Spacewalk for Oracle® Linux

Client Life Cycle Management Guide for Release 2.7

E90694-12
February 2021

Oracle Legal Notices

Copyright © 2018, 2021, Oracle and/or its affiliates.

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 embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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 Inside 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, Epyc, and the AMD 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 how to use Spacewalk 2.7 to provision and manage Spacewalk clients.

Document generated on: 2021-02-08 (revision: 664)

Table of Contents

Preface

Spacewalk for Oracle® Linux: Client Life Cycle Management Guide for Release 2.7 describes how to use the Spacewalk 2.7 web interface and `spacecmd` command-line utility to provision and manage Spacewalk clients.

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:

Oracle® Linux Manager & 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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle recognizes the influence of ethnic and cultural values and is working to remove language from our products and documentation that might be considered insensitive. While doing so, we are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is an ongoing, long-term process.

Chapter 1 Using the Spacewalk Web Interface and the spacecmd Command

You can use either the Spacewalk web interface or the `spacecmd` command to administer Spacewalk.

The Spacewalk Server for Release 2.7 is supported on Oracle Linux 6 and Oracle Linux 7. The Spacewalk Client for Release 2.7 is supported on Oracle Linux 6 and Oracle Linux 7, as well as Oracle Linux 5.

You can also create your own web and command interfaces by using the Spacewalk XML/RPC API. For more information, see Appendix D, *Spacewalk XML/RPC API*.

For an introduction to the concepts and features of Spacewalk and best practices for using Spacewalk for managing Oracle Linux systems, see *Spacewalk for Oracle® Linux: Concepts Guide for Release 2.7*.

For requirements and instructions on installing Spacewalk, see see *Spacewalk for Oracle® Linux: Installation Guide for Release 2.7*.

1.1 About the Spacewalk Web Interface

After installing a Spacewalk server, you will need to set up the main Spacewalk administrator account. Starting with Spacewalk 2.6, you must also create an initial organization as part of the Spacewalk setup.

To launch the Spacewalk web interface, point a browser at the URL that is specified in the output of Step 7 of *Installing a Spacewalk Server* in see *Spacewalk for Oracle® Linux: Installation Guide for Release 2.7*.

The following example is similar to the output that is displayed:

```
Installation complete.
Visit https://swksvr.mydom.com to create the Spacewalk administrator account.
```

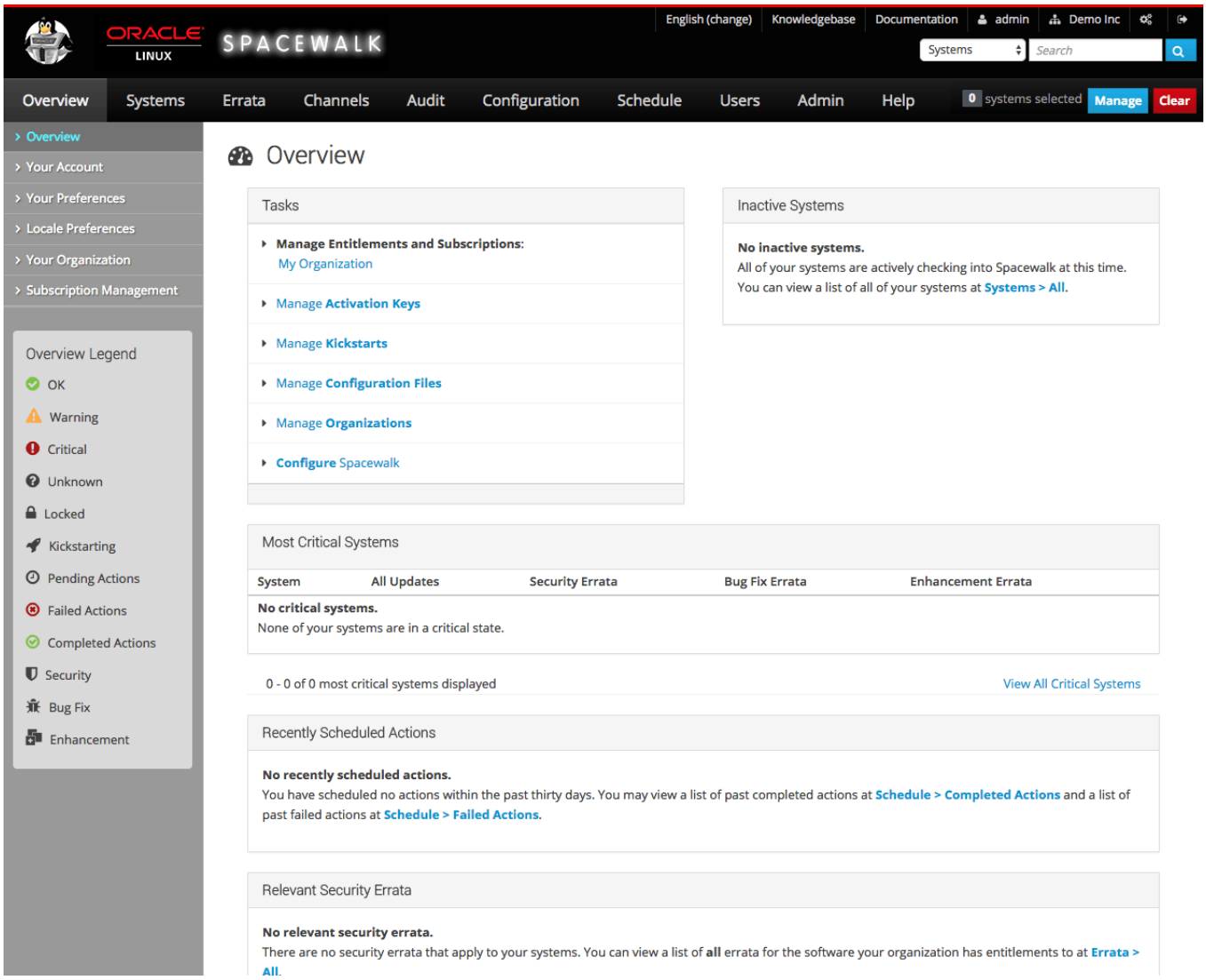
Create the administrator account and an initial organization (if applicable to the Oracle Linux release that you are running), then log in to the Spacewalk web interface. For step-by-step instructions, see *Configuring an Initial Spacewalk Administrator and Organization* in *Spacewalk for Oracle® Linux: Installation Guide for Release 2.7*..

After configuring the main administrator account and an organization, you can log in to the Spacewalk web interface, where the Overview page is displayed. This page shows the individual Spacewalk administrative areas, menus, page views, and various actions that you can perform by using the web interface.

Overview Page

The Spacewalk web interface menu header has the following administrative areas that you can select. The default page for each menu item displays summary information. You can obtain more detailed information and perform actions on the various items by selecting the item from the left-hand menu or by clicking the tab views on a given page.

Figure 1.1 Spacewalk Overview Page



The default Overview page presents a dashboard view of the state of the Spacewalk server. The page displays important information about systems that are inactive or in a critical state, recently scheduled actions, relevant security errata that you can apply to your systems, and lists of system groups and recently registered systems. The page also provides links to administrative tasks that you can perform.

Refer to the Overview Legend pane for the meaning of any icons that the page displays for a system.

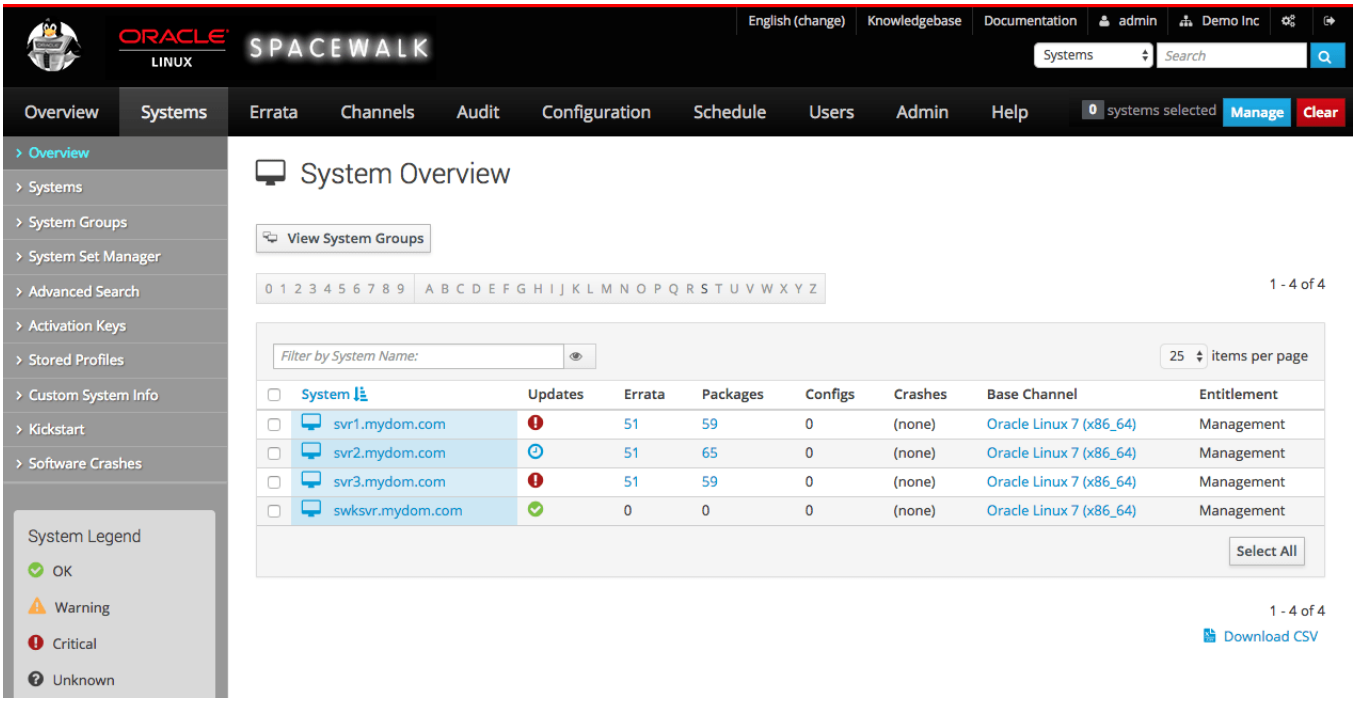
To customize the layout of the Overview page:

1. Select **Your Preferences**.
2. On the Your Preferences page in the Overview Start Page section, select or deselect the check boxes against the information that you want or do not want to display.

   You can configure other preferences on this page, such as whether to receive email notifications, the number of entries per-page in lists, and the separator character for comma-separated values (CSV) files.
3. To save your changes, click **Save Preferences**.

Systems Tab

Figure 1.2 Systems Overview Page



The System Overview page displays a summary of the number of available updates, errata, packages, configuration files and crashes, the name of the base channel, and any entitlements for each managed client system.

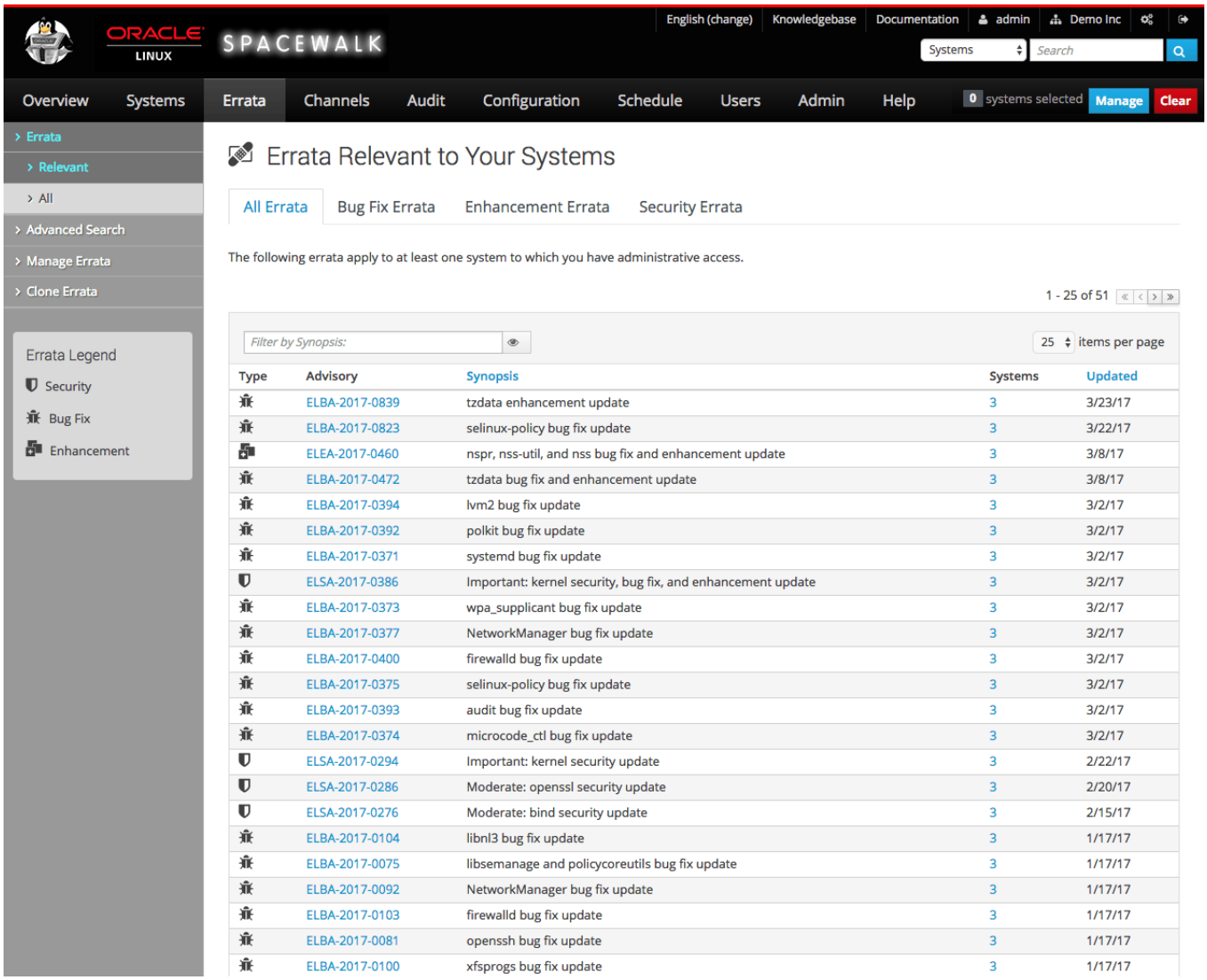
Note

Starting with Spacewalk 2.6, the Management entitlement is automatically assigned to all registered clients. You can also enable the Virtualization entitlement to enable Spacewalk to identify and manage virtual guests on a client server. All other entitlements have been removed.

Refer to the System Legend pane for the meaning of any icons that the page displays for a system.

Errata Tab

Figure 1.3 Errata Relevant to Your Systems Page

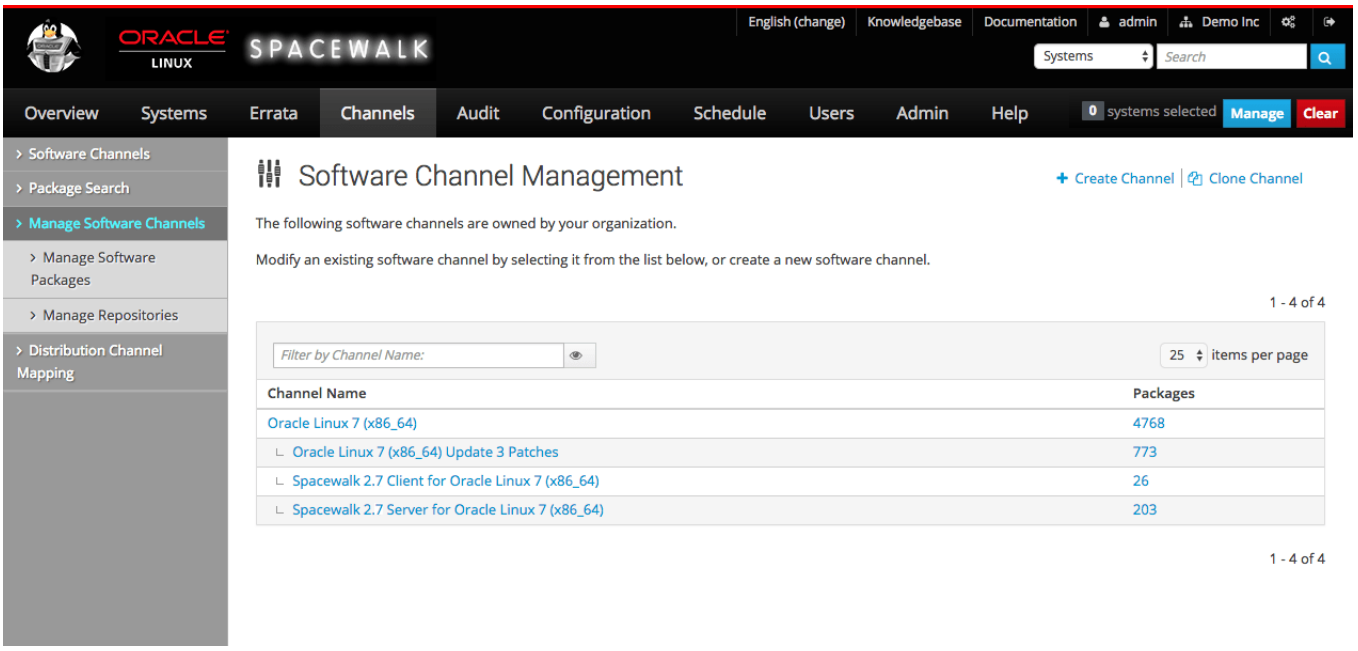


The Errata Relevant to Your Systems page displays information about the errata that are available for your registered systems.

