

The software described in this documentation is either no longer supported or is in extended support.
Oracle recommends that you upgrade to a current supported release.

Spacewalk for Oracle® Linux

Concepts Guide for Release 2.4

ORACLE®

E71709-08
November 2019

Oracle Legal Notices

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

About this document

This document describes the basic concepts and features of Spacewalk 2.4 and provides an overview of how to use Spacewalk, including recommended best practices for using Spacewalk in enterprise environments.

Document generated on: 2019-11-25 (revision: 228)

Table of Contents

Preface	v
1 Introducing Spacewalk	1
2 Spacewalk Concepts	3
3 About the Spacewalk Architecture	5
4 Creating and Populating Software Channels	9
5 Configuring Kickstart Provisioning in Spacewalk	13
6 Maintaining Software Channels in Spacewalk	15
7 Managing Development, Test, Acceptance, and Production Life Cycles	17
8 Using System Groups to Standardize Configurations	21
9 Applying Errata to Systems and System Groups	23
10 Further Information	25

Preface

Spacewalk for Oracle® Linux: Concepts Guide for Release 2.4 describes the basic concepts and features of Spacewalk 2.4 and provides an overview of how to use Spacewalk, including recommended best practices for using Spacewalk in enterprise environments.

Audience

This document is written for system administrators who want to use Spacewalk to manage Oracle Linux systems. It is assumed that readers have a general understanding of the Linux operating system.

Related Documents

The documentation for this product is available at:

[Spacewalk for Oracle® Linux Documentation](#)

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1 Introducing Spacewalk

A central task for IT administrators is to provision systems and keep them up to date with the latest patches and operating system updates. If the security policy at your site mandates errata as critical, it is crucially important to conduct regular testing and apply patches to reduce the risk of systems compromise or data exposure. As data centers expand, requiring administrators to manage ever greater numbers of physical servers and virtual machines, automation becomes a necessity for efficient and cost-effective systems management. Spacewalk is open source software that helps to automate Linux systems management, allowing you to control the system software life cycle from initial installation, through maintenance, software configuration, upgrades, and eventual decommissioning.

Provisioning and maintaining large numbers of physical servers and virtual machines across large deployments that span departments and data centers is a challenging task. However, keeping Linux systems up to date with stable software configurations, the latest security errata, and consistent patch levels is critical to user productivity and for day-to-day business operations.

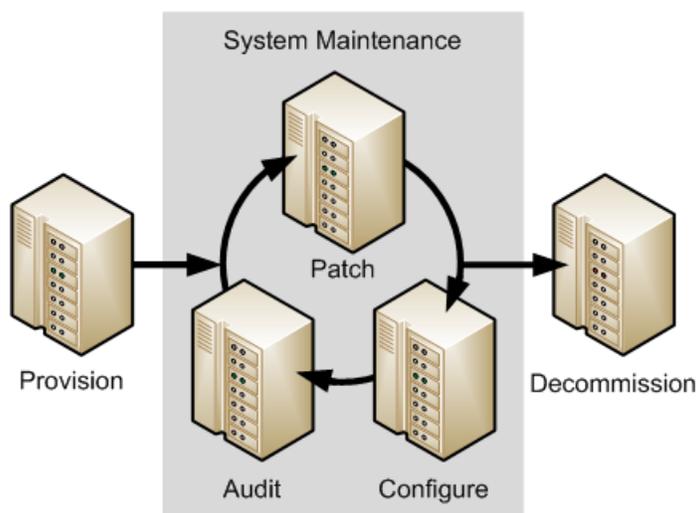
This guide introduces basic concepts and simple commands to help you start using Spacewalk to manage Oracle Linux systems. Beyond the scope of this guide are additional efficiencies that can result from implementing Spacewalk, for example:

- You can use Spacewalk to run automatic OpenSCAP audits against industry-standard security checklists and evaluation profiles.
- You can use the Spacewalk API, which offers powerful and extensive interfaces for high-level scripting, to provide a sophisticated and comprehensive way to automate advanced management tasks.

Spacewalk provides an effective set of tools for managing the Oracle Linux software life cycle in small or large deployments. Spacewalk helps you automate kickstart installation, patching, configuration, and maintenance tasks, and allows you to rapidly deploy proven and consistent software configurations for Oracle Linux systems.

You can use Spacewalk to automate and simplify a number of management tasks, including installing bare metal systems and virtual guests, applying patches and software updates, configuring software, and auditing system security as illustrated in [Figure 1.1, “The Life Cycle of a Typical System”](#).

Figure 1.1 The Life Cycle of a Typical System



You can use either a web interface or command-line utilities to administer Spacewalk. For example, you can use the `spacecmd` command to perform most Spacewalk operations. Spacewalk also features an extensive and powerful XML/RPC-based API that you can use with a high-level scripting language such as Python.

Oracle Linux Premier Support and Oracle Linux Basic Support contracts include support for Spacewalk implementations that are deployed to use Oracle Database, and include a limited-use license for Oracle Database 12c Enterprise Edition for use with Spacewalk. You can use the Spacewalk implementation to automate the management of Oracle Linux 5, Oracle Linux 6, and Oracle Linux 7 systems. When deployed on Oracle Linux, Spacewalk can additionally manage Linux releases such as CentOS, Debian, Fedora, and SLES although Oracle does not support Spacewalk management of these systems.

