**Redhat Satellite**

I understand that in pulp you can create repos and it gives you a good level of control.

I am getting into SpaceWalk and from the looks of it, it does what Pulp does but more?

Not sure what the differences are.

Can someone please give me from experience a high level explanation of the differences?

EDIT: I just found this info and it looks like Pulp is incorporated into SpaceWalk?

Red Hat Satellite 6 is a re-engineering of the Red Hat Satellite product. Spacewalk (and its downstream product, Red Hat Satellite 5) is a systems-management tool, and Satellite 6 is designed to accomplish the same set of goals. However, Satellite 6 has been built from the ground-up on different and more modern technologies. Spacewalk is not the upstream for Satellite 6, and never will be. Instead Satellite 6 is a federation of several upstream open source projects, including Katello, Foreman, **Pulp**, and Candlepin.

<http://spacewalk.redhat.com/faq.html>

Thank you:)

7 comments

88% Upvoted

This thread is archived

New comments cannot be posted and votes cannot be cast

Sort by

level 1

Comment deleted by user[2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbpqcug/)

level 2

[meltingacid](https://www.reddit.com/user/meltingacid)

1 point · [2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbr4icn/)

Have you used katello? How do you find the product?

level 1

[darthcrypto](https://www.reddit.com/user/darthcrypto)

1 point · [2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbr6y3c/)

So Pulp is used for content management. Real world example would be you store your rpms on a Pulp server so that you don't need to reach out to the internet to download packages. So instead of yum installing from a mirror on the web, your yum would point to your pulp server and you would download packages from there. Now obviously your Pulp server would need to have alot of packages to support a development team so you could design your Pulp server to sync with a mirror on the web (very easy to setup). And then for any other obscure packages you could just manually import them onto Pulp.

Spacewalk is designed for managing your systems: pushing out updates, kickstart files, configurations, etc.

level 2

[deleted]

1 point · [2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbrbm28/)

So basically Spacewalk does what pulp does and then some. Yes?

level 3

[meltingacid](https://www.reddit.com/user/meltingacid)

1 point · [2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbrgqwz/)

Not OP, but I would say you are right.

level 4

[CatPosingAsUnixadm](https://www.reddit.com/user/CatPosingAsUnixadm)

1 point · [2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbsl9i5/)

Kind of close, but not entirely. Pulp only does repo management. You can have repos in spacewalk + the stuff [/u/darthcrypto](https://www.reddit.com/u/darthcrypto) mentioned above.

If you want a closer comparison to spacewalk, you may want to check out foreman (<https://theforeman.org/>). Pulp is used as a subcomponent in foreman. *disclaimer* foreman is an absolute beast and has a ton of features, many of which you might not need.

Another big difference between Spacewalk repo management and pulp is that spacewalk is synchronous and pulp is asynchronous. You can get content downloaded MUCH faster with pulp......now if we could just get .deb support in pulp all of my dreams would come true.

level 5

[meltingacid](https://www.reddit.com/user/meltingacid)

1 point · [2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbsljfw/)

Out of curiosity and I haven't used foreman...what features make foreman a beast :)

level 6

[CatPosingAsUnixadm](https://www.reddit.com/user/CatPosingAsUnixadm)

1 point · [2 years ago](https://www.reddit.com/r/linuxadmin/comments/5km7qn/what_is_the_difference_between_pulp_and_spacewalk/dbsml85/)

Foreman is a beast because it's a complete lifecycle management solution. It can do everything that Spacewalk can do, and so much more.

You can think of it as modular, with many optional plugins. Here's a full list: <https://theforeman.org/plugins/>

Katello is not a plugin, but rather it interfaces with foreman. Anything you're doing on the client side with 'subscription-manager' to register servers into foreman/katello is exclusive to katello. Going from Spacewalk --> Katello/Foreman also means switching from RHN to subscription-manager.

You can install foreman on it's own and it will do kickstarts and can also deploy servers to AWS and other cloud services. Foreman can also integrate with config management systems like puppet, saltstack, chef, and others.

If you install *katello* it comes with foreman prepackaged as a dependency.

Some things I ran in to that were difficult (not impossible, just took alot of man hours to figure out) were the repository management in katello with content views and installing custom ssl certificates for the web front-end to foreman.

In short, you can make foreman/katello perform miracles.....if you have enough time and patience.