Refer to the Errata Legend pane for the meaning of any icons that the page displays for a system.

Channels Tab

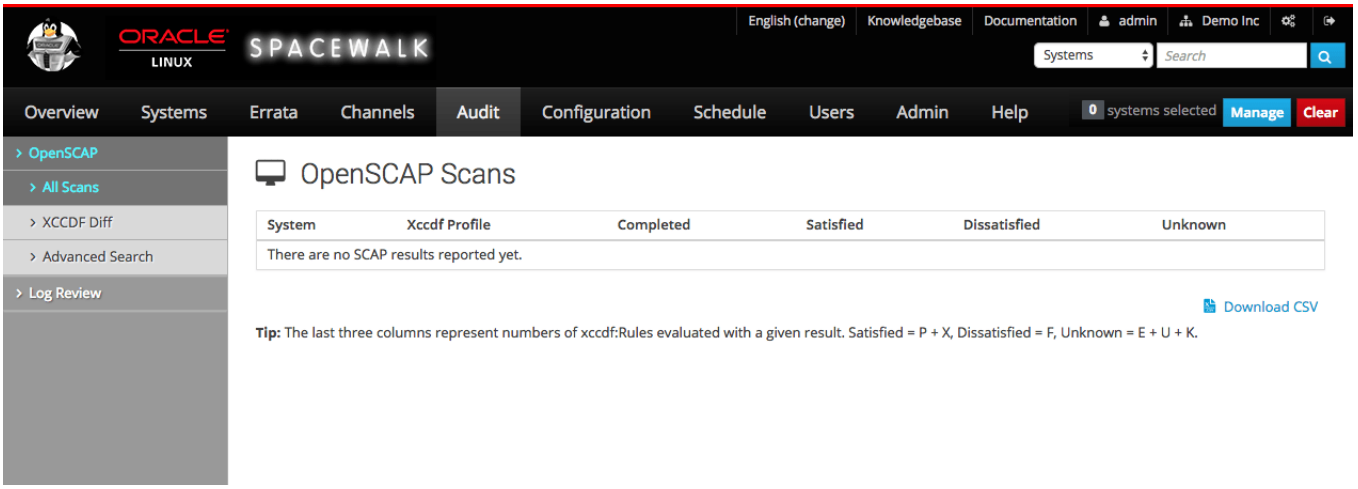
Figure 1.4 Full Software Channel List Page



The Full Software Channel List page displays the channels to which you can subscribe your registered systems. By default, only the base channels are shown. To display child channels, click the plus sign (**+**) next to the name of the base channel.

Audit Tab

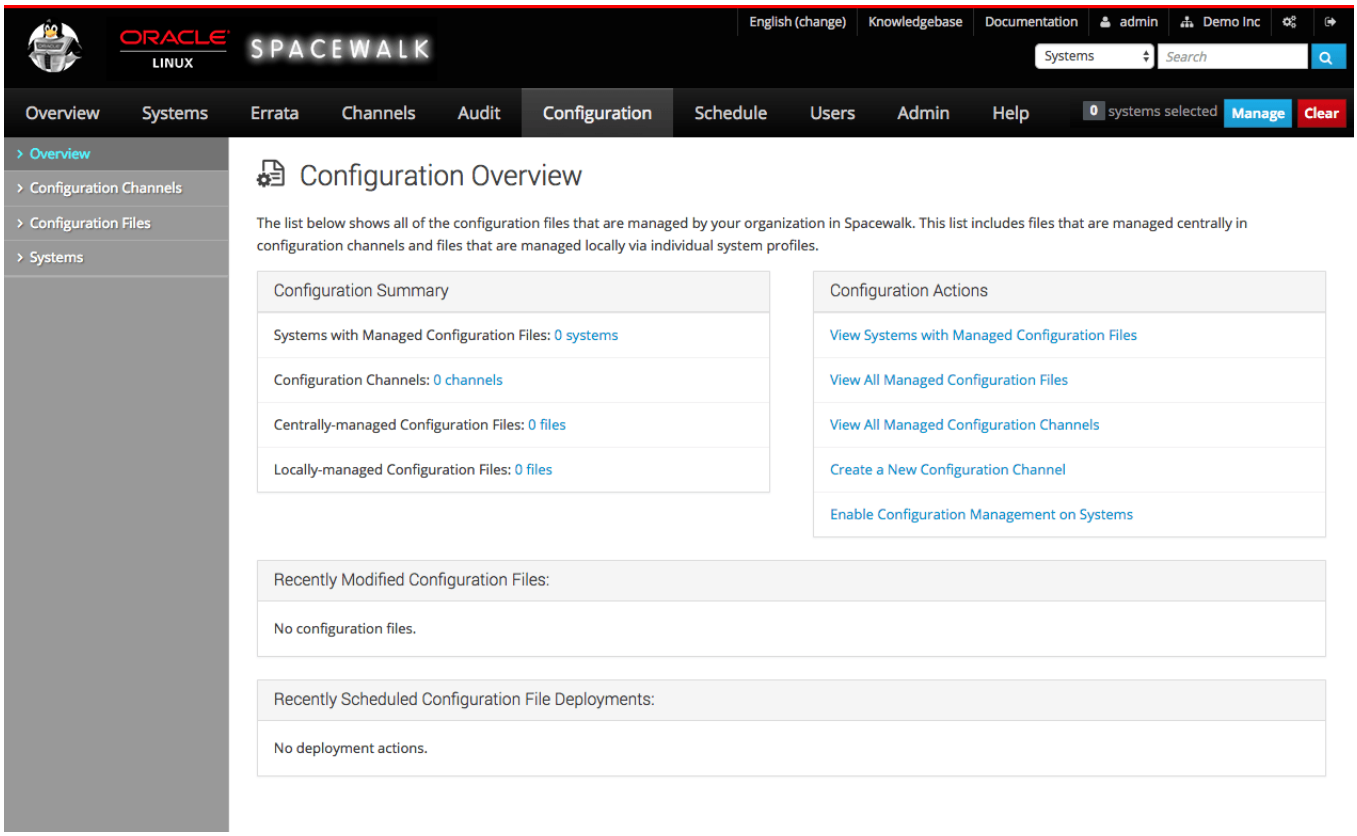
Figure 1.5 OpenSCAP Scans Page



The OpenSCAP Scans page displays a summary of any scans that you have performed on your systems.

Configuration Tab

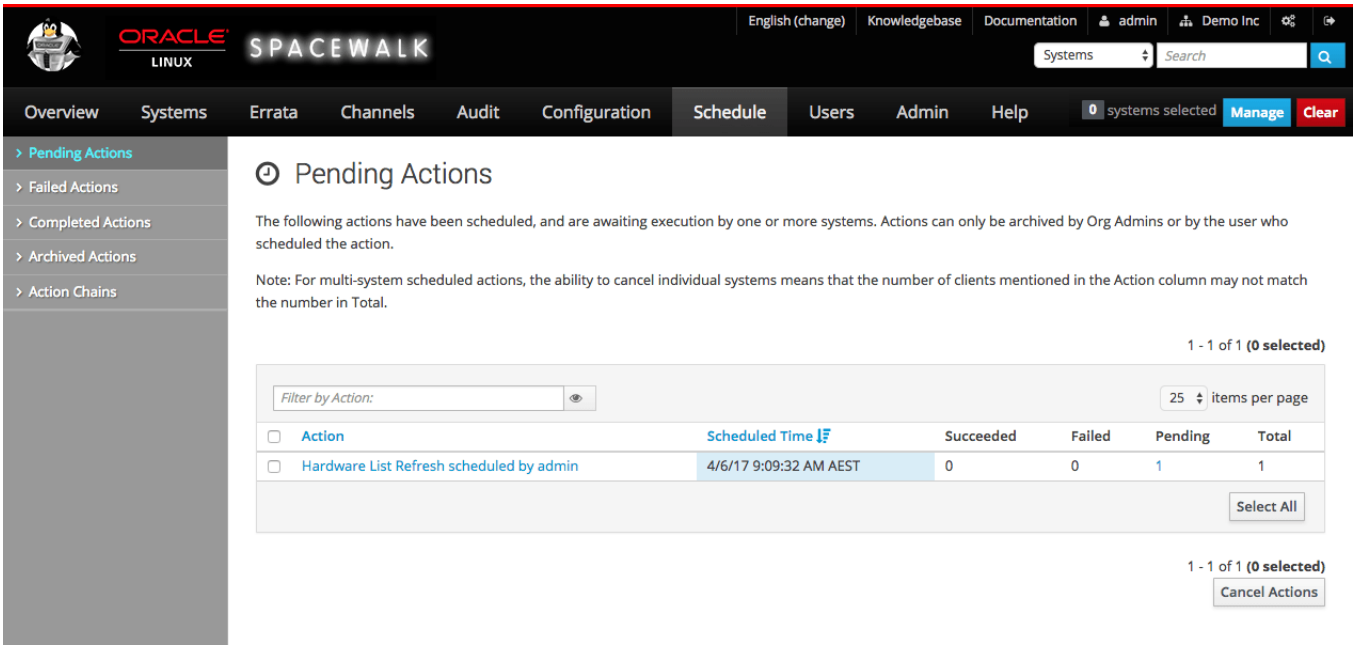
Figure 1.6 Configuration Overview Page



The Configuration Overview page displays a summary of the configuration files that are known to Spacewalk, links to actions you can perform with configuration files, and scheduled deployments of configuration files.

Schedule Tab

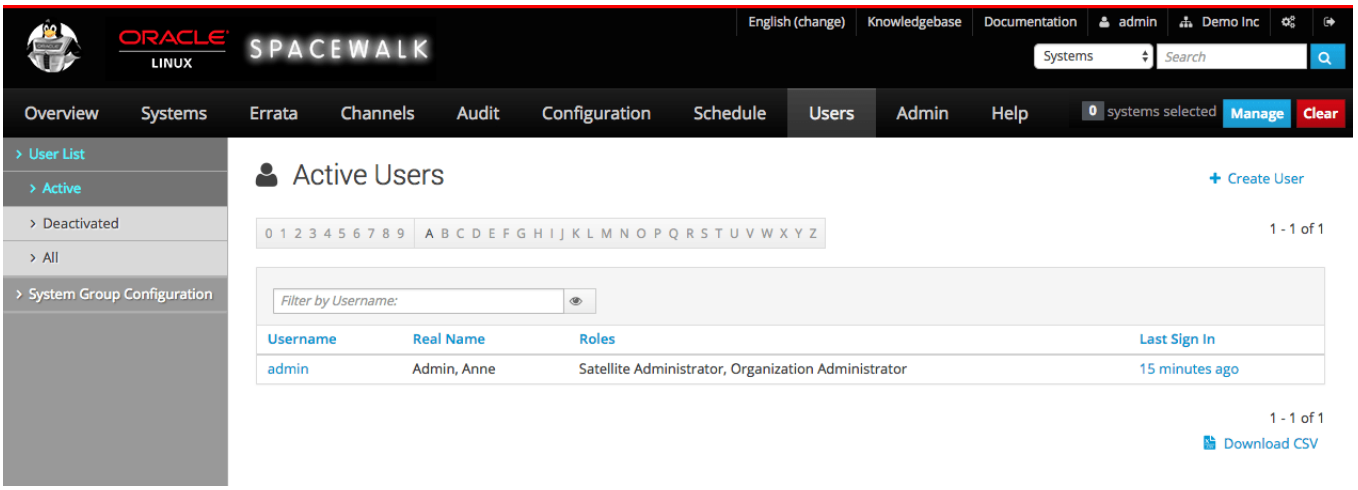
Figure 1.7 Pending Actions Page



The Pending Actions page displays a list of actions that are scheduled to be performed.

Users Tab

Figure 1.8 Active Users Page



The Active Users page displays a list of administrators or other users and their allocated roles.

The Spacewalk Administrator role permits a user to perform all actions in Spacewalk.

You can configure the Organization Administrator role to grant one or more of the following roles to a user who has administrative access to one or more organizations:

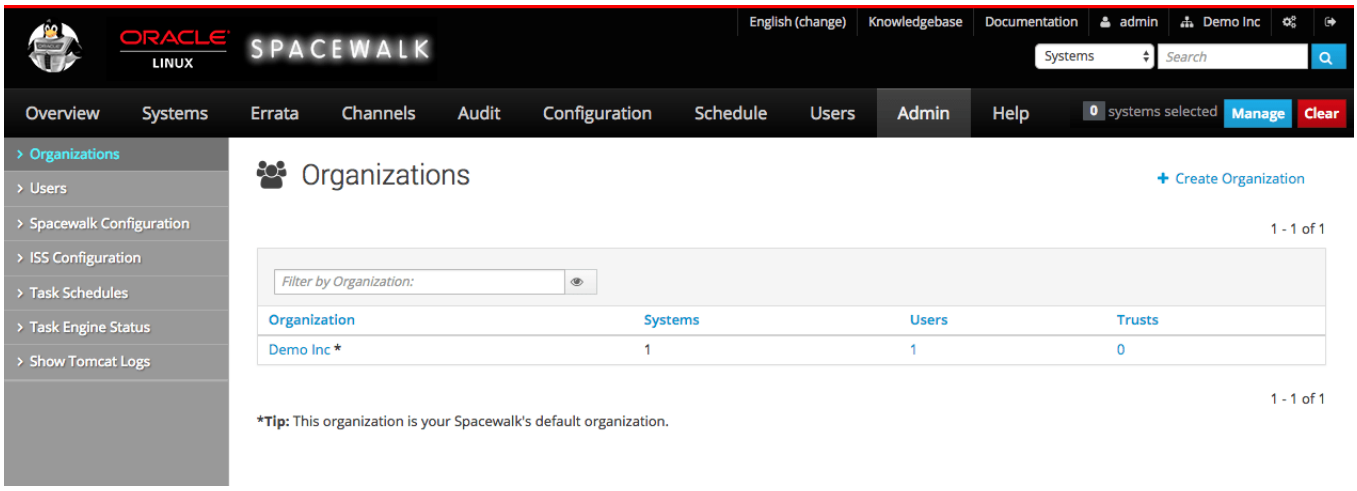
- System Group Administrator

- Channel Administrator
- Activation Key Administrator
- Configuration Administrator
- Monitoring Administrator

A read-only API user has limited access to the XML/RPC API but cannot access the web interface.

Admin Tab

Figure 1.9 Organizations Page



The Organizations page displays the organizations for which you have configured the Spacewalk server to administer and the number of systems, Spacewalk administrators, and trusts that are configured for the system. If a trust is enabled, you can share content and move systems between organizations.

Note

Starting with Spacewalk 2.6, you must define at least one Spacewalk organization as soon as you have installed the Spacewalk server.

1.2 About the spacecmd Command

You can use the `spacecmd` command-line interface (CLI) to perform most of the actions that you perform by using the web interface.

You can run the `spacecmd` command either directly on the Spacewalk server or remotely. If you run the `spacecmd` command remotely, you must specify the server by its IP address or by a resolvable domain name.

The following example shows how you would specify the server by its host name:

```
$ spacecmd -s swksvr.mydom.com
Welcome to spacecmd, a command-line interface to Spacewalk.

Type: 'help' for a list of commands
      'help <cmd>' for command-specific help
      'quit' to quit
```

```
Spacewalk Username: swadmin
Spacewalk Password: password
INFO: Connected to https://swksvr.mydom.com/rpc/api as swadmin
```

where the values for the host name, Spacewalk user, and password are those that you specified when installing and configuring Spacewalk.