Spacewalk provides automated management capabilities at no additional cost. Oracle support for this open source project is especially valuable when transitioning to Oracle Linux if you are already familiar with Red Hat Satellite 5.x or SUSE Manager. However, your IT organization might prefer to implement Oracle Enterprise Manager instead because it provides a comprehensive management solution that extends beyond the management of purely Linux-derived systems. The Oracle Enterprise Manager product family supports management of the entire Oracle software stack, including hardware, hypervisors, operating systems and Oracle Database, Oracle middleware, and Oracle software applications. For more information, see the Oracle Enterprise Manager product page at <https://www.oracle.com/enterprise-manager/>.

Chapter 2 Spacewalk Concepts

The following sections describe some basic Spacewalk concepts.

Activation Key

An activation key is a tag that you can associate with a given server configuration, for example, a Database server on an Oracle Linux 7 (x86_64) system. When a client system registers with a Spacewalk server, it specifies an activation key to select the characteristics that are bound to that key, such as which software channels the client should be subscribed to. You can also use activation keys to trigger Spacewalk to install specific packages and deploy a specialized configuration on registration.

Software Channel

Spacewalk client systems subscribe to Spacewalk software channels to obtain software packages and errata. A base or parent channel provides packages for a specific Oracle Linux release and architecture, and this channel can have a number of child channels that provide additional packages. The following example shows a typical base and child channel configuration for the Oracle Linux 7 (x86_64) release that a client might use:

```
oraclelinux7-x86_64-base
|-- oraclelinux7-x86_64-addons
|-- oraclelinux7-x86_64-ksplite
|-- oraclelinux7-x86_64-optional
|-- oraclelinux7-x86_64-patch
|-- oraclelinux7-x86_64-spacewalk24-client
|-- oraclelinux7-x86_64-uek-r4
```

In this example, you would synchronize the content of the base and patch channels with ULN and the content for the Spacewalk client channel with the Oracle Linux yum server. You could synchronize the other channels with either ULN or Oracle Linux yum server. Oracle recommends that you populate the channels manually when you first configure them, and subsequently use automated Spacewalk scheduling or [cron](#) to synchronize the channels on a daily basis.



Note

Oracle also uses the term channel to refer to the software distribution channels that ULN provides. A Spacewalk channel is the subscription mechanism by which clients can obtain software packages, patches, and updates.

You do not necessarily need to associate an upstream repository with a software channel. For example, you might want instead to add custom-built local packages and errata by using the web interface or the [rhnpush](#) command.

Entitlement

An entitlement limits the number of client systems each organization can register to a Spacewalk server. By default, Spacewalk assigns 20,000 entitlements to a Spacewalk server. When you create an organization, you can specify the number of entitlements for that organization out of the 20,000 that are available in total. If necessary, you can use additional Spacewalk servers to provide more entitlements. When you decommission a system by deleting its entry in Spacewalk, any entitlements assigned to the system are automatically released.

System Group

You can organize your client systems into groups, which allows you to perform management operations on multiple systems at a time. A client system can be a member of more than one group. Typically, the member systems of a group have the same Linux release, system architecture, and kickstart profile. You could also group systems by function (for example, web, application, or database servers), by physical location, or by responsible administrator.

Organization

An organization in Spacewalk provides a useful way to tier or segment your Spacewalk implementation. By defining multiple organizations, you can establish management entities that correspond to different corporate divisions or administrative groups. Organizations provide a way to logically delegate system management responsibilities and allocate entitlements. Depending on organizational trust relationships, organizations can also share system and software entitlements.



Important

Oracle strongly recommends that you define at least one Spacewalk organization immediately after you install a Spacewalk server, even if you do not think your deployment requires organizations. It is difficult to retrofit organizations into a Spacewalk implementation at a later time.

Repository

Spacewalk usually provisions packages for software channels by using repositories from an upstream source. For Oracle Linux, you can use the repositories that are provided by either the Oracle Linux yum server (<https://yum.oracle.com>) or the Unbreakable Linux Network (ULN; <https://linux.oracle.com>). Some repository content, such as update-level specific patch and Ksplice packages, is available from ULN but not from the Oracle Linux yum server, while other repository content, such as the Spacewalk Client and Spacewalk Server packages, is available from the Oracle Linux yum server but not from ULN. Spacewalk provides a ULN plugin that allows it to synchronize Spacewalk repositories with ULN without needing to register the Spacewalk server directly with ULN. Unless you are an experienced Spacewalk administrator, Oracle recommends that you associate only one repository with each channel to obtain upstream packages. Otherwise, the channel attempts to pull packages from multiple sources.