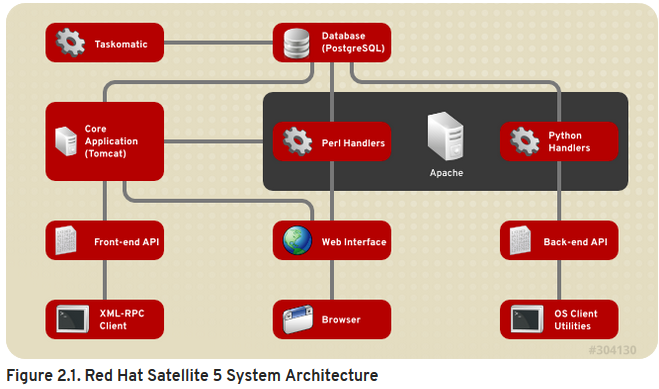
**Comparing Satellite 5 and Satellite 6**

This chapter explores the differences between Red Hat Satellite 5 and Red Hat Satellite 6, especially in the areas of system architecture, organizational structures, and the application life cycle.

The following table outlines some initial concepts and their respective implementation in both Satellite 5 and Satellite 6.

**Table 2.1. Comparison of Satellite 5 and Satellite 6 Concepts**

| **Concept**  **Description** |  | **Satellite 5** | **Satellite 6** |
| --- | --- | --- | --- |
| Open source projects  A single project approach versus a modular approach. | | Spacewalk | Foreman, Katello, and Puppet |
| Subscription types  Pool- or channel-based versus certificate-based. Subscription management has improved over the years from a pool- or channel-based approach to a more specific certificate-based approach. Certificate-based subscription management provides better overall control of subscriptions of those consuming clients. | | Entitlements | Subscriptions |
| Subscription methods (or Satellite subscription consumption).  The way that Satellite is enabled to synchronize and distribute Red Hat content. Certificates are activated during installation; manifests are uploaded after installation. | | Certificate file | Manifest file |
| Organization management  Both Satellite 5 and 6 have a concept of multiple organizations, but Satellite 6 also includes functionality to include the context of the location. | | Organizations | Organizations and Locations |
| Software and configuration content  Distributed through channels versus distributed through content views published and promoted through environments. In Satellite 6 a content view contains a chosen set of software repositories and configuration modules that are published and promoted to an environment. Client systems consume its software and configurations through its environment associations. | | Software Channels | Products and repositories |
| Configuration |  | Configuration Channels | Puppet Repositories |
| Proxy services |  | Red Hat Satellite Proxy Server | Red Hat Satellite Capsule Server |
| Command-line tools |  | Various CLI tools | Hammer |
| Virtualization and cloud providers |  | KVM and Xen | Openstack, Red Hat Enterprise Virtualization, KVM, VMware, EC2 |
| Database support |  | Embedded PostgreSQL, managed PostgreSQL, external PostgreSQL, Oracle Database 10g Release 2 or 11g (Standard or Enterprise Edition) | Embedded PostgreSQL for 6. |



Web UI

The Web UI runs through an Apache web server and provides the main entry point for Satellite operations.

Front-end API

The Front-end API provides the ability to interact with Satellite 5 through an XML-RPC API. This allows system administrators to write scripts to perform repetitive tasks, or develop third-party applications around Satellite. The Front-end API exposes most of the Web UI functionality using XML-RPC.

Back-end API

The back end provides a set of APIs that the different client utilities (rhn\_register, yum) connect to. These are not documented and are used solely by the client utilities.

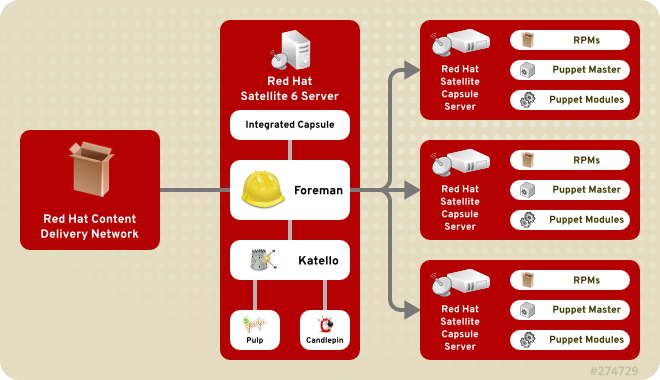
Taskomatic

Taskomatic is generally a separate service within Red Hat Satellite 5 that runs various asynchronous jobs, such as cleaning up the sessions table, or sending email notifications for new errata. The majority of these jobs run periodically, and you can adjust the frequency with which they occur.