Note

You must authenticate yourself as a Spacewalk user with an assigned role with sufficient privileges for performing the requested actions.

As an alternative to entering the Spacewalk user name and password at the prompts, you can use the `-u` and `-p` options to specify these values. Be aware however, that specifying the password on the command line is insecure, as it is visible in command histories and the process list. A more secure method is to create a credentials file, such as `~/.spacecmd/config` with mode 400 that contains information that defines the Spacewalk server, user name, and password, as shown in the following example:

```
[spacecmd]
server=swksvr.mydom.com
username=swadmin
password=password
```

Note the following additional information about using the `spacecmd` CLI:

- To display a list of `spacecmd` shell commands, type `help`.
- To display more help about a command, type `help command`.
- To exit the shell, type `exit` or `quit`.
- The `spacecmd` command attempts `[Tab]` completion of partial commands or arguments.
- You can run the `spacecmd` command as an interactive shell or non-interactively. This guide includes examples of using the interactive shell.

  If you want to run the `spacecmd` command non-interactively, specify the `spacecmd` shell command and its arguments after a `--` delimiter, as shown in the following example:

```
$ spacecmd -- softwarechannel_create -l oraclelinux7-x86_64-ksplice \
  -n "Oracle Linux 7 x86_64 Ksplice Channel" -p oraclelinux7-x86_64 -a x86_64
INFO: Connected to https://swksvr.mydom.com/rpc/api as swadmin
$ spacecmd -q -- softwarechannel_list
INFO: Connected to https://swksvr.mydom.com/rpc/api as swadmin
oraclelinux7-x86_64
oraclelinux7-x86_64-addons
oraclelinux7-x86_64-ksplice
oraclelinux7-x86_64-optional
oraclelinux7-x86_64-patch
oraclelinux7-x86_64-spacewalk26-client
oraclelinux7-x86_64-spacewalk26-server
oraclelinux7-x86_64-uek-r3
oraclelinux7-x86_64-uek-r4
$ spacecmd -q -y -- softwarechannel_delete oraclelinux7-x86_64-ksplice
Channels
--------
oraclelinux7-x86_64-ksplice
$ spacecmd -q -- softwarechannel_list
oraclelinux7-x86_64
oraclelinux7-x86_64-addons
```

```
oraclelinux7-x86_64-optional
oraclelinux7-x86_64-patch
oraclelinux7-x86_64-spacewalk26-client
oraclelinux7-x86_64-spacewalk26-server
oraclelinux7-x86_64-uek-r3
oraclelinux7-x86_64-uek-r4
```

The `-q` option suppresses informational messages. The `-y` option specifies a **yes** answer to all prompts to confirm whether you want to delete or change data. By default, `spacecmd` assumes the answer **no**.

For more information, see the `spacecmd(1)` man page.

Chapter 2 Creating Software Channels and Repositories

This chapter describes how to create software channels in Spacewalk, for which client systems can subscribe and obtain packages and errata. Each channel is associated with at least one repository, which defines the source of the packages and errata.

The examples in this chapter use the Unbreakable Linux Network (ULN) and the Oracle Linux yum server. However, you can also use Spacewalk to obtain software packages from other internal or external sources.

2.1 About Software Channel Configuration

ULN provides more than 100 unique channels, which support the i386, x86_64, IA64, and the 64-bit Arm architectures, for releases of Oracle Linux 4 update 6 and later. To access ULN, go to https://linux.oracle.com/.

ULN provides an `ol*N*_*arch*_latest` repository, all the latest versions of the packages in a distribution, including any errata that are also provided in the patch channel. If no vulnerabilities have been found in a package, the package version might be the same as that included in the original distribution. For other packages, the version is the same as that provided in the patch channel for the highest update level. For example, the `ol6_*arch*_latest` channel for Oracle Linux 6 Update 3 contains a combination of the `ol6_u3_*arch*_base` and `ol6_u3_*arch*_patch` channels. which includes all of the packages for an Oracle Linux release.

The Oracle Linux yum server provides the `ol*N*_latest` repository, which includes all of the packages for an entire Oracle Linux release, in addition to the `ol*N*_u*N*_base` repository for each update. Unlike ULN, the Oracle Linux yum server does not provide `patch` channels for updates.

Some channels, such as those for DTrace userspace, Ksplice, and OFED, are available on ULN, but not on the Oracle Linux yum server.

ULN also provides older versions of packages that are added to a parent channel. The `_archive` suffix is usually added to the channel for which it hosts archive packages. For example, the `_latest` channels have equivalent `_latest_archive` channels to host older versions of packages that have been updated in the `_latest` channels.

Packages are moved to an archive channel when newer versions of the same packages are added to the parent channel. The addition of archive channels helps keep the metadata for the parent channel manageable and also keeps the overall size of the channel down to a minimum. If you require an earlier version of a package, you can subscribe to the equivalent `_archive` channel to obtain it. When performing an installation or downgrade, you must specify the version of the package that you want to install.

Note

Installing packages from an `_archive` channel might result in your system running software that has since been patched for security related issues. This could open your system up to vulnerabilities that could be exploited for malicious purposes.

ULN also provides `base` and `patch` channels for each update of an Oracle Linux release. Depending on the Oracle Linux release, other channels might provide the latest packages for additional features such as DTrace user-space, Ksplice, and the OpenFabric Enterprise Edition (OFED) feature.

Other channels might also be available, such as `_beta` channels for the beta versions of packages.

As each new, major version or minor update of Oracle Linux becomes available, Oracle creates new base and patch channels for each supported architecture to distribute new packages. The existing base

and patch channels for the previous versions or updates remain available and do not include the new packages. The `_latest` channel distributes the latest possible version of any package, and tracks the top of the development tree independently of the update level.

For a complete description of the main channels that are available on ULN, see the chapter on ULN registration in *Oracle® Linux: Unbreakable Linux Network User's Guide for Oracle Linux 6 and Oracle Linux 7*.

Oracle recommends that you design a channel configuration that is based on your particular work flow. For example, if you intend to use Spacewalk's channel cloning feature to promote systems from development through testing to production, you could configure a `base` channel and child `patch` channel together with other child channels.

If you duplicate child channels, you do not need to duplicate their repositories. These channels can use the same repositories as the channel from which they were cloned. For example, each cloned base channel might have a unique `addons` child channel, but each of the child channels would use the same repository. The packages are not duplicated, even though they are referenced in multiple channels.

To make archive channel content available to local clients, you must first create repositories for each ULN archive channel that you want to synchronize, then map these repositories to the software channels that you have created in Spacewalk.

If necessary, you can maintain the `latest` channels separately, without subscribing any systems to these channels. If the need arises, you can copy errata packages form the `latest` channels to the `patch` channels to make the latest fixes available.

Note

You do not need to associate a software channel with a repository if you want to create custom channels that obtain their packages by methods such as `rhnpush` or uploading by using the web interface. These methods enable you to serve locally developed and packaged software that has no upstream repository.

2.2 Configuring Software Channels for ULN

Spacewalk contains a ULN plugin for the `spacewalk-repo-sync` tool. The plugin enables you to synchronize software channels without having to register the Spacewalk server with ULN.

Configure the ULN plugin as follows:

1. Change the mode of the `/etc/rhn/spacewalk-repo-sync/uln.conf` file to `600` (read-write).

```
# chmod 600 /etc/rhn/spacewalk-repo-sync/uln.conf
```

2. Edit the `/etc/rhn/spacewalk-repo-sync/uln.conf` file and add your SSO login user name and password for ULN:

```
[main]
username=ULN_SSO_username
password=ULN_SSO_password
```

3. Change the mode of the `/etc/rhn/spacewalk-repo-sync/uln.conf` file to `400` (read-only).

```
# chmod 400 /etc/rhn/spacewalk-repo-sync/uln.conf
```

Important

To protect your ULN credentials, verify that the `/etc/rhn/spacewalk-repo-sync/uln.conf` file is read-only (file mode `0400`) by `root`.

```
# ls -l /etc/rhn/spacewalk-repo-sync/uln.conf
-r--------. 1 root root 56 Feb  2 14:44 /etc/rhn/spacewalk-repo-sync/uln.conf
```

When you have configured the ULN plugin, you can use either the Spacewalk web interface, the `spacecmd` command, or the `spacewalk-common-channels` command to create Spacewalk software channels, repositories, and activation keys, as follows:

- Section 2.5, “Working With Software Channels”

Tip

Although the `spacewalk-common-channels` command configures software channels to access the Oracle Linux yum server, you can reconfigure the repository entries to access ULN instead. For example, if you want to use the Oracle Linux `base` and `patch` channels for an Oracle Linux release update on ULN, you can reconfigure the base software channel to access the `base` channel and create an additional child channel and associated repository entry for the `patch` channel.

See Section 2.3, “Configuring Software Channels to Obtain Packages From the Oracle Linux Yum Server”.

- Section 2.4, “Working With Repositories”
- Chapter 3, *Working With Activation Keys in Spacewalk*

After you have set up the software channels and repositories, download the packages by synchronizing the software channels with ULN. See Section 2.6, “Synchronizing Software Channels”.

2.3 Configuring Software Channels to Obtain Packages From the Oracle Linux Yum Server

You can use the `spacewalk-common-channels` command, which is in the `spacewalk-utils` package, to configure software channels that use the Oracle Linux yum server. In addition to configuring software channels, you can use this command to configure repositories, GPG keys, and activation keys for Oracle Linux 6 and Oracle Linux 7.

To list the available channels, use the `--list` option:

```
# spacewalk-common-channels --list | grep "^ oracle"
 oraclelinux6:        i386, x86_64
 oraclelinux6-addons: i386, x86_64
 oraclelinux6-mysql55: i386, x86_64
 oraclelinux6-mysql56: i386, x86_64
 oraclelinux6-mysql57: i386, x86_64
 oraclelinux6-openstack30: x86_64
 oraclelinux6-playground: x86_64
 oraclelinux6-scl:    x86_64
 oraclelinux6-spacewalk22-client: i386, x86_64
 oraclelinux6-spacewalk22-server: x86_64
 oraclelinux6-spacewalk24-client: i386, x86_64
 oraclelinux6-spacewalk24-server: x86_64
 oraclelinux6-spacewalk26-client: i386, x86_64
 oraclelinux6-spacewalk26-server: x86_64
 oraclelinux6-spacewalk27-client: i386, x86_64
 oraclelinux6-spacewalk27-server: x86_64
 oraclelinux6-uek-r2: i386, x86_64
 oraclelinux6-uek-r3: x86_64
 oraclelinux6-uek-r4: x86_64
```

```
oraclelinux7:        x86_64
oraclelinux7-addons: x86_64
oraclelinux7-ceph:   x86_64
oraclelinux7-mysql55: x86_64
oraclelinux7-mysql56: x86_64
oraclelinux7-mysql57: x86_64
oraclelinux7-openstack20: x86_64
oraclelinux7-openstack21: x86_64
oraclelinux7-openstack30: x86_64
oraclelinux7-openstack30-extras: x86_64
oraclelinux7-optional: x86_64
oraclelinux7-scl:    x86_64
oraclelinux7-spacewalk22-client: x86_64
oraclelinux7-spacewalk22-server: x86_64
oraclelinux7-spacewalk24-client: x86_64
oraclelinux7-spacewalk24-server: x86_64
oraclelinux7-spacewalk26-client: i386, x86_64
oraclelinux7-spacewalk26-server: x86_64
oraclelinux7-spacewalk27-client: i386, x86_64
oraclelinux7-spacewalk27-server: x86_64
oraclelinux7-uek-r3: x86_64
oraclelinux7-uek-r4: x86_64
```

Note

Unlike ULN, the Oracle Linux yum server does not provide `patch` channels for each update of an Oracle Linux release. Instead, the `spacewalk-common-channels` command configures the base (parent) software channel to use the `olN_latest` repository, which includes all of the packages for the entire release.

Some ULN channels, such as those for DTrace user space, Ksplice, and OFED, are not available on the Oracle Linux yum server.

For example, you would create the software channels for Oracle Linux 7 (x86_64) as follows:

```
# spacewalk-common-channels -v -u swadm -p swadm_passwd -a x86_64 -k unlimited 'oraclelinux7*'
Connecting to http://localhost/rpc/api
Base channel 'Oracle Linux 7 (x86_64)' - creating...
* Activation key 'oraclelinux7-x86_64' - creating...
* Child channel 'Oracle Linux 7 Addons (x86_64)' - creating...
** Activation key '1-oraclelinux7-x86_64' - adding child channel...
* Child channel 'Oracle Linux 7 MySQL 5.5 (x86_64)' - creating...
** Activation key '1-oraclelinux7-x86_64' - adding child channel...
* Child channel 'Oracle Linux 7 MySQL 5.6 (x86_64)' - creating...
** Activation key '1-oraclelinux7-x86_64' - adding child channel...
* Child channel 'Oracle Linux 7 MySQL 5.7 (x86_64)' - creating...
** Activation key '1-oraclelinux7-x86_64' - adding child channel...
...
```

where *`swadm`* and *`swadm_passwd`* are the user name and password of the Spacewalk administrator. The `-k unlimited` option specifies that the command should create an activation key with no limit on the number of servers with which you can use it.

Note

If you omit this option, no activation key is created.

You can use either the Spacewalk web interface or the `spacecmd` command to display, modify, or delete available channels, repositories, and activation keys.

See the following for more information and instructions:

- Section 2.5, “Working With Software Channels”

- Section 2.4, “Working With Repositories”
- Chapter 3, *Working With Activation Keys in Spacewalk*

After you have set up the software channels and repositories, download the packages by synchronizing the software channels with the Oracle Linux yum server. See Section 2.6, “Synchronizing Software Channels”.

2.4 Working With Repositories

Spacewalk repositories define where to obtain packages from ULN or the Oracle Linux yum server.

For ULN, a Spacewalk repository specifies the URL of a ULN channel in the following format:

```
uln:///ULN_channel_label
```

You can obtain a list of available ULN channel labels by logging into ULN (https://linux.oracle.com) and then selecting the **Channels** tab.

The URL must contain three forward slash (`/`) characters, as shown in this example:

```
uln:///ol6_x86_64_latest
```

For the Oracle Linux yum server, a Spacewalk repository specifies the URL of an Oracle Linux yum server repository by using the following format:

```
https://yum.oracle.com/repository_path
```

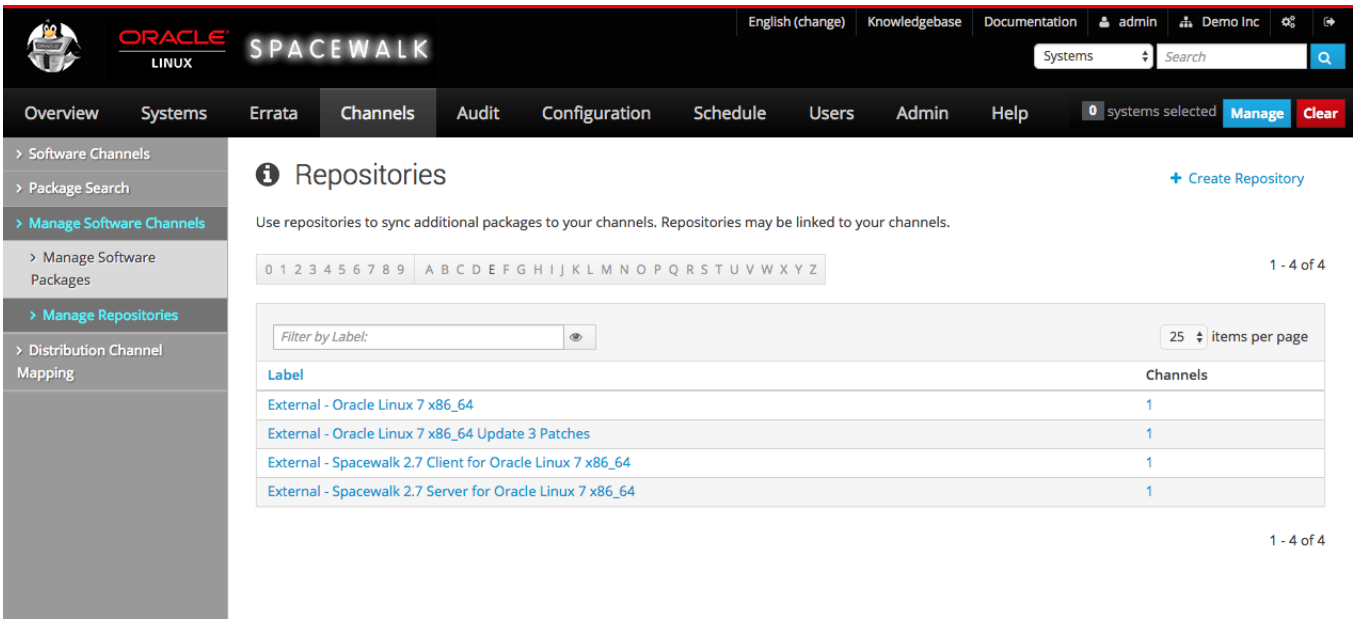
You can obtain the URLs from the Oracle Linux yum server `repo` files at https://yum.oracle.com/.