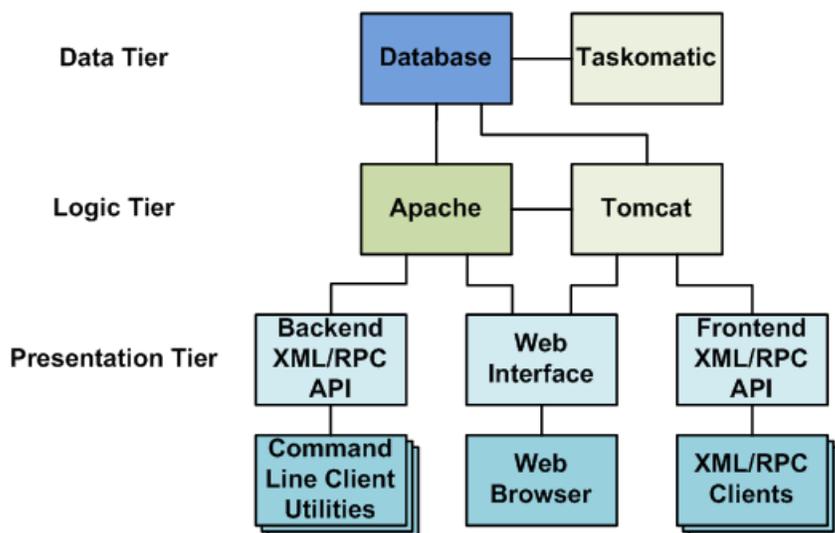
Note

Ksplice is available under the Oracle Linux Premier Support contract.

Chapter 3 About the Spacewalk Architecture

An overview of the internal three-tier Spacewalk server architecture is illustrated in [Figure 3.1, “Overview of the Internal Three-tier Spacewalk Server Architecture”](#).

Figure 3.1 Overview of the Internal Three-tier Spacewalk Server Architecture



The data tier contains the database and Taskomatic. Spacewalk uses the Taskomatic daemon to perform asynchronous scheduled tasks such as resynchronizing software channels, applying software and configuration updates to clients, and notifying you when new errata become available.

The logic tier contains the Apache and Tomcat web servers, which process data for use by the presentation tier.

The presentation tier contains the backend and frontend XML/RPC APIs, which provide programming interfaces for the command line client utilities and other XML/RPC clients, and the web interface, which can be accessed using a web browser.

Spacewalk uses a distributed client-server architecture in which registered client systems subscribe to the software channels that a Spacewalk server hosts. You can distribute server functionality across as many systems as you require to meet your organization's needs, which is especially useful for organizations that are distributed across several geographical regions.

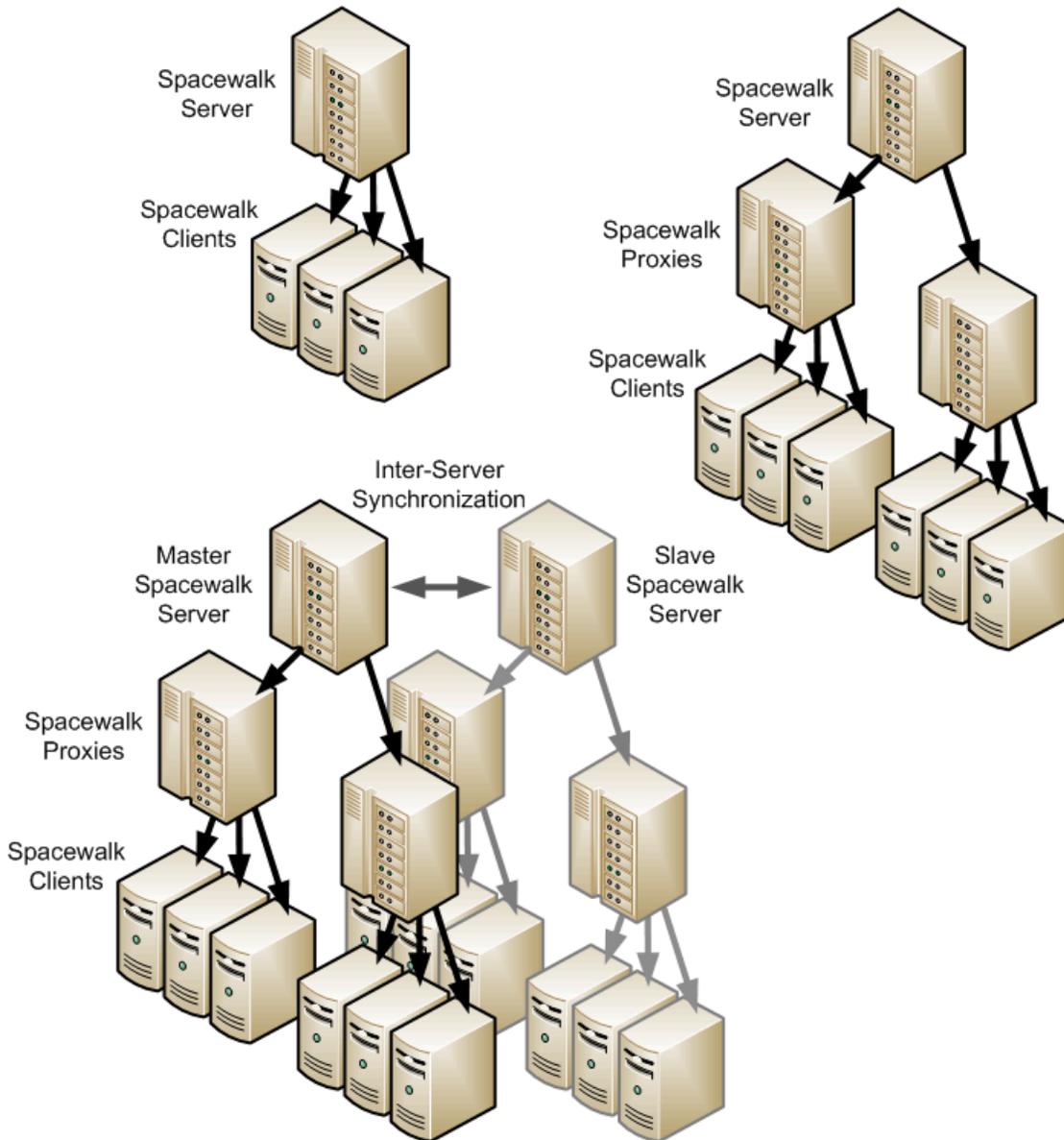
A simple deployment might have a single master server to serve a pool of client systems together with perhaps a proxy server to offload processing from the master so that you can support more client systems. In larger deployments, you can configure multiple Spacewalk master and proxy servers to speed up provisioning and software download speeds. In a distributed configuration, clients are likely to have higher network bandwidth access to servers that are in close proximity.