Search Server

Satellite contains a standalone search server that runs as a daemon that allows you to quickly find a system, package, or errata, as opposed to paging through hundreds of items in a list. It uses Apache's Lucene search engine library, which provides more relevant search results and a richer query language.

Red Hat Satellite 6 is based upon several open source projects arranged in the following architecture.



**Figure 2.2. Red Hat Satellite 6 System Architecture**

Foreman

Foreman is an open source application used for provisioning and life cycle management of physical and virtual systems. Foreman automatically configures these systems using various methods, including kickstart and Puppet modules. Foreman also provides historical data for reporting, auditing, and troubleshooting.

Katello

Katello is a subscription and repository management application. It provides a means to subscribe to Red Hat repositories and download content. You can create and manage different versions of this content and apply them to specific systems within user-defined stages of the application life cycle.

Candlepin

Candlepin is a service within Katello that handles subscription management.

Pulp

Pulp is a service within Katello that handles repository and content management.

Hammer

Hammer is a CLI tool that provides command line and shell equivalents of most Web UI functions.

REST API

Red Hat Satellite 6 includes a RESTful API service that allows system administrators and developers to write custom scripts and third-party applications that interface with Red Hat Satellite.

Capsule

Red Hat Satellite Capsule Server acts as a proxy for some of the main Satellite functions including repository storage, DNS, DHCP, and Puppet Master configuration. Each Satellite Server also contains integrated Capsule Server services.

# Satellite Proxies and Capsules

A key difference between Satellite 5 and Satellite 6 is in the area of "disconnected" content management. Both versions of Satellite can provision and keep hosts synchronized without direct connection to the internet, but the way they achieve this is slightly different.

### 2.2.1. Satellite 5 Proxies

Red Hat Satellite Proxy Server is a package-caching mechanism that reduces the bandwidth requirements for Red Hat Satellite and enables custom package deployment. The Satellite Proxy acts as a go-between for client systems and the Satellite Server.

From the client's perspective, there is no difference between a Satellite Proxy and a Satellite . From the Satellite server's perspective, a Satellite Proxy is a special type of Satellite client.

Satellite Proxy servers are a feature of Satellite 5; you cannot use Satellite Proxy servers with Satellite 6. Instead, Satellite 6 introduces the concept of Capsules, which provide much the same functionality.

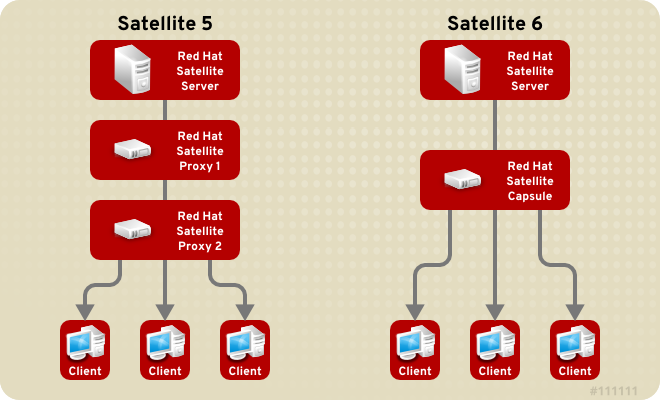
### 2.2.2. Satellite 6 Capsules

Satellite 6 deployments use Capsule servers to provide a similar level of functionality for Satellite 6 that Proxy servers provide for Satellite 5.

The first release of Capsule servers, delivered with Satellite 6.0, can provide the following functionality:

* Mirror content (Pulp Node). Content can be staged on a Pulp Node before it is used by a host.
* Mirror Puppet content (Puppet Master)
* Integrate with DHCP, DNS, TFTP, and IPA.

In Satellite 6.0, Capsule servers do not proxy all data. Consequently, managed systems need connectivity up to Satellite 6, or an HTTP Proxy in place to provide a network bridge if required. Further, you cannot tier Capsule servers like you can with Proxy servers. This is illustrated in the following diagram.



**Figure 2.3. Comparison of Satellite 5 Proxy and Satellite 6 Capsule Servers**