Spacewalk for Oracle® Linux

Concepts Guide for Release 2.6

ORACLE®

Oracle Legal Notices

About this document

This document describes the basic concepts and features of Spacewalk 2.6 and provides an overview of how to use Spacewalk, including recommended best practices for using Spacewalk in enterprise environments.

Document generated on: 2020-08-21 (revision: 556)

Table of Contents

Preface

Spacewalk for Oracle® Linux: Concepts Guide for Release 2.6 describes the basic concepts and features of Spacewalk 2.6 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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	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, administrators are required to manage even greater numbers of physical servers and virtual machines. As a result, automation becomes a necessity for efficient and cost-effective systems management. Spacewalk is open source software that helps to automate Oracle Linux systems management, enabling 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 Oracle Linux systems up to date with stable software configurations, the latest security errata, and consistent patch levels is critical to user productivity, as well as for managing day-to-day business operations.

This guide introduces you to the basic concepts for 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 do the following:

- Run automatic OpenSCAP audits against industry-standard security checklists and evaluation profiles.

- 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 a kickstart installation, system configuration, and maintenance tasks, thereby enabling you to rapidly deploy proven and consistent software configurations for Oracle Linux systems.

Other ways that you can use Spacewalk to simplify management tasks include installing bare metal systems and virtual guests, applying patches and software updates, configuring software, and auditing system security. The following figure illustrates the life cycle of a typical system.

Figure 1.1 Life Cycle of a System
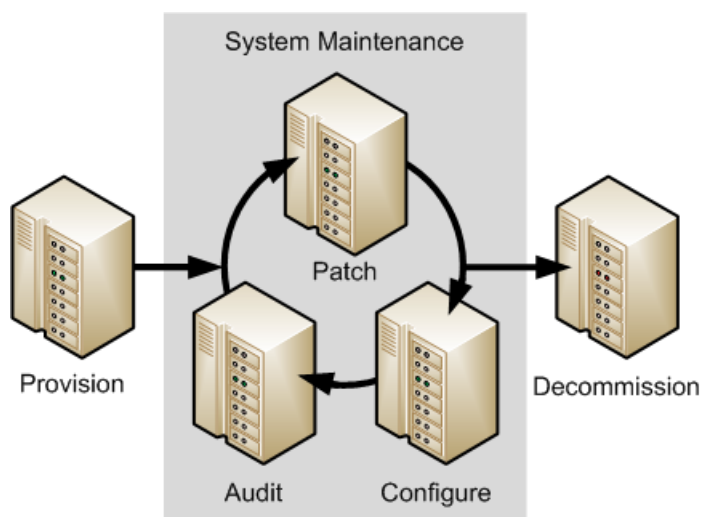
You can administer Spacewalk and perform most Spacewalk operations by using either a web interface or the `spacecmd` command. Spacewalk also features an extensive and powerful XML/RPC-based API that you can use with a high-level scripting language such as Python.

Oracle Linux Premier Support and Oracle Linux Basic Support contracts provide support for Spacewalk and include a limited-use license for Oracle Database 12c Enterprise Edition for use with Spacewalk. You can use Spacewalk 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 other Linux releases, such as CentOS, Debian, Fedora, and SLES. However, note that Oracle does not support Spacewalk for management of these systems.

Spacewalk provides automated management capabilities at no additional cost. Oracle support for Spacewalk is especially valuable when transitioning to Oracle Linux, for those who are already familiar with Red Hat Satellite 5 releases 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 are some basic Spacewalk concepts. For more information about these concepts, as well as step-by-step instructions and examples, see the *Spacewalk for Oracle® Linux Documentation*.

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 to which software channels the client should be subscribed. You can also use activation keys to trigger Spacewalk to install specific packages and deploy a specialized configuration upon registration.

Software channel

Are subscribed to by Spacewalk client systems subscribe to obtain software packages and errata. A base or parent channel provides packages for a specific Oracle Linux release and architecture. This channel can have a number of child channels that provide additional packages.

> **Note**
>
> Oracle also uses the term *channel* to refer to the software distribution channels that the Unbreakable Linux Network (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 by using the `rhnpush` command.

Entitlement

Is used to enable additional functionality that is supported by Spacewalk. Note that starting with Spacewalk 2.6, most entitlements were removed. The only entitlement that is currently supported is the Virtualization entitlement, which installs additional packages on the target host so that Spacewalk is able to manage virtual guests on that host.

System group

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

Organization

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.

> **Note**
>
> Starting with Spacewalk 2.6, you must define at least one Spacewalk organization immediately after you install a Spacewalk server.

Repository

Is used to provision packages for software channels. 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 at https://yum.oracle.com or ULN at 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. Spacewalk provides a ULN plugin that enables 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 Offline packages are available from ULN as part of an Oracle Linux Premier Support contract. For more information, see *Oracle® Linux: Ksplice User's Guide*.

Chapter 3 Overview of the Spacewalk Architecture

The following figure shows an overview of the internal, three-tier Spacewalk Server architecture.