As the Spacewalk architecture is flexible and scalable, many deployment configurations are feasible. [Figure 3.2, “Example Deployment Configurations for Spacewalk”](#) illustrates some possible scenarios:

- A simple deployment with a single Spacewalk server that has several clients.
- A more complex deployment with one Spacewalk server and two Spacewalk proxies, where each proxy has several clients.

- A deployment with two Spacewalk servers, each at different sites. The Spacewalk server at one site acts the master server, the Spacewalk server at the other site acts as a slave server, and Inter-Server Synchronization (ISS) is used to manage channel content, channel permissions, and organizational trust settings between the two servers.

Figure 3.2 Example Deployment Configurations for Spacewalk



Oracle consultants can help your organization to design an optimal Spacewalk solution, especially if you need to manage a large, geographically dispersed environment of client systems and servers.

You register client systems, whether physical servers or virtual guests, with a Spacewalk server to subscribe them to software channels. The clients can then obtain packages from the Spacewalk server. You can also use Spacewalk with kickstart to automate Oracle Linux installation using software packages from a network installation server. If you use Spacewalk with kickstart to provision new client systems, you can configure the Spacewalk server to automatically register these client systems. A new client system is

The software described in this documentation is either no longer supported or is in extended support.
Oracle recommends that you upgrade to a current supported release.

automatically registered as a Spacewalk client if you have associated an activation key with its kickstart profile and you configure kickstart to install the Spacewalk Client software on the system.

You can register Oracle Linux 7 Update 1 or later and Oracle Linux 6 Update 7 or later systems without first installing the Spacewalk client software, although Oracle recommends installing the client software to obtain full Spacewalk client management capabilities.

You can use Spacewalk to administer a registered Kickstarted system right away, which simplifies subsequent patching, configuration management, and security auditing. You can also register previously installed legacy systems with a Spacewalk server if you want to bring these systems under Spacewalk control.

Chapter 4 Creating and Populating Software Channels

Before you can use Spacewalk and kickstart to provision client systems, you must first define channels and populate them with software packages. To populate the channels, you can use the web interface, `spacewalk-repo-sync`, or `spacecmd` to pull packages from a repository, or use `rhnpush` to push packages to the Spacewalk server. For example, you might use `rhnpush` to extract packages from an ISO image or to upload internally developed packages.

To create a channel in the web interface, navigate to **Channels**, then **Manage Software Channels**, and click **+ create new channel**. Specify a channel name, channel label, parent channel (select **None** if you are creating a base channel), and the GPG key information for the channel.

For example, if you plan to provision systems with Oracle Linux Release 7 Update 2, you might create a base channel with the label `ol7u2_x86_64_base` and child channels for patches, Spacewalk 2.4 client software, and the Unbreakable Enterprise Kernel Release 4 from Oracle, as follows:

```
ol7u2_x86_64_base
|-- ol7u2-x86_64_patch
|-- ol7u2-x86_64_spacewalk24-client
|-- ol7u2-x86_64_uek-r4
```

These four channels are typical for configuring clients that you provision using kickstart. From Oracle Linux Release 6 Update 7 and Oracle Linux 7 Release 1 onward, you do not need to install the Spacewalk client packages to register clients. However, Oracle recommends installing the Spacewalk client channel packages to take full advantage of Spacewalk's management capabilities, including provisioning and auditing. You also need to enable the Spacewalk Client channel if you want kickstart to register a client automatically. Kickstart does not recognize the built-in client for this purpose.

There are a number of ways to populate the created software channels with packages. An easy way to populate a software channel initially so that it can be used with kickstart is to mount an ISO image on the server and use `rhnpush`. After mounting an ISO image on the server, you can push packages from the ISO distribution, for example:

```
# mount -o loop /var/ISOs/V100082-01.iso /var/distro-trees/ol7u2-x86_64-server
# rhnpush -v --channel=ol7u2_x86_64_base --server=http://swksvr.mydom.com \
--dir=/var/distro-trees/ol7u2-x86_64-server/Packages
```

Alternatively, you can synchronize channels from the ULN or the Oracle Linux yum server repositories by using the web interface, `spacewalk-repo-sync`, or `spacecmd`.



