificate and post installation script that configures rhsm.' --post /root/ssl-build/rhsm-katello-reconfigure-myca-${KATELLO_SERVER} /etc/rhsm/ca/candlepin-local.pem:666=/root/ssl-build/candlepin-cert.crt /etc/rhsm/ca/myca.pem:666=/etc/pki/myCA/certs/myca.crt && /sbin/restorecon ./ *rpm; done
```

This creates a new RPM for each interface in the multi-homed configuration. For example:

```
./candlepin-cert-consumer-server1.noarch.rpm
./candlepin-cert-consumer-server1.src.rpm
./candlepin-cert-consumer-server2.noarch.rpm
./candlepin-cert-consumer-server2.src.rpm
./candlepin-cert-consumer-server3.noarch.rpm
./candlepin-cert-consumer-server3.src.rpm
```

5. Install the updated RPMs, for each interface, on all of the Subscription Asset Manager clients.

```
[root@sam-client ~]# yum -y install http://server1/pub/candlepin-cert-consumer-server1.noarch.rpm
[root@sam-client ~]# yum -y install http://server2/pub/candlepin-cert-consumer-server2.noarch.rpm
[root@sam-client ~]# yum -y install http://server3/pub/candlepin-cert-consumer-server3.noarch.rpm
```

9.5. SUBSCRIPTION ASSET MANAGER LOG AND FILE LOCATIONS

Subscription Asset Manager is comprised of different components for its different services, such as the subscription application, different interfaces, and certificate services. Each component has its own logs.

Table 9.1. Different Logs for Subscription Asset Manager

Log Location	Description
/var/log/katello/katello-configure/main.log	Installation logs.

Log Location	Description
<code>/var/log/tomcat6/catalina.out</code>	Logs for the subscription services, which run as Tomcat applications.
<code>/var/log/katello/production.log</code>	Logs for the Subscription Asset Manager UI and REST API.
<code>/var/log/katello/thin-log.<i>port#</i>.log</code>	Output from the thin server, one per active port.
<code>/var/log/thumbslug/error.log</code>	An errors log for the certificate proxy.

CHAPTER 10. RED HAT ACCESS PLUG-IN

10.1. RED HAT ACCESS PLUG-IN

The Red Hat Access pre-installed plug-in lets you access several Red Hat Customer Portal services from within the Red Hat SAM web interface.

The Red Hat Access plug-in provides the following services:

- **Search:** Search solutions in the Customer Portal from within the Red Hat SAM interface.
- **Support:** Access your open support cases, modify an open support case, and open a new support case from within the Red Hat SAM interface.



NOTE

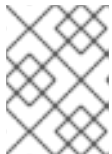
To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

10.1.1. Searching for Solutions in the Red Hat Access Plug-in

The Red Hat Access plug-in provides search capabilities that look through the solutions database available in the Red Hat Customer Portal without needing to log in to the Customer Portal interface.

To search for solutions from the Red Hat SAM Server:

1. In the top right, click **Red Hat Access** → **Search**.
2. To log into the Red Hat Customer Portal: In the main panel top right, click **Log In**.



NOTE

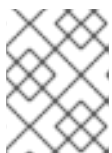
To access Red Hat Customer Portal resources, you need to log in with your Red Hat Customer Portal user identification and password.

3. In the **Red Hat Search:** field, enter your search query. Search results display in the left-hand **Recommendations** list.
4. In the **Recommendations** list, click a solution. The solution article displays in the main panel.

10.1.2. Viewing Existing Support Cases Using the Red Hat Access Plug-in

To view existing support cases from the Red Hat SAM Server:

1. In the top right, click **Red Hat Access** → **Support** → **My Cases**.
2. In the main panel top right, click **Log In** to log into the Red Hat Customer Portal. If you are already logged in, skip this step.



NOTE

To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

3. To search for a specific support case from existing cases, do any of the following:
 1. In the **Search** field, provide a key word or phrase.
 2. From the drop-down list, choose a specific **Case Group**. Your organization has defined **Case Groups** inside the Red Hat Customer Portal.
 3. Choose a Case Status.
4. From the results, choose a specific support case and click the **Case ID**. The support case is ready to view.

10.1.3. Modifying Existing Support Cases Using the Red Hat Access Plug-in

Prerequisites

Complete the instructions from the previous section.

Update Support Cases from the Red Hat SAM Server web interface. When viewing the support case, scroll down to the sections marked to do the following:

- **Attachments:** - Attach a local file from the system. Add a filename to make it easier to identify.



NOTE

Filenames must have less than 80 characters. The maximum file size for web uploaded attachments is 250 MB. Use FTP for larger files.

- **Case Discussion:** - Add any updated information about the case you wish to discuss with Global Support Services. After adding information, click **Add Comment**.

10.1.4. Creating New Support Cases Using the Red Hat Access Plug-in

1. In the top right, click **Red Hat Access** → **Support** → **New Case**.
2. In the main panel top right, click **Log In** to log into the Red Hat Customer Portal. If you are already logged in, skip this step.



NOTE

To access Red Hat Customer Portal resources, you must log in with your Red Hat Customer Portal user identification and password.

3. The **Product** and **Product Version** fields are automatically populated. Complete the other relevant fields, as follows:
 - **Summary:** - Provide a brief summary of the issue.
 - **Description:** - Write a detailed description of the issue.

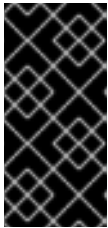
**NOTE**

Based on the summary, recommendations for possible solutions display in the main panel.

4. Click **Next**. A second screen displays.
5. Choose the appropriate options, as follows:
 - **Severity**: Select the ticket urgency as 4 (low), 3 (normal), 2 (high) or 1 (urgent).
 - **Case Group**: Based on who needs to be notified, create case groups associated with the support case. Select Case Groups in Red Hat Satellite. Create Case Groups within the Customer Portal.
6. Attach any required files. Add a file description and click **Attach**.

To ensure you provide relevant information, it is recommended that you attach the output of the following commands:

```
# sosreport
# foreman-debug
```

**IMPORTANT**

foreman-debug removes all security information such as password, tokens and keys while collecting information. However, the tarball can still contain sensitive information about the Red Hat Satellite Server. It is recommended to send this information directly to the intended recipient and not publicly.

**NOTE**

Filenames must have less than 80 characters. The maximum file size for web uploaded attachments is 250 MB. Use FTP for larger files.

7. Click **Submit**. The system uploads the case to the Customer Portal, and provides a case number for your reference.

APPENDIX A. REVISION HISTORY

Revision 1.3-20	March 25, 2015	Jo Somers
SAM 4.1 Installing Subscription Asset Manager: Added section 2.2.2.1: Configuring Red Hat SAM Manually with an HTTP Proxy		
Revision 1.3-19	March 23, 2015	Jo Somers
SAM 4.1 Installing Subscription Asset Manager Prerequisites: Added 6.6 or higher, as follows: Red Hat Enterprise Linux 6.6 or later Server, 64-bit		
Revision 1.3-18	March 05, 2015	Jo Somers
Added new chapter: SAM User Interface Plug-ins and added Installing SAM through an ISO Image		
Revision 1.3-17	April 13, 2014	Ella Deon Ballard
Updating instance-based and virtual setup sections.		
Revision 1.3-14	October 1, 2013	Deon Ballard
New content and reorganization for the SAM 1.3 release.		