Figure 3.1 Overview of the Internal Three-Tier Spacewalk Server Architecture
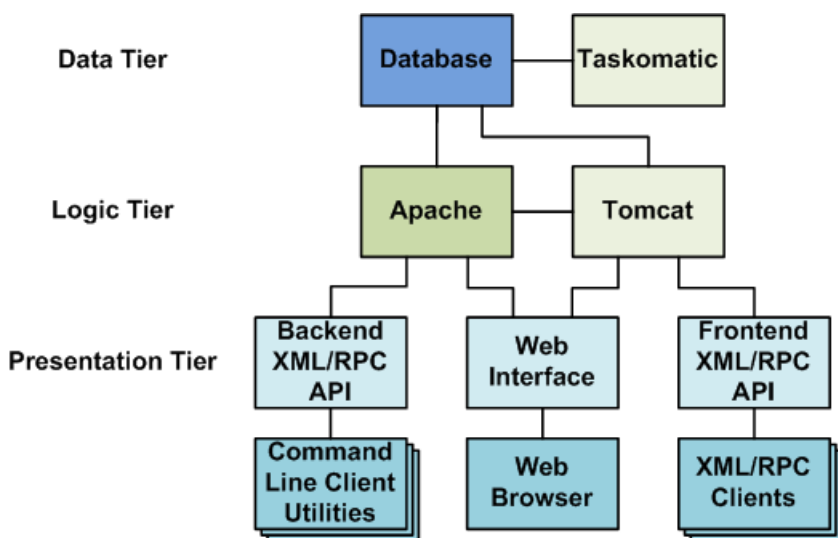
The server architecture includes the following three tiers:

- 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.

- Logic tier – Contains the Apache and Tomcat web servers, which process data for use by the presentation tier.

- Presentation tier – Contains the back-end and front-end 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 by 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. This capability is especially useful for organizations that are distributed across several geographical regions.

For example, a simple deployment might have a single master or primary server to serve a pool of client systems, along with a proxy server to offload processing from the primary server so that you can support more client systems. In larger deployments, you can configure multiple Spacewalk primary servers 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 within close proximity.

Because the Spacewalk architecture is flexible and scalable, many different deployment configurations are feasible. The following scenarios, which are also illustrated in Figure 3.2, "Example Deployment Configurations for Spacewalk", are possible:

- Simple deployment with a single Spacewalk server that has several clients.

- More complex deployment with one Spacewalk server and two Spacewalk proxies, where each proxy has several clients.

- Deployment with two Spacewalk servers, each at different sites. The Spacewalk server at one site acts as the primary server, while the Spacewalk server at the other site acts as a worker (slave) server. 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
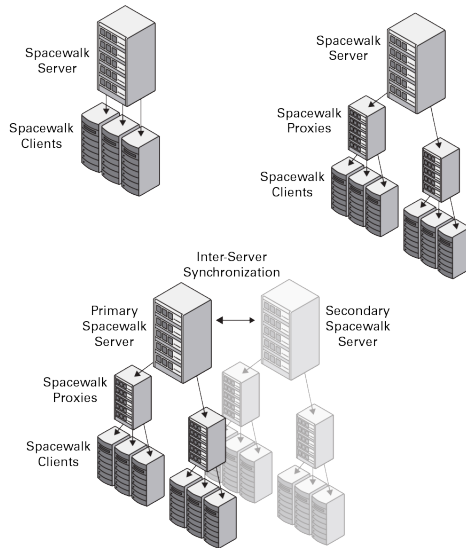
Oracle consultants can help your organization design an optimal Spacewalk solution, especially if you need to manage a large, geographically dispersed environment of client systems and servers.

You register client systems, whether physical servers or virtual guests, with a *Spacewalk Server Proxy* 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 an Oracle Linux installation by 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 automatically registered as a Spacewalk client if you have associated an activation key with its kickstart profile and configured kickstart to install the Spacewalk Client software on the system.

You can register Oracle Linux 7 Update 1 and later and Oracle Linux 6 Update 9 or later systems without first installing the Spacewalk client software. However, 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 to bring these systems under Spacewalk control.

Chapter 4 About Software Channels

Before you can use Spacewalk to provision client systems, you must first define software channels and populate them with software packages. You can configure Spacewalk to automatically populate packages from external repositories. Or, you can use command-line tools to manually push packages into a software channel.

From Oracle Linux 7 Update 1 onward and Oracle Linux 6 Update 9 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, as 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 the `rhnpush` command. After mounting an ISO image on the server, you can push packages from the ISO distribution.

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

> **Note**
>
> Before running the `rhnpush` command, the system that you are using must be registered with the Spacewalk server, even if the command is being run on the same system that is hosting the Spacewalk server. The `rhnpush` command uses the Transport Layer Security/Secure Socket Layer (TLS/SSL) protocols to communicate with the Spacewalk server. If the system is not registered, the `rhnpush` command fails with a certificate error. Client registration updates the configuration to use the correct Spacewalk SSL certificate. For more information about client registration, see *Registering Client Systems* in *Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.6*.