Because each Spacewalk repository is specific to the i386 or x86_64 architecture, replace `$basearch` with the architecture, as shown in the following example:

```
https://yum.oracle.com/repo/OracleLinux/OL6/6/base/x86_64/
```

2.4.1 Working With Repositories by Using the Spacewalk Web Interface

Figure 2.1 Repositories Page



Go to **Channels**, select **Manage Software Channels**, and then select **Manage Repositories**:

- To create a repository:

  1. Click **+ create new repository**.
  2. On the Create New Repository page, enter the following repository settings:

     Repository Type — Specify `yum` for the Oracle Linux yum server repositories or `uln` for the ULN repositories.

     Repository Label — Enter a name for the repository, for example, `Oracle Linux 6 (x86_64)`.

     Repository URL — Enter the URL of the source for the repository's packages. For example, `uln:///ol6_x86_64_latest` or `https://yum.oracle.com/repo/OracleLinux/OL6/6/base/x86_64/`.

     Leave the remaining fields unset for Oracle Linux.
  3. Click **Create Repository** to create the repository.

- To view a repository, select its entry to display its details.
- To modify a repository:

  1. Select the repository that you want to edit.
  2. On the Repository Details page, modify the repository settings and click **Update Repository** to save your changes.

- To delete a repository:

  1. Go to **Channels**, select **Manage Software Channels**, and then select **Manage Repositories**.
  2. Select the repository that you want to delete.
  3. On the Repository Details page, click **delete repository** and then click **Delete Repository** to confirm.

To associate a software channel with a repository, see Section 2.5.1, “Working With Software Channels by Using the Spacewalk Web Interface”.

2.4.2 Working With Repositories by Using the spacecmd Command

Create a repository by using the `repo_create` command as follows:

```
spacecmd {SSM:0}> repo_create
Name: Ksplice for Oracle Linux 7
URL: uln:///ol7_x86_64_ksplice
```

To list all repositories, use the `repo_list` command:

```
spacecmd {SSM:0}> repo_list
External - Addons for Oracle Linux 6 (x86_64)
External - CentOS 6 (i386)
External - CentOS 6 (x86_64)
```

```
External - Latest Unbreakable Enterprise Kernel Release 2 for Oracle Linux 6 (x86_64)
External - Latest Unbreakable Enterprise Kernel Release 3 for Oracle Linux 6 (x86_64)
External - Latest Unbreakable Enterprise Kernel Release 4 for Oracle Linux 6 (x86_64)
External - MySQL 5.5 for Oracle Linux 6 (x86_64)
External - MySQL 5.6 for Oracle Linux 6 (x86_64)
External - OpenStack 3.0 packages for Oracle Linux 6 (x86_64)
External - Oracle Linux 6 (x86_64)
External - Playground (Mainline) Kernels for Oracle Linux 6 (x86_64)
External - Software Collection Library packages for Oracle Linux 6 (x86_64)
External - Spacewalk 2.2 Client for Oracle Linux 6 (x86_64)
External - Spacewalk 2.2 Server for Oracle Linux 6 (x86_64)
External - Spacewalk 2.4 Client for Oracle Linux 6 x86_64
External - Spacewalk 2.4 Server for Oracle Linux 6 (x86_64)
External - Spacewalk 2.6 Server for Oracle Linux 6 (x86_64)
External - Spacewalk 2.7 Client for Oracle Linux 6 (x86_64)
External - Spacewalk 2.7 Server for Oracle Linux 6 (x86_64)
epel6-p-repo
ol5-i386-repo
ol5-i386-repo-test
ol5-x64-repo
ol6-i386-repo
ol6-x64-repo
ol7-repo
repo1-label
repo2-label
uln_repo
```

To list the details of a repository, use the `repo_details` command:

```
spacecmd {SSM:0}> repo_details "External - Oracle Linux 6 \(x86_64\)"
Repository Label:                 External - Oracle Linux 6 (x86_64)
Repository URL:                   https://yum.oracle.com/repo/OracleLinux/OL6/latest/x86_64/
Repository Type:                  yum
Repository SSL Ca Certificate:    None
Repository SSL Client Certificate: None
Repository SSL Client Key:        None
```

The parentheses in the name must be escaped with backslashes to protect them from the shell.

To delete a repository, use the `repo_delete` command:

```
spacecmd {SSM:0}> repo_delete "Ksplice for Oracle Linux 6"
Repos
-----
Ksplice for Oracle Linux 6

Delete these repos [y/N]: y
```

To associate a software channel with a repository, see Section 2.5.2, “Working With Software Channels by Using the softwarechannel_create Command”.

2.5 Working With Software Channels

The main software channel for an Oracle Linux release is termed a *base* (or *parent*) software channel. You can associate a number of child software channels with the base software channel. Each child software channel usually provides packages that are not available with the base software channel. If multiple versions of a package exist in different subscribed channels, `yum` versioning and dependency resolution ensure that the most up-to-date version of a package is installed.

You can subscribe a client to a single base channel and one or more of its child channels. For channels that are not specific to an update, such as `addons`, you can create an `addons` child channel for each update-level base channel and associate this child channel with the same `addons` repository.

If you set up Spacewalk to obtain Oracle Linux packages from ULN, Oracle recommends that you configure a separate `olN_arch_un_base` base software channel and `olN_arch_un_patch` child software channel for each update of Oracle Linux as it becomes available. This practice ensures that software channels stay small and helps to speed up channel cloning. Client systems are not upgraded across update levels unless you either change the source channel used for channel cloning or reconfigure the channels to which a client system subscribes.

The following example illustrates a typical configuration of the base and child software channels for Oracle Linux, where the base and patch channels are synchronized with ULN:

```
oraclelinux7-x86_64-base
 |-- oraclelinux7-x86_64-addons
 |-- oraclelinux7-x86_64-ksplice
 |-- oraclelinux7-x86_64-optional
 |-- oraclelinux7-x86_64-patch
 |-- oraclelinux7-x86_64-spacewalk27-client
 |-- oraclelinux7-x86_64-uek-r4
```

Note

Software channels other than the base and patch software channels do not have to be associated with ULN.

Other channels, such as those for DTrace user-space, Ksplice, and OFED packages, are available on ULN but not the Oracle Linux yum server.

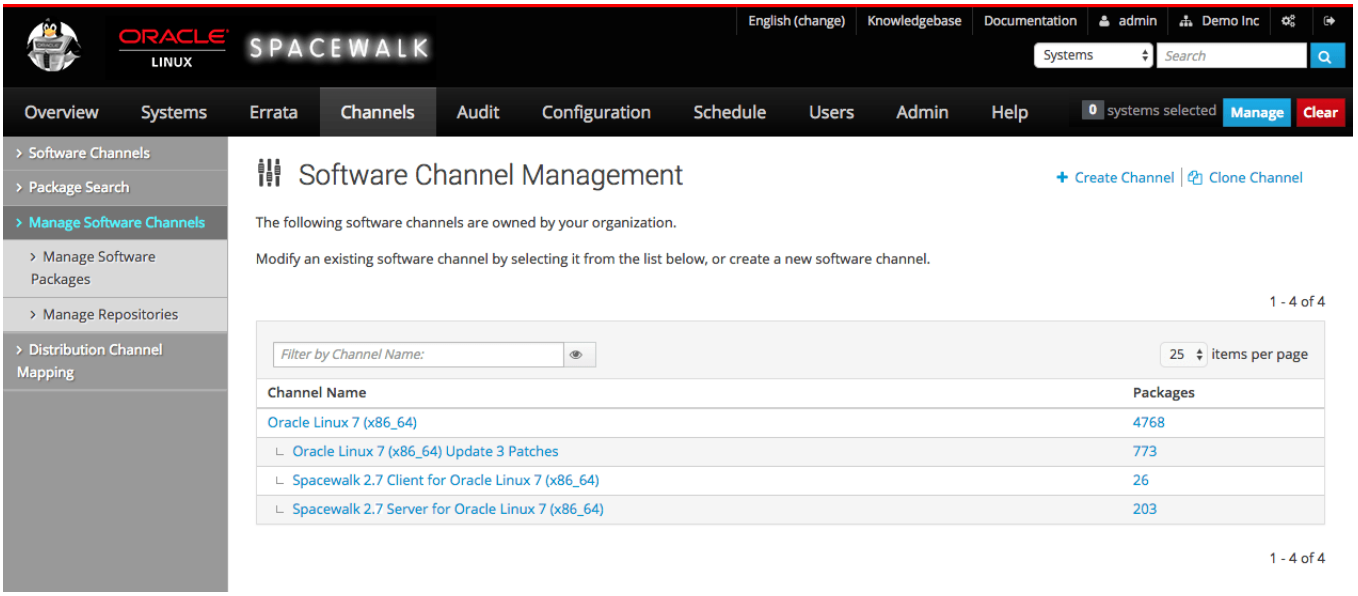
Oracle Linux yum server provides a `_olN_latest` channel, which includes all packages for an entire Oracle Linux release. As individual patch channels for each update are not available, configure the base software channel to use the `_olN_latest` channel. You can use the `spacewalk-common-channels` command to configure the software channels, repositories, GPG keys, and activation keys for Oracle Linux 6 and Oracle Linux 7. See Section 2.3, “Configuring Software Channels to Obtain Packages From the Oracle Linux Yum Server”.

The following example illustrates a typical configuration of the base and child software channels for Oracle Linux, where the base channel is synchronized with the `ol7_latest` channel on the Oracle Linux yum server:

```
oraclelinux7-u4-x86_64-base
 |-- oraclelinux7-u4-x86_64-addons
 |-- oraclelinux7-u4-x86_64-ksplice
 |-- oraclelinux7-u4-x86_64-optional
 |-- oraclelinux7-u4-x86_64-patch
 |-- oraclelinux7-u4-x86_64-spacewalk27-client
 |-- oraclelinux7-u4-x86_64-uek-r4
```

2.5.1 Working With Software Channels by Using the Spacewalk Web Interface

Figure 2.2 Software Channel Management Page



Go to **Channels** and select **Manage Software Channels**:

- To create a software channel:

  1. Click **+ create new channel**.

  2. On the Create Software Channel page, enter channel settings in the following fields, which are the most important for the initial configuration of a channel:

     Channel Name — Enter a descriptive short name for the channel. For example, `Oracle Linux 6 (x86_64) Base`.

     Channel Label — Enter a unique label for the channel that is used by the software. For example, `oraclelinux6-x86_64`.

     Note

     Channel labels *must* be unique across the entire Spacewalk installation. So, if you are going to have multiple child channels that use the same upstream repository, for example, if you have multiple parents, then each child channel must have its own unique label. A recommended practice is to specify the update level for each hierarchy in the label, such as `oraclelinux6-u9-x86_64`.

     Parent Channel — Select **None** if this is a base software channel or select the name of the parent channel if this is a child software channel.

     Architecture — Select **IA32** (for i386 repositories) or **x86_64**, as appropriate.

Yum Repository Checksum Type
For Oracle Linux 6 and Oracle Linux 7, select **sha256**.

Channel Summary
Enter a short, descriptive summary of the channel, for example the channel name.

Note

This field *cannot* be left blank.

Channel Description
Enter a long description of the channel or leave the field blank.

GPG key URL
Enter the URL of the local GPG key. For Oracle Linux, enter `file:///etc/pki/rpm-gpg/RPM-GPG-KEY`.

For third-party repositories, you must import the GPG key into Spacewalk and deploy the key by using provisioning or other method, as appropriate to your site.

Note

The Spacewalk Client requires locally stored GPG keys. Do not use an HTTP based URL. Use a GPG key that you have imported into the local file system.

GPG key ID , **GPG key Fingerprint**
Enter the appropriate key ID and fingerprint for the Oracle Linux release, per the information in following table:

Release	Key ID	Key Fingerprint
Oracle Linux 6	EC551F03	4214 4123 FECF C55B 9086 313D 72F9 7B74 EC55 1F03
Oracle Linux 7	EC551F03	4214 4123 FECF C55B 9086 313D 72F9 7B74 EC55 1F03

3. Click **Create Channel** to create the channel.

- To associate a software channel with a repository:

  1. Select the channel that you want to associate with a repository.
  2. On the Basic Channel Details page, select **Repositories**, select the check box of the repository, and click **Update Repositories**.

- To view a software channel, select its entry to display its details.

  Alternatively, go to **Channels**, select **Software Channels** and click **+** next to the name of the base channel to display its child channels. Select the entry for a software channel to display its details.

- To modify a software channel:

  1. Select the channel that you want to edit.
  2. On the Basic Channel Details page, modify the channel settings and click **Update Channel** to save your changes.

Note

You cannot change the channel label after you have created the channel.

- To delete a software channel:
  1. Select the channel that you want to delete.
  2. On the Basic Channel Details page, click **delete software channel** and then click **Delete Channel** to confirm.

2.5.2 Working With Software Channels by Using the softwarechannel_create Command

To create a software channel, use the `spacecmd` command as follows:

```
spacecmd {SSM:0}> softwarechannel_create
Channel Name: Ksplice for Oracle Linux 7
Channel Label: oraclelinux7-x86_64-ksplice
Base Channels
-------------
oraclelinux6-x86_64
oraclelinux7-x86_64

Select Parent [blank to create a base channel]: oraclelinux7-x86_64

Architecture
------------
i386-sun-solaris
ia32
ia64
ppc
sparc-sun-solaris
x86_64

Select: x86_64

Checksum type
------------
sha256
sha384
sha512

Select: sha1

GPG URL
------------

GPG URL: file:///etc/pki/rpm-gpg/RPM-GPG-KEY

GPG ID
------------

GPG ID: EC551F03

GPG Fingerprint
---------------

GPG Fingerprint: 4214 4123 FECF C55B 9086 313D 72F9 7B74 EC55 1F03
```

To associate a software channel with a repository, use the `softwarechannel_addrepo` command, as shown in this example:

```
spacecmd {SSM:0}> softwarechannel_addrepo ksplice-ol7-x86_64 "Ksplice for Oracle Linux 7"
```

To list all software channels, use the `softwarechannel_list` command:

```
spacecmd {SSM:0}> softwarechannel_list oraclelinux7*
oraclelinux7-x86_64
oraclelinux7-x86_64-addons
oraclelinux7-x86_64-optional
oraclelinux7-x86_64-spacewalk27-client
oraclelinux7-x86_64-spacewalk27-server
oraclelinux7-x86_64-uek-r3
oraclelinux7-x86_64-uek-r4
```

The `oraclelinux7*` argument filters out all channels except those whose labels start with `oraclelinux7`.

To list all base (parent) software channels, use the `softwarechannel_listbasechannels` command:

```
spacecmd {SSM:0}> softwarechannel_listbasechannels
oraclelinux6-x86_64
oraclelinux7-x86_64
```

To list the children of a base software channel, use the `softwarechannel_listchildchannels` command:

```
spacecmd {SSM:0}> softwarechannel_listchildchannels oraclelinux7-x86_64
oraclelinux7-x86_64-addons
oraclelinux7-x86_64-optional
oraclelinux7-x86_64-spacewalk27-client
oraclelinux7-x86_64-spacewalk27-server
oraclelinux7-x86_64-uek-r3
oraclelinux7-x86_64-uek-r4
```

To list the systems that subscribe to a software channel, use the `softwarechannel_listsystems` command:

```
spacecmd {SSM:0}> softwarechannel_listsystems oraclelinux7-x86_64
svr1.mydom.com
svr2.mydom.com
...
```

To display the details of a software channel, use the `softwarechannel_details` command:

```
spacecmd {SSM:0}> softwarechannel_details oraclelinux7-x86_64
Label:              oraclelinux7-x86_64
Name:               Oracle Linux 7 (x86_64)
Architecture:       x86_64
Parent:
Systems Subscribed: 0
Number of Packages: 0

Summary
-------
Oracle Linux 7 (x86_64)

GPG Key:            EC551F03
GPG Fingerprint:    4214 4123 FECF C55B 9086  313D 72F9 7B74 EC55 1F03
GPG URL:            file:///etc/pki/rpm-gpg/RPM-GPG-KEY

Repos
-----
External - Oracle Linux 7 (x86_64)
```

To delete a software channel, use the `softwarechannel_delete` command:

```
spacecmd {SSM:0}> softwarechannel_delete oraclelinux7-x86_64
Channels
--------
oraclelinux7-x86_64

Delete these channels [y/N]: y
```

2.6 Synchronizing Software Channels

After you have configured the software channels and the associated repositories, you can synchronize the software, either by performing an immediate manual synchronization or by scheduling a recurring synchronization job.