Note

The system that is used to run `rhnpush`, must be registered with Spacewalk server before this command is used, even if the command is being run on the same system that is hosting the Spacewalk server. The `rhnpush` command uses TLS/SSL to communicate with the Spacewalk server. If the system is not registered, `rhnpush` fails with a certificate error. Client registration updates the configuration to use the correct Spacewalk SSL certificate. For more information on client registration, please see *Registering Client Systems* in [Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.4](#).

About Channel Differences Between ULN and Oracle Linux Yum Server

ULN provides base and patch channels for each update of an Oracle Linux release. The Oracle Linux yum server does not provide separate patch channels. Instead, it provides a [public_olN_latest](#) repository that includes all packages for an entire release. Consider the following example of populating four channels:

```
ol7u2_x86_64_base
|-- ol7u2-x86_64_patch
|-- ol7u2-x86_64_spacewalk24-client
|-- ol7u2-x86_64_uek-r4
```

While you can populate the base and UEK R4 channels from an Oracle Linux 7 Update 2 full ISO image, only ULN provides a patch channel for this release, and only the Oracle Linux yum server provides the Spacewalk 2.4 client channel. If you populate channels from ULN, Oracle recommends that you configure separate base and patch channels for each update (for example, [ol6u7_x86_64_base](#) and [ol6u7_x86_64_patch](#) for Oracle Linux 6 Update 7; [ol7u2_x86_64_base](#) and [ol7u2_x86_64_patch](#) for Oracle Linux 7 Update 2, and so on).

About the Speed of Channel Synchronization

Populating base channels for an Oracle Linux release from an ISO image is generally faster than pulling the base packages from an upstream repository. It can take several days to synchronize an Oracle Linux latest channel from the Oracle Linux yum server repository.

After you have created channels and initially populated them from an ISO image, you can synchronize them with the Oracle Linux yum server and ULN to bring them up to date. The Oracle Linux yum server is mirrored on a number of servers globally, so it can be faster to synchronize the base channels with the Oracle Linux yum server and subsequently synchronize any patch channels with ULN.

If you specify the `--latest` option to [spacewalk-repo-sync](#) or select the equivalent optional check box in the web interface, the server synchronizes only the latest packages that are available at the time of synchronization. It does not remove older packages from the channel. If the synchronization interval is large, you might miss a particular version of a package. This can have implications for errata handling, where errata are associated with specific package versions. If errata consistency is important to you, Oracle recommends that you do not select these options. Using `--latest` with a Ksplice channel is an exception to this rule because its packages are always cumulative.

About Channel Storage Space

Spacewalk requires a significant amount of disk space because it maintains all available versions of all packages in each software channel. Typically, the Oracle Linux binary repositories require approximately 60 GB for each combination of Oracle Linux release and architecture. An extra 40 GB is required for source packages and 80 GB for Ksplice updates for each combination of Oracle Linux release and architecture. However, Spacewalk conserves space where possible. If the same package is available in more than one channel, Spacewalk stores only a single instance of the package and makes the package metadata for each channel point to the same package.

For example, the [addons](#) channel for Oracle Linux Release 7 Update 2 is the same as for the initial Oracle Linux Release 7 and Oracle Linux Release 7 Update 1. A typical channel configuration has a separate base channel for each release with an associated [addons](#) child channel: [ol7u0_x86_64_addons](#),

The software described in this documentation is either no longer supported or is in extended support.
Oracle recommends that you upgrade to a current supported release.

About Channel Storage Space

`o17u1_x86_64_addons`, and `o17u2_x86_64_addons`. In this case, the Spacewalk database stores a common set of packages for all three channels.

Chapter 5 Configuring Kickstart Provisioning in Spacewalk

You can use Spacewalk in conjunction with kickstart to automate the installation of Oracle Linux systems. When used in conjunction with Spacewalk for provisioning, Kickstart obtains software packages for target systems from channels on a Spacewalk server instead of a networked, kickstart installation server.

Configuring Spacewalk and kickstart to install and provision systems requires several steps, which can be summarized as follows:

1. Set up distribution trees on the Spacewalk server for each Oracle Linux release and system architecture combination that you want to provision.

Each distribution tree must contain an installation kernel and RAM-disk image that a target system can use to boot. For example, you could mount a full distribution ISO image for Oracle Linux 7 Update 2 at `/var/distro-trees/ol7u2-x86_64-server` in the same way as you would do to populate a base channel initially. Kickstart can access the installation kernel and RAM-disk image in the `images/pxeboot` subdirectory of the mount point in the distribution tree.

2. Use the web interface or the `spacecmd distribution_create` command to define a kickstart distribution in Spacewalk.

Specify a name for the distribution, the path of the kickstart tree, and the base channel to associate with the distribution.

3. Use the web interface or the `spacecmd kickstart_create` command to configure a kickstart profile.

A kickstart profile generates an associated kickstart file, which contains all of the details that Anaconda requires to install a system: base channel, kickstart tree, software preferences, root password, locale, partitions, activation keys, preinstallation and postinstallation scripts, and so on. If you have an existing kickstart file, you can upload it into Spacewalk and use the web interface to edit and manage it.