Differences Between ULN and Oracle Linux Yum Server Channels

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 `olN_latest` repository that includes all of the packages for an entire release. See *Working With Software Channels* in *Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.6* for further information.

Optimizing the Speed of Channel Synchronization

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

After you have created and initially populated channels 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 with the `spacewalk-repo-sync` command 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. The process 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. Note that using `--latest` with a Ksplice channel is an exception to this rule because its packages are always cumulative.

Chapter 5 About Provisioning Client Systems in Spacewalk

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

Configuring Spacewalk and kickstart to install and provision systems involves several steps, including creating a kickstart profile and using a kickstart file to install client systems. For complete instructions, see *Provisioning Client Systems* in *Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.6*.

The web interface provides a convenient way to kickstart a client system that boots by using DHCP and PXE boot, as illustrated in the following figure.

Figure 5.1 Kickstart Provisioning Process for a Client System that Uses DHCP and PXE Boot
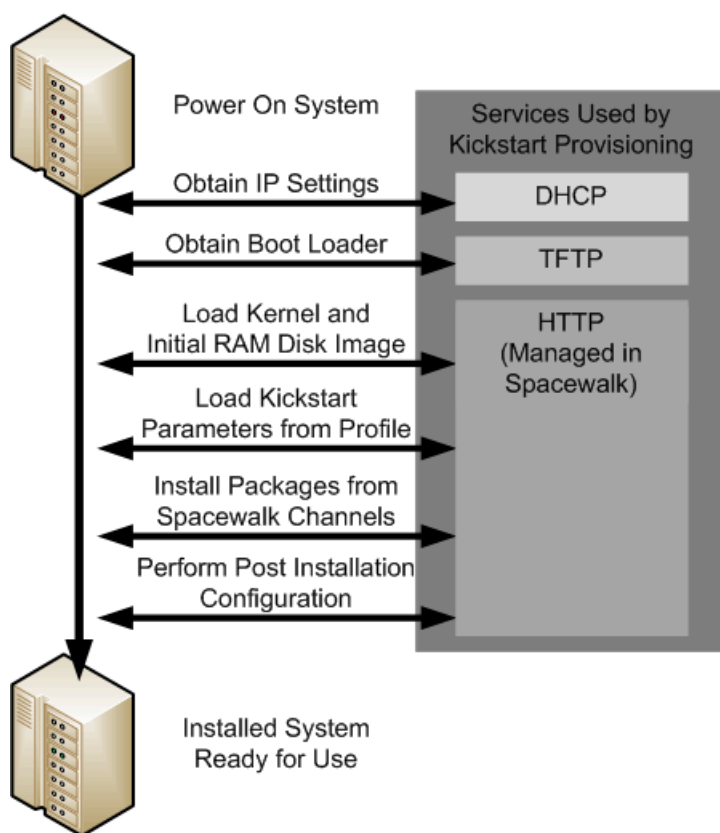
The figure illustrates the following process:

1. After powering on the client system, it obtains its IP settings from DHCP, including the address of the TFTP server.

2. Next, the client 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.

3. 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 About 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 methods that you can use to configure 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 that contain 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 Overview of 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 the `spacecmd` command to clone channels in the following ways:

- Clone a channel in its current state, including all errata.

- Clone a channel in its original state, with only a selected set of errata.

- Clone a channel in its original state, with no errata.

Note the following additional points about cloning:

- 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.

> **Note**
>
> 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` command to ensure that the new channel contains packages and errata up to and including a specified date.

- You can blocklist or remove certain packages and choose types of errata to include or exclude when cloning.

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

- 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 have been accepted, you can then manually clone them as production channels.

  While it is not possible to use the `spacewalk-clone-by-date` command 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, in the following ways:

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

  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:

- 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` command.

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

For more information, including instructions and examples, see *Cloning Software Channels* in *Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.6*.

Chapter 8 About Configuring 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, and change channel subscriptions. Using system groups enables 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 the `spacecmd` command.

You can also make changes for all servers in a system group by specifying the *group name*, for example, to change the base channel for all of the servers in a system group. Or, you could change the channel subscriptions for many systems at the same time to promote all of the systems that are currently subscribed 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. See *Configuring System Groups to Manage Client Systems* in *Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.6*.

Chapter 9 About Applying Errata to Systems and System Groups

Spacewalk enables 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 enables you to display errata according to whether they are noncritical, bug fix advisory, product enhancement advisory, or security advisory.

You can also use the `spacecmd` command with the following subcommands to monitor and manage errata:

- `system_listerrata` – Provides similar functionality. For example, list the errata that are available for a specified system.

- `erratadetails` – Displays additional information about a given erratum.

- `system_applyerrata` – Applies errata to multiple systems within a system group at the same time.

If you 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 the system 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 the need to reboot.

Chapter 10 For More Information

For more information about Spacewalk 2.6, see *Spacewalk for Oracle® Linux Documentation*.

For more information about Oracle Linux, Ksplice, and ULN, see the *Oracle® Linux 6 Documentation* and *Oracle® Linux 7 Documentation*

For more information about the Spacewalk open source project, go to https://spacewalkproject.github.io.