At minimum, Oracle recommends that you update the Oracle Linux latest channels daily.

The initial synchronization of the Oracle Linux channels can take several days to complete. Oracle recommends that you perform an initial manual synchronization to populate the channels, and then configure a recurring job to keep them updated.

You can use the Spacewalk web interface, the `spacecmd` command, or the `spacewalk-repo-sync` command to synchronize software channels.

2.6.1 Memory Considerations When Building Repositories

When building repository metadata, Spacewalk can fail with insufficient memory issues. This failure is caused by the default Java memory settings for the Taskomatic daemon, which are set to 512 MB of minimum RAM and 1024 MB of maximum RAM. These settings are defined in the global Spacewalk configuration file `/etc/rhn/rhn.conf`.

If this type of failure occurs, increase the maximum memory to 4096 MB by setting the following property in the configuration file:

```
taskomatic.java.maxmemory=4096
```

However, to assign a value greater than 4096 MB, follow these steps:

1. Disable the automatic memory property settings.
2. Manually add the minimum and maximum Java memory values (`-Xms` and `-Xmx`, respectively) to the `taskomatic.java.additional.1` and `taskomatic.java.additional.2` properties.

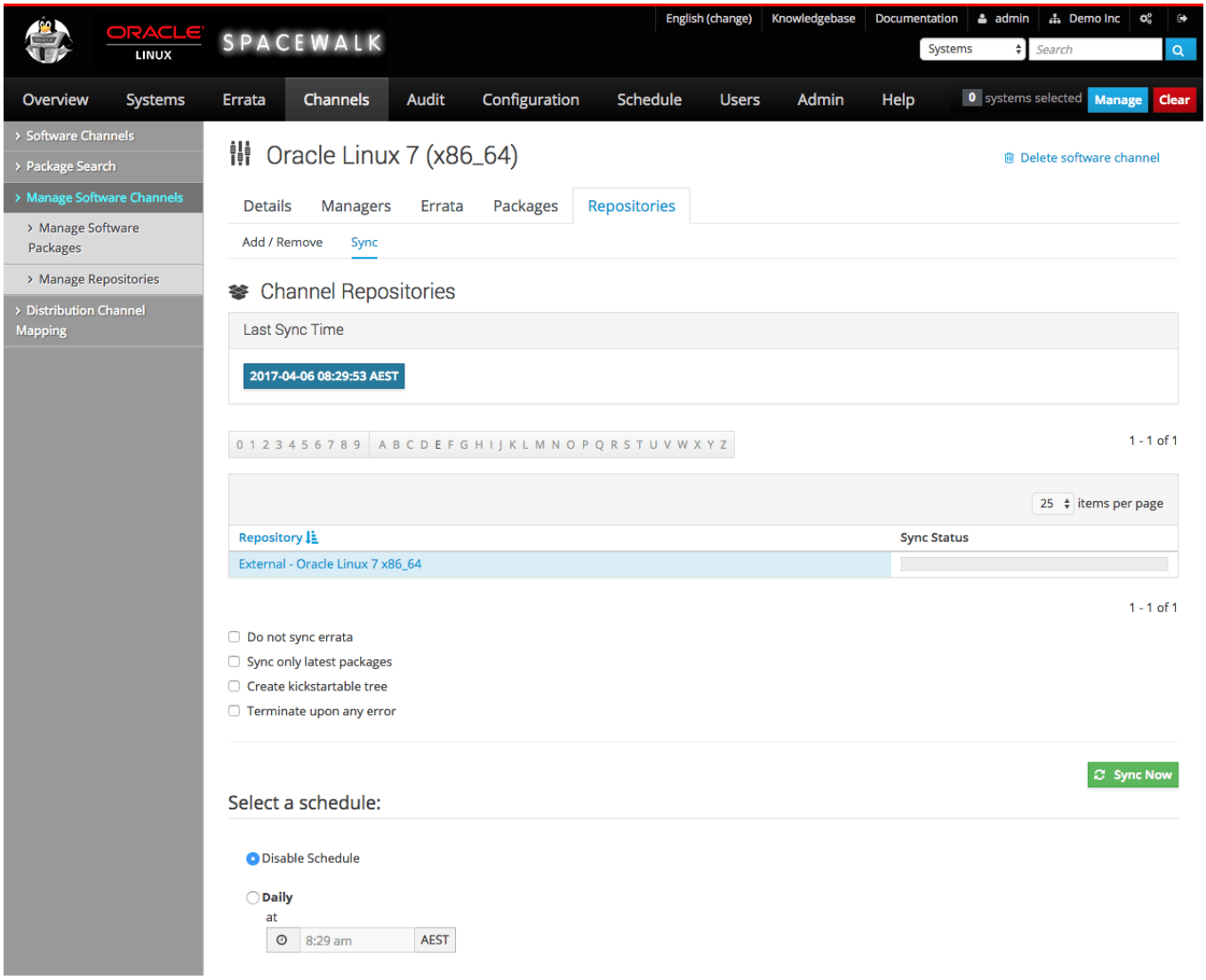
   The configuration file would resemble the following example:

   ```
   taskomatic.java.initmemory=0
   taskomatic.java.maxmemory=0
   taskomatic.java.additional.1=-Xms1024m
   taskomatic.java.additional.2=-Xmx8192m
   ```

Similar memory issues can also occur in the web interface if you have big data sets, such as a large number of servers or packages. To resolve these issues, increase the Tomcat memory limits in the `/etc/sysconfig/tomcat6` file for Oracle Linux 6 and `/etc/sysconfig/tomcat` file for Oracle Linux 7. Specifically, edit the `JAVA_OPTS` environment variable and increase `-Xms` and `-Xmx`, which are the parameters, respectively, for the initial and the maximum amounts of memory.

2.6.2 Synchronizing Software Channels by Using the Spacewalk Web Interface

Figure 2.3 Channel Repositories Page



To synchronize software channels:

1. Go to **Channels**, select **Manage Software Channels** and then select the required channel.
2. On the Channel Details page, select **Repositories** and then select **Sync**.
3. On the Channel Repositories page:
   a. Select the following check boxes as required:

Do not sync erratas	Select if you do not want to synchronize any errata that are available for the channel.
Create kickstartable tree	Select if you want to be able to associate a kickstart profile with the channel.

Note

ULN and the Oracle Linux yum server do not host the boot image files that you require to create a kickstartable tree. Instead, you can obtain the files from a Oracle Linux Media Pack DVD image and make them available on a local file system. See Chapter 4, *Provisioning Client Systems*.

Terminate upon any error Select this option for synchronization to stop if an error occurs.

b. Synchronize the software channel:

- To perform an immediate manual synchronization, click **Sync Now**.
- To schedule a recurring synchronization job, select the preferred schedule and times, and then click **Schedule**.

  You can specify a schedule by using Quartz format. For example, `0 30 22 ? * *`, would specify that Spacewalk should resynchronize the channel every day at 10:30 PM. Using Quartz format is the only way to schedule a synchronization several times a day. For example, `0 0 0,2,22 ? * *`, would specify that synchronization should take place at 10 PM, midnight, and 2 AM.

2.6.3 Synchronizing Software Channels by Using the spacecmd Command

You can synchronize a software channel by using the `softwarechannel_syncrepos` command as follows:

```
spacecmd {SSM:0}> softwarechannel_syncrepos oraclelinux7-x86_64-ksplice
```

Note that the command returns immediately and does not show the status of the synchronization.

You can use the `tail -f` command to view the log file, `/var/logs/rhn/reposync/`*`channel_label`*`.log`.

Set up a schedule for channel synchronization by using the `softwarechannel_setsyncschedule` command as follows:

```
spacecmd {SSM:0}> softwarechannel_setsyncschedule oraclelinux7-x86_64-ksplice 0 30 2 ? * *
```

The previous command configures the `oraclelinux7-x86_64-ksplice` channel to be resynchronized once every day at 2:30 AM. Specify the schedule in Quartz format. For more information, see http://www.quartz-scheduler.org/overview/ for more information.

List the scheduled channel synchronizations by using the `softwarechannel_listsyncschedule` command as follows:

```
spacecmd {SSM:0}> softwarechannel_listsyncschedule
  key  Channel Name                          Update Schedule
-----  --------------------                  ---------------
  114  oraclelinux7-x86_64                   0 0 1 ? * *
  115  oraclelinux7-x86_64-addons
  177  oraclelinux7-x86_64-ksplice           0 30 2 ? * *
  120  oraclelinux7-x86_64-optional
  127  oraclelinux7-x86_64-patch             0 0 3 ? * *
```

```
  123  oraclelinux7-x86_64-spacewalk27-client   0 0 4 ? * *
  124  oraclelinux7-x86_64-spacewalk27-server   0 30 4 ? * *
  125  oraclelinux7-x86_64-uek-r3               0 0 1 ? * *
  126  oraclelinux7-x86_64-uek-r4               0 30 0 ? * *
```

To remove a scheduled channel synchronization, use the `softwarechannel_removesyncschedule` command as follows:

```
spacecmd {SSM:0}> softwarechannel_removesyncschedule oraclelinux7-x86_64-uek-r3
```

2.6.4 Synchronizing Software Channels by Using the spacewalk-repo-sync Command

You can use the `spacewalk-repo-sync` command to synchronize software channels. Using this command requires that you are the `root` user or that you have been granted permission in the `/etc/sudoers` file.

You can run the `spacewalk-repo-sync` command manually or in a `cron` job. If you run the command in a `cron` job, include the `-q` or `--quiet` options to prevent large email messages from being sent to `root`.

You can use the `spacewalk-repo-sync -l` command to display the channel label and the URL of the repository, as shown in the following example

```
# spacewalk-repo-sync -l | grep ksplice
ksplice-ol7-x86_64 | uln:///ol7_x86_64_ksplice
ksplice-ol6-i386 | uln:///ol6_i386_ksplice
ksplice-ol6-x86_64 | uln:///ol6_x86_64_ksplice

# spacewalk-repo-sync -l | grep addons
oraclelinux7-x86_64-addons | https://yum.oracle.com/repo/OracleLinux/OL7/addons/x86_64/
oraclelinux6-x86_64-addons | https://yum.oracle.com/repo/OracleLinux/OL6/addons/x86_64/
```

To synchronize a channel with either a Yum or a ULN repository, use the `-c` option to specify the channel label, as shown in the following example for Oracle Linux 6:

```
# spacewalk-repo-sync -c ksplice-ol6-x86_64
#### Channel label: ksplice-ol6-x86_64 ####
Repo URL: uln:///ol6_x86_64_ksplice
The download URL is: https://linux-update.oracle.com/XMLRPC/GET-REQ/ol6_x86_64_ksplice
Packages in repo:              1296
Packages already synced:          0
Packages to sync:              1296
1/1296 : ksplice-snmp-plugin-0.1.0-2.el6-0.x86_64
2/1296 : uptrack-updates-2.6.39-400.210.2.el6uek.x86_64-20150206-0-0.noarch
...
1295/1296 : uptrack-updates-2.6.32-220.el6.x86_64-20150130-0-0.noarch
1296/1296 : uptrack-updates-2.6.32-200.20.1.el6uek.x86_64-20141216-0-0.noarch
Linking packages to channel.
Repo uln:///ol6_x86_64_ksplice has 0 errata.
Sync completed.
Total time: 1 day, 8:56:47
```

In the previous example, all of the packages that were downloaded as the channel had not previously been synchronized with ULN. The total time taken was approximately 33 hours.

You can use the same command to synchronize a channel with an Oracle Linux yum server repository, as shown in the following example:

```
# spacewalk-repo-sync -c oraclelinux6-x86_64-addons
#### Channel label: oraclelinux6-x86_64-addons ####
Repo URL: https://yum.oracle.com/repo/OracleLinux/OL6/addons/x86_64/
Packages in repo:               308
No new packages to sync.
```

```
Repo https://yum.oracle.com/repo/OracleLinux/OL6/addons/x86_64/ has 6 errata.
Sync completed.
Total time: 0:01:09
```

In the previous example, no new packages were available for download.

Use the `-p` *`parent_channel`* option and argument to synchronize a parent channel and all of its children in one operation, as shown in this example:

```
# spacewalk-repo-sync -p oraclelinux7-x86_64
```

If you also specify the `--latest` option, as shown in the following example, the server synchronizes only the latest packages that are available:

```
 spacewalk-repo-sync -p oraclelinux7-x86_64 --latest
#### Channel label: oraclelinux7-x86_64 ####
Repo URL: https://yum.oracle.com/repo/OracleLinux/OL7/optional/latest/x86_64/
Packages in repo:             10133
Packages already synced:          0
Packages to sync:              5845
1/5845 : bind-lite-devel-9.9.4-18.el7_1.3-32.i686
2/5845 : bind-sdb-chroot-9.9.4-18.el7_1.2-32.x86_64
...
```

In the previous example, only the 5,845 latest packages (of the 10,133 total packages in the repository) needed to be downloaded.

Note

The `--latest` option downloads the latest packages that are available at the time of synchronization. Running the command does not remove older packages from the channel. If the synchronization interval is large, you might miss a particular version of a package, which 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* use `--latest`. Note that the exception is using `--latest` with a Ksplice channel because these packages are always cumulative.

For more information, see the `spacewalk-repo-sync(8)` manual page.

Caution

To reduce the overall space consumed by Ksplice Offline packages, Oracle strongly recommends using repository filters to limit downloaded packages to only those required by your client systems. You can apply a filter either on the Repository configuration in the Manage Repositories page in the web UI or by providing the `-i` or `--include` parameter with the `spacewalk-repo-sync` command line, for example:

```
# spacewalk-repo-sync --channel ol6_x86_64_ksplice -t uln \
  -i uptrack-updates-installed kernel base version,...
```

2.7 Cloning Software Channels

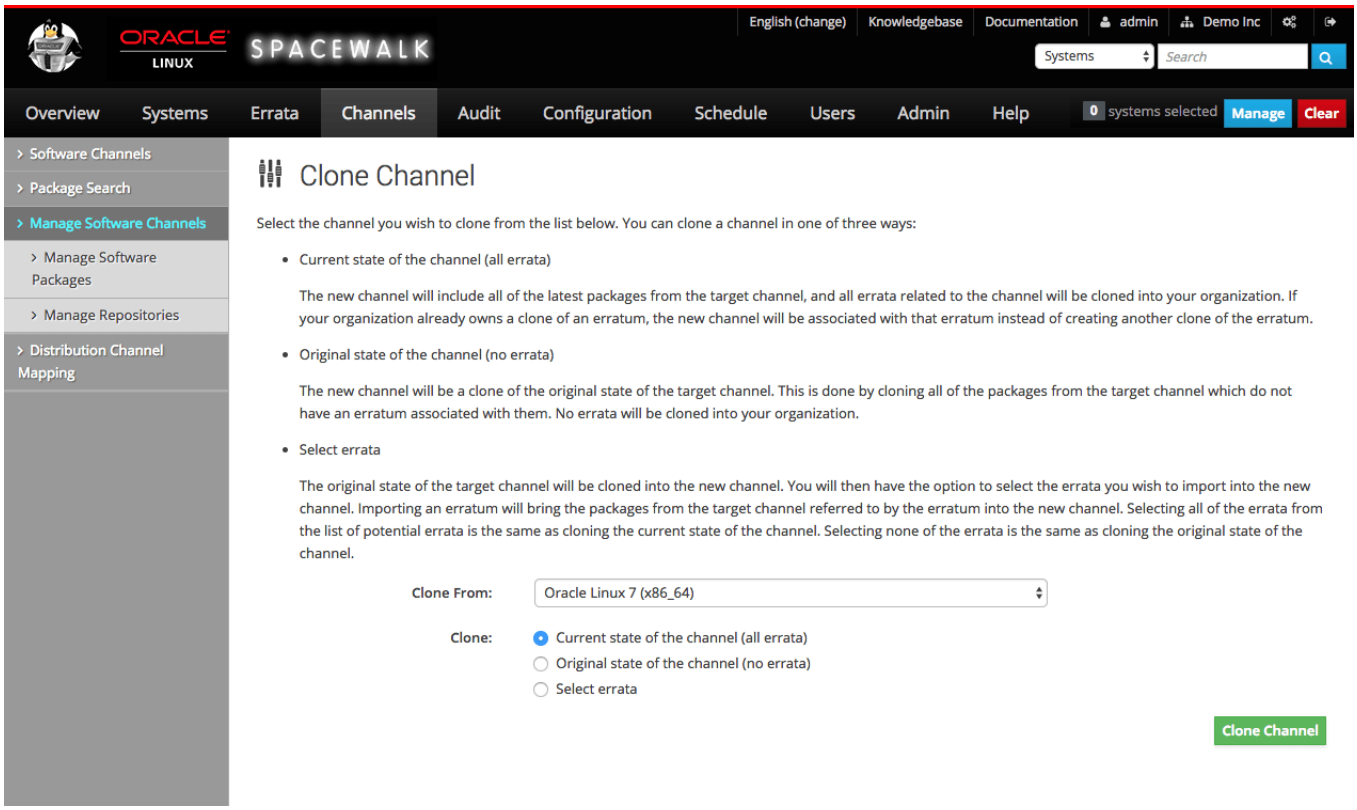
You can clone a software channel to capture the state of its packages and errata at a given point. Cloned channels are useful for providing a stable reference base when developing and testing server systems before deployment. Cloning channels are not recommended for deployed systems, as they might be exposed to security vulnerabilities.