To automatically register a Spacewalk client system, associate an activation key with its kickstart profile and configure kickstart to install the Spacewalk Client software on the system.

4. (Optional) Specify the installation of the OSA daemon (`osad`) in the kickstart profile.

If you install `osad`, the Spacewalk server can apply updates and actions to a client system immediately rather than once every four hours (which is the default behavior of the default `rhnsd` daemon).



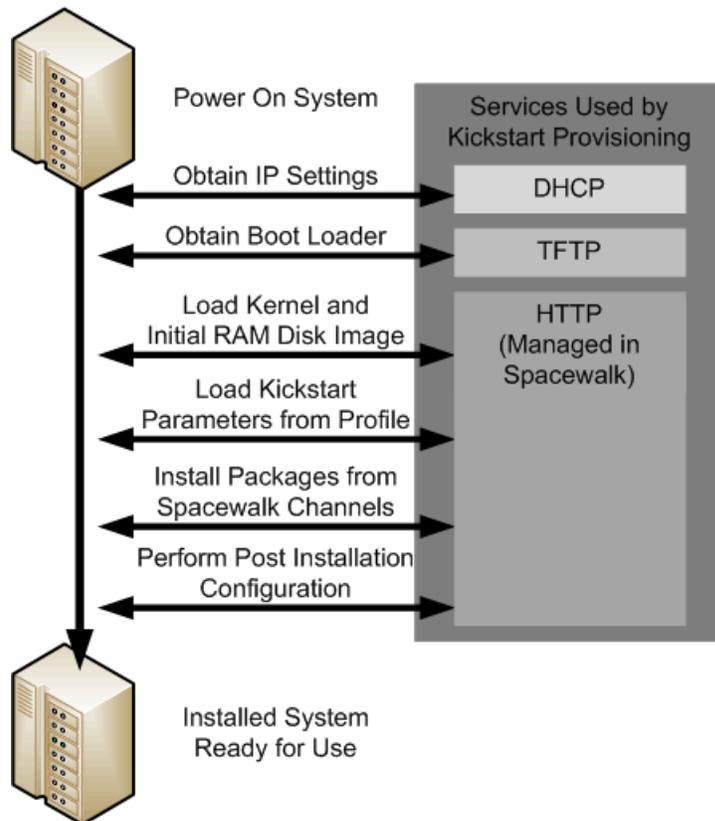
Note

`osad` can sometimes experience connection issues that can usually be avoided by configuring a `cron` job to stop it, delete its database, and restart it once per day.

After creating a kickstart profile, you can use the associated kickstart file to install client systems. To ensure consistency across installed client systems, you might find it advantageous to have a small core kickstart file that you use for most installations. You can then use Spacewalk to subscribe updates for a client and customized the software channels for which it is registered.

The web interface provides a convenient way to kickstart a client system that boots using DHCP and PXE boot as illustrated in [Figure 5.1, “The Kickstart Provisioning Process for a Client System that Uses DHCP and PXE Boot”](#).

Figure 5.1 The Kickstart Provisioning Process for a Client System that Uses DHCP and PXE Boot



After powering on the client system, it obtains its IP settings from DHCP, including the address of the TFTP server. The client next downloads the boot loader ([pxelinux](#)) from the TFTP server. The boot loader then downloads the installation kernel and initial RAM-disk images from the Spacewalk server. The installation kernel runs Anaconda, which downloads the kickstart parameters, including the kickstart file, from the Spacewalk server and uses this information to install packages from Spacewalk software channels, and perform any required system configuration.

If you are familiar with Cobbler, you might prefer to use the [cobbler](#) command to configure and manage kickstart profiles.

Chapter 6 Maintaining Software Channels in Spacewalk

After you have used Spacewalk and kickstart to provision client systems, you must maintain these systems. For example, whenever new security errata are released, you can apply updates to protect systems, applications, and data. Spacewalk is particularly useful in this respect. When client systems register with a Spacewalk server and subscribe to software channels, Spacewalk makes the latest software packages available to the clients whenever the channels are updated.

There are countless ways to configure Spacewalk channels and maintain client systems, and almost every organization and system administrator have their own approach. When configuring a Spacewalk environment, it is helpful to plan how updates should occur. Should Spacewalk update systems automatically with the latest errata as soon as patches are released or is explicit manual control required over which errata are applied to which systems?

Spacewalk is extremely flexible and you can configure channels to meet the requirements of a variety of deployment scenarios. Oracle consultants can help your organization analyze its operational requirements and design an optimal Spacewalk configuration that meets your system security and maintenance policies.

The following are a few suggestions for configuring channels:

- Create base and patch channels for each release from ULN.

A common strategy for channel creation is to create a base and patch channel for each Oracle Linux release and update level and then regularly synchronizing these channels with ULN. Periodically synchronize the patch channel to pull in the most recent errata. Spacewalk offers effective mechanisms, using the web interface and the command line, to view and manage errata, including cloning errata for channels, if required.

- Create an Oracle Linux Ksplice channel.

You can configure a Spacewalk server to mirror the Oracle Linux Ksplice channels on ULN, downloading the latest Ksplice update packages to a software channel. Using the Ksplice Offline Client software, Spacewalk clients can install kernel updates from the Spacewalk server without needing to reboot and incurring downtime.