Note the following addiitional information about cloning software channels:

- You can use the Spacewalk web interface to clone one channel at a time.
- If you want to clone a channel to preserve its state on a given date, use the `spacewalk-clone-by-date` command.
- If you want to clone a base channel and all of its child channels at the same time, consider using the `spacecmd` or `spacewalk-clone-by-date` command.
- The `spacewalk-manage-channel-lifecycle` command enables you to manage the life cycles of software channels from development, through testing to production. Oracle recommends using `spacewalk-manage-channel-lifecycle` in preference to `spacewalk-clone-by-date` as it supports archiving, roll back, and is designed for repetitive use. See Section 2.8, “Managing Channel Life Cycles by Using the spacewalk-manage-channel-lifecycle Command”.

2.7.1 Cloning Software Channels by Using the Spacewalk Web Interface

Figure 2.4 Clone Channel Page



Clone a software channel as follows:

1. Go to **Channels** and select **Manage Software Channels**.
2. Click **clone channel**.
3. On the Clone Channel page, select the source channel that you want to clone from the pull-down menu, and select the clone type:

   Current state of the channel (all errata) The clone channel includes all packages and errata from the source channel.

Original state of the channel (no errata)	The cloned channel includes all of the packages that were originally in the source channel, but no associated errata.
Select errata	The cloned channel includes all of the packages that were originally in the source channel and any errata that you select. Selecting all errata is equivalent to cloning the current state of the channel. Selecting no errata is equivalent to cloning the original state of the channel.

4. Click **Create Channel**.
5. On the Edit Software Channel page, you can change the channel details. The default label is the source channel label prefixed with `clone-`.
6. Click **Create Channel** again.
7. If you specified **Select errata** as the clone type, the Clone Errata page displays the available errata. For each erratum, you can choose to merge it with the source erratum, to create a separate cloned erratum, or to do nothing and exclude the erratum. By default, an erratum is merged with the source erratum, which means that the source erratum is used instead of creating a cloned copy.

   Click **Clone Errata** when you have finished cloning errata.
8. On the Details page for the channel, you can also edit the channel details other than the channel label.

   If you select the **Errata** tab, you can add errata from other channels or clone errata from the source channel.

   If you select the **Packages** tab, you can add or remove packages from the channel.

2.7.2 Cloning Software Channels by Using the spacecmd Command

You can clone a single channel by using the `softwarechannel_clone` command by running the following command:

```
spacecmd {SSM:0}> softwarechannel_clone -s ol6u9-x86_64 -x "s/$/-clone/" -o
spacecmd {SSM:0}> softwarechannel_details ol6u9-x86_64-clone
```

The `-x` option appends `-clone` to the new channel's name and label. The `-o` option excludes all errata from the cloned channel.

To compare the differences between the package contents of two channels, use the `softwarechannel_diff` command as follows:

```
spacecmd {SSM:0}> softwarechannel_diff ol6u9-x86_64-clone ol6u9-x86_64
```

To clone a base channel and all of its child channels, use the `softwarechannel_clonetree` command as follows:

```
spacecmd {SSM:0}> softwarechannel_clonetree -s ol6u9-x86_64 -p "clone-"
```

2.7.3 Cloning Software Channels by Using the spacewalk-clone-by-date Command

You use the `spacewalk-clone-by-date` command to clone Oracle Linux channels for a given date, which preserves the state of the channel's errata and their associated packages from its original release

up to and including that date. If required, you can blocklist or remove packages, and choose which types of errata to include or exclude.

For example, the following command clones only security errata from the `ol6-x86_64-latest` channel up to January 31, 2017 to `ol6-x86_64-latest-sec-20170131`:

```
# spacewalk-clone-by-date --username=swadmin --password=swpasswd \
  --to_date=2017-01-31 --channels=ol6-x86_64-latest ol6-x86_64-latest-sec-20170131 \
  --security_only --background --assumeyes
```

The command runs uninterrupted in the background. The specified spacewalk user must have Organizational Administrator or Channel Administrator privileges.

The following example clones both a base channel and a patch child channel up to January 31, 2017, excluding all versions of the `ntp` package and packages that start with `fuse`:

```
# spacewalk-clone-by-date --username=swadmin --password=swpasswd \
  --channels=ol6-x86_64-base ol6-x86_64-base-20170131
  --channels=ol6-x86_64-patch ol6-x86_64-patch-20170131
  --to_date=2017-01-31 --blacklist=ntp,fuse*
```

You can run `spacewalk-clone-by-date` remotely by using the `-s` option to specify the Spacewalk server's URL for XML/RPC API connections, for example `-s https://swksvr_FQDN/rpc/api`.

A common use case is to run `spacewalk-clone-by-date` at regular intervals to keep cloned channels up to date. To generate a sample configuration file, use the following command:

```
# spacewalk-clone-by-date --sample-config
```

For more information, see the `spacewalk-clone-by-date(8)` manual page.

2.8 Managing Channel Life Cycles by Using the spacewalk-manage-channel-lifecycle Command

The `spacewalk-manage-channel-lifecycle` command enables you to perform the following tasks to manage the life cycle of a software channel from development, through testing to production:

- Create a development channel `dev-ol6-x86_64-appsvr` based on the latest available packages in `ol6-x86_64-appsvr`.

  ```
  # spacewalk-manage-channel-lifecycle -c ol6-x86_64-appsvr --init
  ```

- Promote the packages from the development channel to the test channel `test-ol6-x86_64-appsvr`.

  ```
  # spacewalk-manage-channel-lifecycle -c dev-ol6-x86_64-appsvr --promote
  ```

- Promote the packages from the test channel to the production channel `prod-ol6-x86_64-appsvr`.

  ```
  # spacewalk-manage-channel-lifecycle -c test-ol6-x86_64-appsvr --promote
  ```

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

```
# spacewalk-manage-channel-lifecycle -c prod-ol6-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:

```
# spacewalk-manage-channel-lifecycle -c archive-20110520-test-ol6-x86_64-appsvr --rollback
```

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

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

 1. archive-20160203-ol6-x86_64-appsvr
      \__ archive-20160203-prod-ol6-x86_64-appcmd
      \__ archive-20160203-prod-ol6-x86_64-applib

 2. dev-ol6-x86_64-appsvr
      \__ dev-ol6-x86_64-appcmd
      \__ dev-ol6-x86_64-applib

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

 4. prod-ol6-x86_64-appsvr
      \__ prod-ol6-x86_64-appcmd
      \__ prod-ol6-x86_64-applib

 5. test-ol6-x86_64-appsvr
      \__ test-ol6-x86_64-appcmd
      \__ test-ol6-x86_64-applib
```

Chapter 3 Working With Activation Keys in Spacewalk

This chapter describes how to create and manage activation keys in Spacewalk.

3.1 About Spacewalk Activation Keys

An *activation key* enables a client system to register with Spacewalk without having to provide a user name and password. After you have configured and synchronized a base software channel and any child channels, create an activation key so that client systems can register with Spacewalk.

Note

If you use the `spacewalk-common-channels` command with the `-k` option to set up Oracle Linux software channels and repositories, the command also creates an activation key and associates this key with the software channels.

If required, you can also use an activation key to define the default parent and child software channel subscriptions and any configuration channels. Spacewalk subscribes a client to these channels during registration. However, you can change the channels at any later time without changing the activation key.

Note

Take care not to oversubscribe client systems to channels. Oracle recommends that you configure activation keys to subscribe a client to a minimal number of channels.

If required, you can create an activation key for each combination of base channel, system architecture, and server type. For example, you could create separate activation keys for web, mail, or application servers running on Oracle Linux 6 (i386), Oracle Linux 6 (x86_64), and Oracle Linux 7 (x86_64). Alternatively, you can create a single, default activation key without any channel assignments and then use it for all server types.

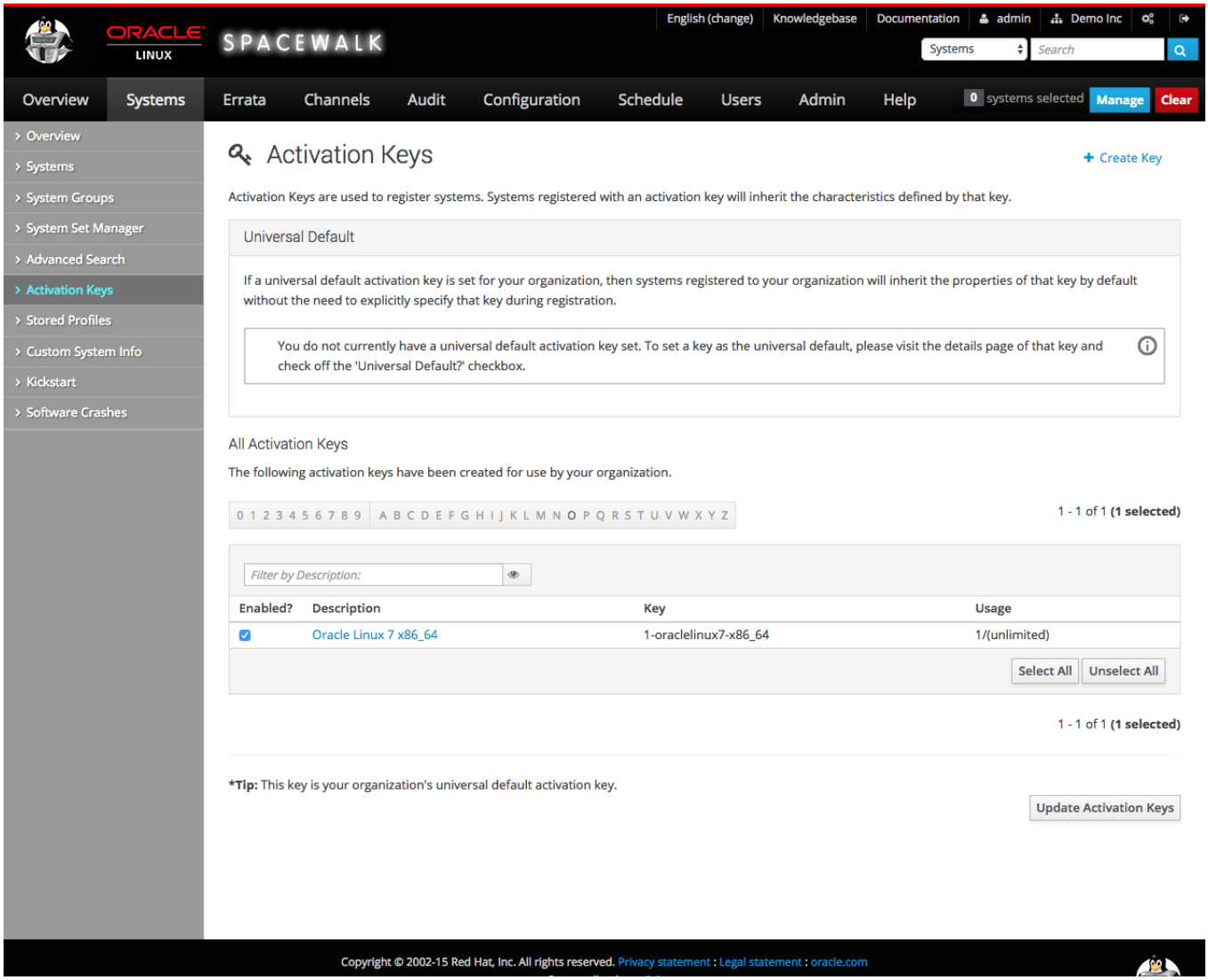
Oracle recommends that you enter a meaningful label for the activation key in the **Key** field and that you do not use automatic key generation.

Create a key with a label that is easy to understand, for example, based on the version number and architecture (`oraclelinux6-x86_64`), or based on the server type (`webserver` or `appserver`).

Spacewalk automatically prefixes the organization ID to the activation key label. For example, if you select `oraclelinux-x86_64` as the label, Spacewalk creates a key named `1-oraclelinux-x86_64`, where the prefix identifies the organization. You can create multiple activation keys for the same base channel, each with different configuration options. The name that you use is presented during Spacewalk client registration. Creating your own key labels helps you to select the correct key.

3.2 Working with Activation Keys by Using the Spacewalk Web Interface

Figure 3.1 Activation Keys Page



Go to **Systems** and select **Activation Keys**:

- To create an activation key:

  1. Click **+ create new key**.
  2. On the Create Activation Key page, enter the key settings in the following fields:

     Description — Enter a description for the key, for example, `Oracle Linux 6 (x86_64)`.

     Key — Enter a meaningful label for the activation key, for example, `oraclelinux6-x86_64`.

Usage — Leave this field blank to allow unlimited use by clients.

Base Channels — Select the base channel with which the key is associated. For example, `Oracle Linux 6 (x86_64) Base`.

Add-on Entitlements — If you are running the KVM hypervisor on Oracle Linux 6 or Oracle Linux 7, you should enable the **Virtualization** entitlement. This entitlement installs additional packages on the target host so that Spacewalk is able to identify and manage virtual guests on that host.

Note

Starting with Spacewalk 2.6, most entitlements have been removed, with the exception of the Virtualization entitlement.

Universal Default — Select if the key should be used as the default activation key for all newly-registered systems.

Note

Oracle strongly recommends that you do not associate any channels with a universal default key. Spacewalk uses the universal default key if a key is not specified so it might be used by any version of any operating system.

3. Click **Create Activation Key** to create the activation key.

- To view an activation key, select its entry to display its details.
- To modify an activation key:
  1. Select the activation key whose settings you want to edit.
  2. On the Activation Key Details page, modify the key settings.
  3. Click **Update Activation Key** to save your changes.
- To delete an activation key:
  1. Select the activation key that you want to delete.
  2. On the Activation Key Details page, click **delete key** and then click **Delete Activation Key** to confirm.

3.3 Working with Activation Keys by Using the spacecmd Command

To create an activation key, use the `activationkey_create` command as follows:

```
spacecmd {SSM:0}> activationkey_create
Name (blank to autogenerate): oraclelinux6-x86_64
Description [None]: Oracle Linux 6 (x86_64)

Base Channels
```

```
-------------
oraclelinux6-x86_64
oraclelinux7-x86_64

Base Channel (blank for default): oraclelinux6-x86_64

virtualization_host Entitlement [y/N]: N

Universal Default [y/N]: N
INFO: Created activation key 1-oraclelinux6-x86_64
```

To list all activation keys, use the `activationkey_list` command:

```
spacecmd {SSM:0}> activationkey_list
1-oraclelinux6-x86_64
1-oraclelinux7-x86_64
```

To display the details of an activation key, use the `activationkey_details` command:

```
spacecmd {SSM:0}> activationkey_details 1-oraclelinux7-x86_64
Key:                   1-oraclelinux7-x86_64
Description:           Oracle Linux 7 x86_64
Universal Default:     False
Usage Limit:           0
Deploy Config Channels: False

Software Channels
-----------------
oraclelinux7-x86_64
 |-- oraclelinux7-x86_64-addons
 |-- oraclelinux7-x86_64-mysql55
 |-- oraclelinux7-x86_64-mysql56
 |-- oraclelinux7-x86_64-mysql57
 |-- oraclelinux7-x86_64-optional
 |-- oraclelinux7-x86_64-spacewalk27-client
 |-- oraclelinux7-x86_64-uek-r4

Configuration Channels
----------------------

Entitlements
------------

System Groups
-------------

Packages
--------
```

To delete an activation key, use the `activationkey_delete` command:

```
spacecmd {SSM:0}> activationkey_delete 1-oraclelinux7-x86_64
1-oraclelinux7-x86_64

Delete activation key(s) [y/N]: y
```

Chapter 4 Provisioning Client Systems

Oracle supports the provisioning of Oracle Linux servers as Spacewalk client systems. You can use Spacewalk to manage Fedora-based clients and other systems by using upstream client binaries and repositories. However, note that Oracle does not provide support for these clients.