- Maintain the *latest* channels for applying errata to systems locked to earlier updates.

Some organizations have application requirements or policies that require some systems to remain at a particular update level of an Oracle Linux release. For example, perhaps certain production machines must remain locked to Oracle Linux 6 Update 6. Oracle releases new errata and patches for the latest release, such as Oracle Linux 6 Update 7. If upgrading to the latest update is not feasible for these systems, Oracle strongly recommends that you apply the latest security errata to avoid compromising the security of these systems.

One strategy is to maintain *latest* channels, containing the latest errata, separately, usually without any client systems subscribed to these channels. You can then copy the errata and dependent packages from these latest channel to other channels such as release and update-specific patch channels if you need to make the latest fixes available.

Chapter 7 Managing Development, Test, Acceptance, and Production Life Cycles

Many Oracle Linux environments implement an application and system software life cycle known as *DTAP*, which stands for Development, Test, Acceptance, and Production. Spacewalk supports software channel cloning to simplify the process of promoting software configurations from development to test, test to acceptance, and acceptance to production.

Cloning an existing channel creates a new channel that reflects the state of the original channel's packages and errata at a particular point in time. You can modify a channel before cloning it, perhaps applying the errata available at a certain date or a set of errata that you have specifically selected. The cloned channel then provides a stable base for development and testing before release to production.

Spacewalk keeps track of all registered client systems and the software channels to which they are subscribed. Updates are available to client systems whenever their subscribed channels are updated. Alternatively, you can subscribe clients to a different channel, such as a cloned channel that contains new errata that have been tested. Cloning gives you more precise control over software configurations and establishes a methodical workflow to reduce the risk of production problems in mission-critical application environments.

You can use either the Spacewalk web interface or `spacecmd` to clone channels. You can specify whether to clone a channel in its current state including all errata, in its original state with only a selected set of errata, or in its original state with no errata.

The following example shows how you could use the `softwarechannel_clone` subcommand of `spacecmd` to clone a channel:

```
spacecmd {SSM:0}> softwarechannel_clone -s ol7u2-x86_64 -x "s/$/-clone/" -o
```

To clone a base channel and all of its children:

```
spacecmd {SSM:0}> softwarechannel_clonetree -s ol7u2-x86_64 -p "clone-"
```

Cloning is usually very fast as Spacewalk only needs to duplicate metadata entries in its database. It does not duplicate the packages themselves. Keeping channels small in size also helps to speed up channel cloning.

Cloning the base and patch channels for an Oracle Linux release is significantly faster than cloning the latest channel of an Oracle Linux release because the latest channel contains every version of every package from the initial release date to the current date. A base channel for an Oracle Linux release contains only what was shipped on the release's ISO image and a patch channel contains the updates made available since the ISO image was created. For this reason, base and patch channels are preferred sources for channel cloning.

When establishing a DTAP workflow, you might want to clone channels based on a particular date. You can use the `spacewalk-clone-by-date` utility to ensure that the new channel contains packages and errata up to and including a specified date.

For example, the following command clones a patch channel up to and including November 12, 2015:

```
# spacewalk-clone-by-date --username=swadmin --password=password \  
--channels=ol7u1-x86_64-patch ol7u1-x86_64-patch-151112 --to_date=2015-11-12
```

It is also possible to blacklist or remove certain packages, and to choose types of errata to include or exclude.

You can also make `spacewalk-clone-by-date` perform channel cloning based on a configuration file. You can create a configuration file by using the `--sample_config` option and then edit the file according to your requirements.

For example, the following configuration file `clone-151112.conf` clones the `addons` and `patch` channels up to and including November 12, 2015. The `removelist` section additionally specifies that any packages involving `sendmail` should not be included in the cloned channels.

```
{
  "username": "admin ",
  "passwd": "spacepw",
  "assumeyes": true,
  "skip_depsolve": false,
  "security_only": false,
  "blacklist": {
  },
  "removelist": {
    "ALL": ["sendmail"]
  },
  "channels": [
    {
      "ol7u1-x86_64-addons": "ol7u1-x86_64-addons-151112",
      "ol7u1-x86_64-patch": "ol7u1-x86_64-patch-151112",
    }
  ]
}
```

The command `spacewalk-clone-by-date -d 2015-11-12 -c clone-151112.conf` would create the cloned channels based on this configuration file.

In a DTAP environment, you might want to clone Oracle Linux channels on a periodic schedule, such as monthly or weekly. A typical DTAP workflow involves the Spacewalk server pulling packages and the latest errata from the ULN repository for development channels. Assuming integration testing requires less than five days in this scenario, you could configure Spacewalk to create test channel clones of the updated development channels automatically each week. After the test channels have passed all quality assurance tests and been accepted, you could then manually clone them into production channels.

While it is possible to use `spacewalk-clone-by-date` to set up a DTAP workflow, you might find it more convenient to use the `spacewalk-manage-channel-lifecycle` command to manage the life cycle of a software channel from development, through testing to production, as shown in the following examples:

- Create a development channel `dev-ol7-x86_64-appsvr` based on the latest available packages in `ol7-x86_64-appsvr`.

```
# spacewalk-manage-channel-lifecycle -c ol7-x86_64-appsvr --init
```

- Promote the packages from the development channel to the test channel `test-ol7-x86_64-appsvr`.

```
# spacewalk-manage-channel-lifecycle -c dev-ol7-x86_64-appsvr --promote
```

- Promote the packages from the test channel to the production channel `prod-ol7-x86_64-appsvr`.

```
# spacewalk-manage-channel-lifecycle -c test-ol7-x86_64-appsvr --promote
```

You can save the state of a channel by creating an archive channel `archive-date-channel`.