Oracle provides Spacewalk Client packages for Oracle Linux 6, for both x86_64 and x86_64 architectures. For Oracle Linux 7, only packages for the x86_64 architecture are provided.

If you configure a Spacewalk server to mirror the Spacewalk client channel that is provided on the Oracle Linux yum server and then enable this channel for a kickstart profile, Spacewalk automatically installs the Spacewalk Client software on any Oracle Linux server that it provisions and registers this server as a Spacewalk Client. You can use the `spacewalk-common-channels` command to configure the Spacewalk client channel, as described in Section 2.3, “Configuring Software Channels to Obtain Packages From the Oracle Linux Yum Server”.

4.1 About Kickstart Trees, Distributions, and Profiles

You can use kickstart to automate the installation of Oracle Linux systems and then use Spacewalk to provide packages during the installation.

If you want to provision bare-metal systems and virtual machines, create a distribution in Spacewalk for each combination of Oracle Linux release and system architecture that you want to install by using kickstart.

You need to set up a local directory such as `/var/distro-trees` on your Spacewalk server that contains the entire kickstart tree for each distribution, which includes the installation kernel, the initial ram-disk image, installation files, and information about the repositories. This directory must be readable and accessible by the `httpd` and `tomcat6` services. The kickstart tree does not need to include any packages, as Spacewalk provides these.

Relative to the root of the kickstart tree, the installation kernel and initial ram-disk images should be located at `./images/pxeboot`. For example, if the root of the kickstart tree for Oracle Linux 6 (x86_64) server installation is `/var/distro-trees/ol6-x86_64-server`, the installation kernel and the initial ram-disk images would be located at `/var/distro-trees/ol6-x86_64-server/images/pxeboot`.

See Section 4.2, “Setting Up Kickstart Trees”.

You create a kickstart distribution by associating a kickstart tree with existing channels. A client boots by using the kickstart tree but installs its software packages from the existing channels. The packages that are installed on the client will be as up to date as those that are currently available from the channels.

See Section 4.3, “Working With Kickstart Distributions”.

Note

Currently, you cannot use the `spacewalk-repo-sync --sync-kickstart` command to create a kickstart distribution from the channels that are available on the Oracle Linux yum server or on ULN.

After you have created a kickstart distribution, you can use it with kickstart profiles. Typically, each profile provisions a different type of server. You can configure a profile to generate a kickstart file, or you can use an existing kickstart file. You can associate as many profiles with a single distribution as you need to provision servers that share the same combination of Oracle Linux release and system architecture.

See Section 4.4, “Working With Kickstart Profiles”.

4.2 Setting Up Kickstart Trees

To set up a kickstart tree for a distribution on the Spacewalk server, do the following:

1. If the root for all kickstart trees, typically, `/var/distro-trees`, does not already exist, create this directory. If required, set its SELinux file type as `httpd_sys_content_t` so that `httpd` and `tomcat6` can make the files available:

   a. Create the `root` directory for the kickstart tree as follows:

   ```
   # mkdir -p /var/distro-trees/ol7u2-x86_64-server
   ```

   b. If SELinux is enabled in enforcing mode on your system, do the following:

      i. Use the `semanage` command to define the default file type of the kickstart tree as `httpd_sys_content_t`.

      ```
      # /usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/var/distro-trees(/.*)?"
      ```

      ii. Use the `restorecon` command to apply the file type to the entire directory hierarchy.

      ```
      # /sbin/restorecon -R -v /var/distro-trees
      ```

      Note

      The `semanage` and `restorecon` commands are provided by the `policycoreutils-python` and `policycoreutils` packages.

2. Download the full Oracle Linux Media Pack DVD image for the Oracle Linux release and system architecture from the Oracle Software Delivery Cloud at https://edelivery.oracle.com/linux and mount it on a suitable mount point.

   Note

   For the mount to succeed, you will need to create the `/var/distro-trees/olNun-arch-server` directory if it does not already exist.

   ```
   # mount -o loop /var/ISOs/DVDimage.iso /var/distro-trees/olNun-arch-server
   ```

3. Create an entry in the `/etc/fstab` file so that the system always mounts the DVD image after a reboot.

   ```
   /var/ISOs/V100082-01.iso  /var/distro-trees/OL7u2-x86_64-server  iso9660  loop,ro  0 0
   ```

4. If you want to associate a kickstart tree with a software channel or to be able to boot iPXE clients, create a symbolic link from `/var/www/html` to `/var/distro-trees`.

   ```
   # ln -s /var/distro-trees /var/www/html/distro-trees
   ```

   The installation images will then be available at a URL such as `https://swksvr_FQDN/distro-trees/olNun-arch-server/images`, where *`swksvr_FQDN`* is the FQDN of the Spacewalk server or proxy.

   You should be able to see the contents of the mounted installation image listed at the URL by using a browser. If you cannot see the files, do the following:

a. Edit the `/etc/httpd/conf/httpd.conf` file and enable support for directory indexing and symbolic links by specifying `Options Indexes FollowSymLinks` in the `<Directory "/var/www/html">` section.

b. Reload the `httpd` service.

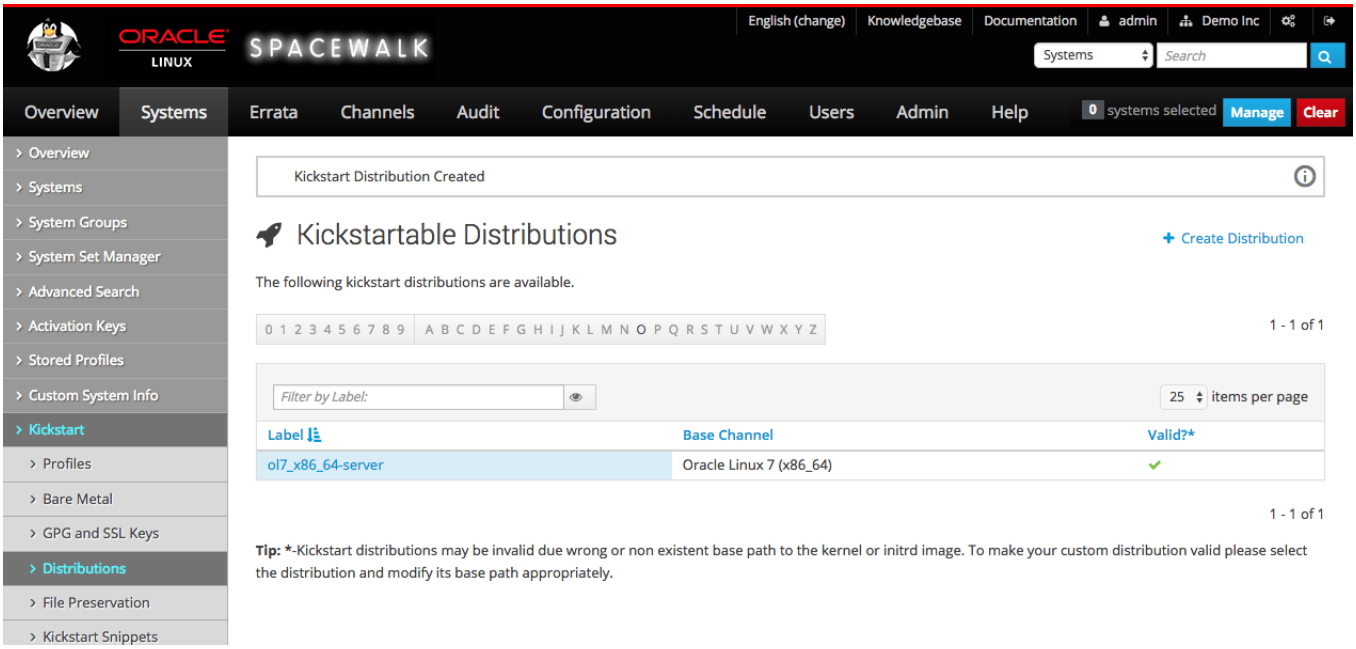
```
# service httpd reload
```

4.3 Working With Kickstart Distributions

If you want to create a kickstart distribution that uses a kickstart tree with existing software channels, you can use either the Spacewalk web interface or the Spacewalk CLI.

4.3.1 Working With Kickstart Distributions by Using the Spacewalk Web Interface

Figure 4.1 Kickstartable Distributions Page



Go to **Systems**, select **Kickstart** and then **Distributions**.

- To create a distribution:

1. Click **+ create new distribution**.
2. On the Create Activation Key page, enter the key settings in the following fields:

Distribution Label	Enter a label for the distribution, for example, `ol7u4-x86_64-server`.
Tree Path	Enter the path of the kickstart tree for the distribution, for example, `/var/distro-trees/ol7u2-x86_64-server`.
Base Channel	Select the base channel with which the distribution is associated, for example, `Oracle Linux 7 (x86_64) Base`.

Installer Generation	Select the operating system release that provided the installer, for example, `Red Hat Enterprise Linux 7/Oracle Linux 7`.
Kernel Options	Enter any options that should be specified when booting the installation kernel, for example, `noapic` or `text`.
Post Kernel Options	Enter any options that should be specified when booting the installed system's kernel, for example, `3` or `selinux=0`.

3. Click **Create Kickstart Distribution** to create the distribution.

- To view a distribution, select its entry to display its details.
- To modify a distribution:
  1. Select the distribution whose settings you want to edit.
  2. On the Edit Kickstart Distribution page, modify the settings as required.
  3. If you want to create, modify, or delete kickstart variables:
     a. Select the **Variables** tab.
     b. On the Kickstart Variables page, define new variables or edit or delete existing variable entries.
     c. Click **Update Variables** to save your changes.
     d. Select the Edit tab to return to the Edit Kickstart Distribution page.
  4. Click **Update Kickstart Distribution** to save your changes.
- To delete a distribution:
  1. Select the distribution that you want to delete.
  2. On the Edit Kickstart Distribution page, click **delete distribution** and then click **Delete Distribution** to confirm.

4.3.2 Working With Kickstart Distributions by Using the spacecmd Command

Create a distribution by using the `distribution_create` command, as shown in the following example:

```
spacecmd {SSM:0}> distribution_create
Name: ol7-x86_64-server
Path to Kickstart Tree: /var/distro-trees/ol7-x86_64-server

Base Channels
-------------
oraclelinux6-x86_64
oraclelinux7-x86_64

Base Channel: oraclelinux7-x86_64

Install Types
-------------
fedora
generic_rpm
rhel_2.1
rhel_3
rhel_4
```

```
rhel_5
rhel_6
rhel_7
suse

Install Type: rhel_7
```

To list all activation keys, use the `distribution_list` command:

```
spacecmd {SSM:0}> distribution_list
ol6-x86_64-server
ol7-x86_64-server
```

To display the details of an activation key, use the `distribution_details` command:

```
spacecmd {SSM:0}> distribution_details ol7-x86_64-server
Name:    ol7-x86_64-server
Path:    /var/distro-trees/ol7-x86_64-server
Channel: oraclelinux7-x86_64
```

To delete an activation key, use the `distribution_delete` command:

```
spacecmd {SSM:0}> distribution_delete ol7-x86_64-server
ol7-x86_64-server

Delete distribution tree(s) [y/N]: y
```

4.4 Working With Kickstart Profiles

A kickstart configuration file contains all of the information that kickstart requires to perform an automated installation of a server. Every Oracle Linux installation creates a kickstart file, which is `/root/anaconda-ks.cfg`. You can use this file to repeat an installation, or you can customize the settings in this file for different system configurations. The file is also useful for troubleshooting a boot-time problem with an installed system. You can use Spacewalk to create a kickstart profile that generates a kickstart file or you can create a profile that contains a kickstart file that you have uploaded or copied into Spacewalk.

You can use either the Spacewalk web interface or the Spacewalk command line to configure kickstart profiles.

4.4.1 Adding GPG Keys and SSL Certificates by Using the Spacewalk Web Interface

Oracle Linux ships with the GPG key that is required for each version by default. However, you must manually add GPG keys for any third-party vendors.

The Oracle GPG key that is appropriate for the installed version is stored by default at `file:///etc/pki/rpm-gpg-RPM-GPG-KEY` on every Oracle Linux server.

Add a GPG key or SSL certificate to Spacewalk as follows:

1. Go to **Systems**, select **Kickstart** and then **GPG and SSL Keys** to display the GPG Public Keys and SSL Certificates page.
2. Click **Create Stored Key/Cert** to display the Create GPG/SSL Key page.
3. Enter a text description of the key or certificate in the Description field.
4. Select **GPG** or **SSL**, as appropriate, from the **Type** pull-down menu.
5. Either click **Browse** and select the key or certificate file to upload or paste the file contents into the Key contents field.

Note

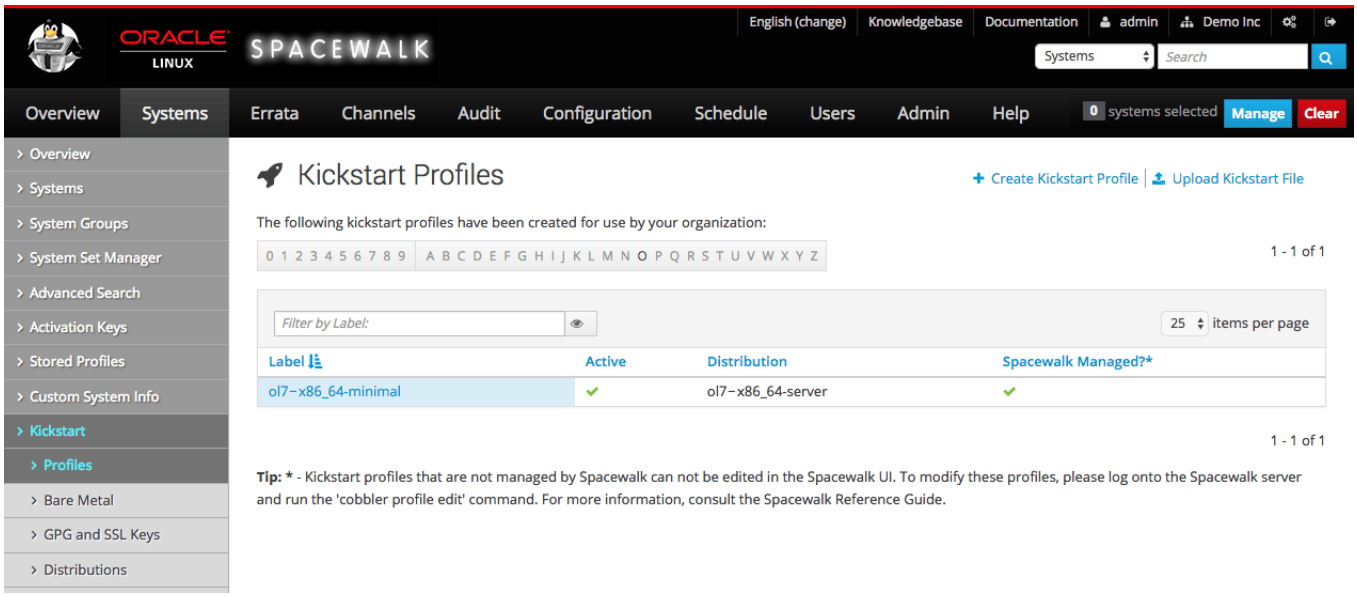
GPG keys must be in ASCII, not binary, format.

6. Click **Create Key**.

After you have added the GPG keys and SSL certificates to Spacewalk, you can associate them with kickstart profiles as described in Section 4.4.2, “Working With Kickstart Profiles by Using the Spacewalk Web Interface”.

4.4.2 Working With Kickstart Profiles by Using the Spacewalk Web Interface

Figure 4.2 Kickstart Profiles Page



Go to **Systems**, select **Kickstart** and then **Profiles**.

- To create a profile that contains a kickstart file generated by Spacewalk:

  1. Click **+ create new kickstart profile**.
  2. On Step 1, Create kickstart Profile page, enter the profile settings in the following fields:

     Label — Enter a label for the profile, for example, `ol7u4-x86_64-minimal`.

     Base Channel — Select the base channel with which the distribution is associated, for example, `Oracle Linux 7 (x86_64) Base`.

     Kickstartable Tree — Select the kickstart distribution with which the profile is associated, for example, `ol7u4-x86_64-server`.

     Virtualization Type — Select the virtualization type.

     For Oracle Linux installations on virtual machines that are hosted by Oracle VM or Oracle VM VirtualBox, select **None**.

For Oracle Linux 6 and Oracle Linux 7 as a Kernel-based Virtual Machine (KVM) guest, select **KVM Virtualized Guest**.

Click **Next**.

3. On Step 2, Distribution File Location page, click **Next** to accept the default download location that Spacewalk creates from the kickstart tree.