The software described in this documentation is either no longer supported or is in extended support.
Oracle recommends that you upgrade to a current supported release.

```
# spacewalk-manage-channel-lifecycle -c test-ol7-x86_64-appsvr --archive
```

If you need to restore the state of a channel, use the `--rollback` option and specify the archived version of the channel that you want to restore, for example:

```
# spacewalk-manage-channel-lifecycle -c archive-20110520-test-ol7-x86_64-appsvr --rollback
```

Use the `-l` option to list the channels:

```
# spacewalk-manage-channel-lifecycle -l
Channel tree:

1. archive-20160203-ol7-x86_64-appsvr
   \__ archive-20160203-prod-ol7-x86_64-appcmd
   \__ archive-20160203-prod-ol7-x86_64-applib

2. dev-ol7-x86_64-appsvr
   \__ dev-ol7-x86_64-appcmd
   \__ dev-ol7-x86_64-applib

3. ol7-x86_64-appsvr
   \__ ol7-x86_64-appcmd
   \__ ol6-x86_64-applib

4. prod-ol7-x86_64-appsvr
   \__ prod-ol7-x86_64-appcmd
   \__ prod-ol7-x86_64-applib

5. test-ol7-x86_64-appsvr
   \__ test-ol7-x86_64-appcmd
   \__ test-ol7-x86_64-applib
```

After promoting a test channel to a production channel, you can put the accepted software configuration into production by subscribing your production systems to that channel on the **Software Channel Subscriptions** page of the web interface or by using the `spacecmd system_setbasechannel`, for example:

```
spacecmd {SSM:0}> system_setbasechannel svr5.mydom.com prod-ol7-x86_64-appsvr
```

To change the base channel for multiple production systems, you can map an existing base channel to a new base channel, for example:

```
spacecmd {SSM:0}> system_setbasechannel channel:ol7-x86_64-QAr1-base \
ol7-x86_64-QAr2-base
```


Chapter 8 Using System Groups to Standardize Configurations

Specifying groups of systems provides a way to make changes to many systems at the same time, for example to apply errata, update packages, or change channel subscriptions. System groups allow you to standardize software configurations so that member systems have consistent release and patch levels. In large-scale deployments, using system groups helps to automate system management tasks.

You can create a system group on the **System Groups** page of the web interface or by using the `group_create` and `add_systems` subcommands of `spacecmd`.

For example, create the group `grp-prod-db-mke` and add all systems that currently subscribe to the channel `o17u1-x86_64-20150807` to that group:

```
spacecmd {SSM:0}> group_create grp-pdb "Production DB Server Group"
spacecmd {SSM:0}> add_systems grp-pdb channel:o17u1-x86_64-20150807
```

To change the base channel for all servers in a system group, you can specify the group name to `spacecmd`, as shown in the following example:

```
spacecmd {SSM:0}> system_setbasechannel group:grp-pdb o17-x86_64-QAr2-base
```

You can use a command such as this to change the channel subscriptions for many systems at the same time, for example, to promote all systems that currently subscribe to test channels to subscribe to production channels instead. You can also perform Spacewalk management operations on a single system group or on the union or intersection of two or more system groups.

Chapter 9 Applying Errata to Systems and System Groups

Spacewalk allows you to monitor available errata and gives you precise control when applying errata to systems. You can see the errata that are available for a selected system or system group on the **Errata** page of the web interface. Applying filters allows you to display errata according to whether they are noncritical, bug fix advisory, product enhancement advisory, or security advisory.

The `system_listerrata` subcommand of `spacecmd` provides similar functionality. For example, list the errata that are available for a specified system:

```
spacecmd {SSM:0}> system_listerrata svr1.mydom.com
```

The `errata_details` subcommand displays additional information about a given erratum:

```
spacecmd {SSM:0}> errata_details ELSA-2015-1115
```

You can specify a system group to the `system_applyerrata` subcommand to apply errata to multiple systems at the same time:

```
spacecmd {SSM:0}> system_applyerrata group:grp-prod-db-mke ELSA-2015-1115
```

If you have configured the OSA daemon (`osad`) on a client, the erratum is applied immediately. If a client system is not configured with `Ksplice`, you must reboot it for kernel changes to take effect. A client system that uses `Ksplice` can take advantage of the kernel changes that result from a channel change without needing to reboot.

Chapter 10 Further Information

To learn more about Oracle Linux, Ksplice, Spacewalk, and ULN, see the [Oracle® Linux 6 Documentation](#), [Oracle® Linux 7 Documentation](#) and [Spacewalk for Oracle® Linux Documentation](#).

More information about using the Spacewalk web interface and `spacecmd` is available in [Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.4](#).

For more information about Spacewalk, go to the Spacewalk open source project page at <https://fedorahosted.org/spacewalk/>.