4. On Step 3, Root Password page, enter and verify the root password for newly installed systems, and click **Finish** to create the profile.

   You can now configure the kickstart profile itself. The following steps describe the changes that are usually required to create a useable profile.

   Note

   At any stage, you can select **Kickstart File** to view the kickstart file that Spacewalk would generate from the profile by using the saved configuration settings.

5. Select **Kickstart Details** to display the Details page.

   a. On the Details page, you can do the following:

      - Edit the Kickstart label.
      - Change the virtualization type.
      - Activate or de-activate the profile.
      - Configure custom post and pre script logging.
      - Choose whether to save a copy of the kickstart configuration to `/root` on an installed system.
      - Select an organization default profile.
      - Specify installation and post-installation kernel options.
      - Add a description of the profile.

      Click **Update Kickstart** to save your changes.

   b. Select **Operating System** and select the check boxes for the child channels that you want to associate with the profile.

      Note

      To enable Spacewalk to register the system automatically, select the Spacewalk Client channel.

      The **Software URL** path is the virtual location where Spacewalk hosts the installation packages. It is not a real path in the file system.

      Click **Update Kickstart** to save your changes.

   c. (Optional) Select **Variables** to define any kickstart variables that you require, then click **Update Variables** to save your changes.

d. Select **Advanced Options** to modify the kickstart options, then click **Update Kickstart** to save your changes.

   For more information about the available kickstart options, see Appendix A, *Kickstart Options*.

e. If you intend to install bare-metal systems, select **Bare Metal Kickstart** and follow the instructions on the Bare Metal Kickstart page. This page lists the URL of the kickstart file that you

can use to install bare-metal systems and enables you to define the IP address ranges that are associated with the profile.

6. Select **System Details** to display the Details page:

   a. On the Details page, you can do the following:

   - Choose the default SELinux mode for the installed system.
   - Enable or disable Spacewalk configuration file management by selecting or deselecting the **Enable Spacewalk Configuration Management** check box.

Note

You must also include the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages for installation.

   - Enable or disable Spacewalk remote commands by selecting or deselecting the **Enable Spacewalk Remote Commands** check box.

Note

You must also include the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages for installation.

   - Choose whether to reuse an existing profile, replace the existing profile, or create a new profile but retain the existing profile.
   - Change the root password for installed systems.

Note

If you make any other changes on this page, you must re-enter and verify the root password.

   Click **Update System Details** to save your changes.

   b. Select **Locale**, then select the default time zone for installed systems and whether the hardware clock uses UTC, and click **Update Locale Preferences** to save your changes.

   c. Select **Partitioning** to define the partitions to be created during installation, then click **Update Partitions** to save your changes.

Note

Clear the partitioning configuration if you select the automatic-partitioning option `autopart` on the Advanced Options page.

   d. Select **GPG & SSL** to display a list of the GPG keys and SSL certificates that are known to Spacewalk. Select the keys and certificates that should be imported into the `%post` section of the kickstart profile, then click **Update Keys** to save your selection.

   For information about adding a GPG key or SSL certificate to Spacewalk, see Section 4.4.1, “Adding GPG Keys and SSL Certificates by Using the Spacewalk Web Interface”.

7. Select **Software** to display the Package Groups page.

a. Edit the list of packages to be installed:

- For sample lists of packages, see Appendix B, *Sample Minimum Package Lists*.
- The `@ Base` entry installs a minimal group of packages that are required to install a system.

  If you want to specify the list of base packages explicitly, select the **Don't install @Base package group** check box.
- If you do not want the installation to halt if it cannot locate a package, select the **Ignore missing packages** check box.
- If you have associated the Spacewalk Client channel with the profile, Spacewalk installs the Spacewalk Client packages automatically. You do not need to specify them in this list.
- If you enable configuration file management and remote commands by selecting the **Enable Spacewalk Configuration Management** and **Enable Spacewalk Remote Commands** check boxes on the Details page, include the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages.
- If you want to be able to apply updates and actions to a client system immediately from the Spacewalk server, include the `osad` package.

b. Click **Update Packages** to save your changes.

8. Select **Activation Keys**, then select the activation key to associate with the profile, and click **Update Activation Keys** to save your changes.

Note

A Spacewalk server activates the channels that are associated with an activation key when it registers the Spacewalk client at the end of the provisioning process. Enabling the Spacewalk Client channel by specifying the activation key is not sufficient to install the Spacewalk client software during the kickstart process. Instead, you must specify the packages in the kickstart profile. The channels that are available to a Spacewalk client during a kickstart installation and the channels that are available after installation are

independent. You can use channels during a kickstart installation that are not available after installation if the activation key does not enable them.

9. Select **Scripts** to define commands that you want to run on the system before or after installation.

   You can configure a preinstallation or post-installation script by using the following fields:

   Scripting Language — (Optional) The path name of the script language interpreter, such as `/usr/bin/python`.

   Leave blank if you want to run `bash` shell commands.

   Script Name — Enter a name for the script.

   Script Contents — Select the script type from the pull-down list: **Shell**, **XML**, **Ruby**, **Python**, or **perl**, and enter the script in the text area.

   Script Execution Time — Select the time at which the script is executed from the pull-down list: **Pre Script** for before installation or **Post Script** for after installation.

   nochroot — (Optional) Select whether the script should run outside a `chroot` jail.

   erroronfail — (Optional) Select to stop the installation if an error occurs when the script runs.

   Template — (Optional) Select to enable Cobbler templating for the script.

Note

If you want to be able to apply updates and actions to a client system immediately from the Spacewalk server, include the `osad` package for installation, which contains the Open Source Architecture (OSA) daemon and use the following kickstart option to enable the `osad` service:

```
services --enabled=osad
```

If you do not enable configuration file management and remote commands by selecting the **Enable Spacewalk Configuration Management** and **Enable Spacewalk Remote Commands** check boxes on the Details page, you can alternatively include the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages for installation and configure `rhn-actions-control` to run on the client system in the post-installation shell:

```
rhn-actions-control --enable-all
```

For more information, see the `rhn-actions-control(8)` manual page.

- To create a profile that contains a kickstart file that you upload or copy into Spacewalk, do the following:

  1. Click **upload new kickstart file**.
  2. On the Kickstart Details page, enter the key settings in the following fields:

     Label — Enter a label for the profile, for example, `ol6-x86_64-custom`.

     Kickstartable Tree — Select the kickstart distribution with which the profile is associated, for example. `ol6-x86_64-server`.

     Virtualization Type — Select the virtualization type. For Oracle Linux installations on virtual machines that are hosted by Oracle VM or Oracle VM VirtualBox, select **None**.

  3. Do one of the following:

     - Copy and paste the contents of a kickstart file into the **File Contents** text box.
     - Click **Browse...**, then select the path of a kickstart file and click **Upload file** to upload it to the **File Contents** text box.

  4. If necessary, edit the kickstart file contents in the **File Contents** text box.
  5. Click **Update**.

- To view a profile, select its entry to display its details.
- To modify a profile, do the following:

  1. Select the profile for which you want to modify settings.
  2. Select each tab and page that contains settings that you want to modify.
  3. Click the Confirmation button on each page to save your changes.

- To delete a profile, do the following:

  1. Select the distribution that you want to delete.
  2. On the Kickstart Details page, click **delete kickstart** and then click **Delete Kickstart** to confirm.

4.4.3 Working With Kickstart Profiles by Using the spacecmd Command

You can list all of the kickstart profiles by using the `kickstart_list` command as follows:

```
spacecmd {SSM:0}> kickstart_list
ol6u6-x86_64-minimal
```

To display the details of a kickstart profile, use the `kickstart_details` command:

```
spacecmd {SSM:0}> kickstart_details ol6u6-x86_64-minimal
Name:        ol6u6-x86_64-minimal
Label:       ol6u6-x86_64-minimal
Tree:        ol6-x86_64-server
Active:      True
Advanced:    False
Org Default: False
```

```
Configuration Management: False
Remote Commands:          False

Software Channels
-----------------
ol6u6-x86_64

Advanced Options
----------------
auth --enableshadow --passalgo=sha256
bootloader --location mbr
clearpart --all
firewall --disabled
keyboard us
lang en_US
network --bootproto dhcp
rootpw $5$ZdYXHxbNqu76Q5dG$.KWiOPyrGk8V5q/FEqYbWpCZdD5St387sn7jOyPH400
selinux --permissive
timezone America/New_York
url --url /var/distro-trees/ol6-x86_64-server

Software
--------
@ Base

Crypto Keys
-----------
RHN-ORG-TRUSTED-SSL-CERT

Variables
---------
org = 1
```

To display the contents of the kickstart file that a profile generates, use the `kickstart_getcontents` command:

```
spacecmd {SSM:0}> kickstart_getcontents ol6u6-x86_64-minimal
# Kickstart config file generated by Spacewalk Config Management
# Profile Label : ol6u6-x86_64-minimal
# Date Created  : 2015-06-11 11:34:15.157666

install
text
network --bootproto dhcp
url --url http://swksvr.mydom.com/ks/dist/ol6-x86_64-server
lang en_US
keyboard us
zerombr
clearpart --all
bootloader --location mbr
timezone America/New_York
auth --enableshadow --passalgo=sha256
rootpw --iscrypted $5$ZdYXHxbNqu76Q5dG$.KWiOPyrGk8V5q/FEqYbWpCZdD5St387sn7jOyPH400
selinux --permissive
reboot
firewall --disabled
skipx
autopart
...
```

The `spacecmd` command provides a large number of commands for managing kickstart profiles. Use the `help` command to find out more information:

```
spacecmd {SSM:0}> help

Documented commands (type help <topic>):
```

```
========================================
...
kickstart_addactivationkeys
kickstart_addchildchannels
kickstart_addcryptokeys
kickstart_addfilepreservations
kickstart_addoption
kickstart_addpackages
kickstart_addscript
kickstart_addvariable
kickstart_clone
kickstart_create
kickstart_delete
kickstart_details
kickstart_diff
kickstart_disableconfigmanagement
kickstart_disableremotecommands
kickstart_enableconfigmanagement
kickstart_enablelogging
kickstart_enableremotecommands
kickstart_export
kickstart_getcontents
kickstart_getupdatetype
kickstart_import
kickstart_import_raw
kickstart_importjson
kickstart_list
kickstart_listactivationkeys
kickstart_listchildchannels
kickstart_listcryptokeys
kickstart_listcustomoptions
kickstart_listoptions
kickstart_listpackages
kickstart_listscripts
kickstart_listvariables
kickstart_removeactivationkeys
kickstart_removechildchannels
kickstart_removecryptokeys
kickstart_removefilepreservations
kickstart_removeoptions
kickstart_removepackages
kickstart_removescript
kickstart_removevariables
kickstart_rename
kickstart_setcustomoptions
kickstart_setdistribution
kickstart_setlocale
kickstart_setpartitions
kickstart_setselinux
kickstart_setupdatetype
kickstart_updatevariable
...
spacecmd {SSM:0}> help kickstart_create
kickstart_create: Create a Kickstart profile
usage: kickstart_create [options]

options:
  -n NAME
  -d DISTRIBUTION
  -p ROOT_PASSWORD
  -v VIRT_TYPE ['none', 'para_host', 'qemu', 'xenfv', 'xenpv']
```

4.5 Installing Client Systems by Using Kickstart

To install a client system from a generated kickstart file, you can do one of the following:

- Boot the system from a real or virtual CD-ROM drive by using a boot ISO image or a full DVD image that you have downloaded from the Oracle Software Delivery Cloud at https://edelivery.oracle.com/linux, specifying the network location of the kickstart file as a boot option.

  This installation method is suitable for installing virtual machines or if you need to install only a small number of bare-metal systems at a local site.

- After configuring DHCP to support network booting of PXE clients and Cobbler to support the requirements of individual clients, boot the system from the network.

  This installation method is suitable for installing virtual machines or if you need to install bare-metal systems at both local and remote sites.

4.5.1 Configuring Cobbler and DHCP to Support Network Booting

The following procedure assumes that you have configured a DHCP server on the same system as the Spacewalk server. If DHCP is already provided on your network, you will need to work with your Network Administrator to configure network (PXE) boot support.

To configure Cobbler and DHCP to support the booting of client systems over the network, follow these steps:

1. Install the `cobbler-loaders` and `dhcp` packages.

   ```
   # yum install cobbler-loaders dhcp
   ```

2. To configure Cobbler to manage the DHCP service, edit `/etc/cobbler/settings` and modify the `manage_dhcp` setting.

   ```
   manage_dhcp: 1
   ```

3. Edit the DHCP server configuration template file (`/etc/cobbler/dhcp.template`) and change the subnet configuration for your local configuration.

   The following example show how to select either the pxelinux boot loader for BIOS-based PXE clients or the GRUB boot loader for UEFI-based PXE clients:

   ```
   # ******************************************************************
   # Cobbler managed dhcpd.conf file
   #
   # generated from cobbler dhcp.conf template ($date)
   # Do NOT make changes to /etc/dhcpd.conf. Instead, make your changes
   # in /etc/cobbler/dhcp.template, as /etc/dhcpd.conf will be
   # overwritten.
   #
   # ******************************************************************

   ddns-update-style interim;

   allow booting;
   allow bootp;

   ignore client-updates;
   set vendorclass = option vendor-class-identifier;
   option pxe-system-type code 93 = unsigned integer 16;
   set pxetype = option pxe-system-type;

   option domain-name "mydom.com";

   subnet 192.168.1..0 netmask 255.255.255.0 {
     option domain-name-servers 192.168.1.1;
     option broadcast-address 192.168.1.255;
   ```

```
  option routers 192.168.1.254;
  default-lease-time 14400;
  max-lease-time 28800;
  pool {
    range 192.168.1.101 192.168.1.200;
  }
}

#for dhcp_tag in $dhcp_tags.keys():
    ## group could be subnet if your dhcp tags line up with your subnets
    ## or really any valid dhcpd.conf construct ... if you only use the
    ## default dhcp tag in cobbler, the group block can be deleted for a
    ## flat configuration
# group for Cobbler DHCP tag: $dhcp_tag
group {
        #for mac in $dhcp_tags[$dhcp_tag].keys():
            #set iface = $dhcp_tags[$dhcp_tag][$mac]
    host $iface.name {
        hardware ethernet $mac;
        #if $iface.ip_address:
        fixed-address $iface.ip_address;
        #end if
        #if $iface.hostname:
        option host-name "$iface.hostname";
        #end if
        #if $iface.netmask:
        option subnet-mask $iface.netmask;
        #end if
        #if $iface.gateway:
        option routers $iface.gateway;
        #end if
        if substring(vendorclass, 0, 9)="PXEClient" {
          if pxetype=00:06 or pxetype=00:07 {
              filename "/grub/grub.efi";
          } else {
              filename "/pxelinux.0";
          }
        }
        ## Cobbler defaults to $next_server, but some users
        ## may like to use $iface.system.server for proxied setups
        next-server $next_server;
        ## next-server $iface.next_server;
    }
        #end for
}
#end for
```

The previous example also configures a pool of generally available IP addresses in the range `192.168.1.101` through `192.168.1.200` on the `192.168.1/24` subnet. Systems in this pool do not boot by using PXE.

All comments or commented-out DHCP directives in `/etc/cobbler/dhcp.template` are preceded by a double hash (`##`) to prevent Cobbler from interpreting them.

Spacewalk configures Cobbler to use TFTP to serve the boot-loader configuration files from the `/var/lib/tftpboot` directory. For more information about the format of these files, see Section 4.5.3, “About Boot Loader Configuration Files”.

If you want DHCP to support network booting of iPXE clients, see Section 4.5.4, “Configuring DHCP to Support iPXE Clients”.

4. If SELinux is enabled in `enforcing` mode on your system, configure SELinux for Cobbler operation as follows:

   a. Permit the `httpd` service to act as a proxy for Cobbler.

   ```
   # setsebool -P httpd_can_network_connect=1
   ```

   b. Set the `public_content_t` file type on the `/var/lib/tftpboot` file and `/var/www/cobbler/images` directory hierarchies as follows:

   ```
   # /usr/sbin/semanage fcontext -a -t public_content_t "/var/lib/tftpboot/.*"
   # /usr/sbin/semanage fcontext -a -t public_content_t "/var/www/cobbler/images/.*"
   ```

   Note

   The `semanage` command is provided by the `policycoreutils-python` package.

5. Restart the `cobblerd` service:

   ```
   # service cobblerd restart
   ```

6. Start the `httpd` service and configure it to start after a reboot.

   ```
   # service httpd start
   # chkconfig httpd on
   ```

   Note

   If you make any changes to `/etc/cobbler/dhcp.template`, run the `cobbler sync` command.

   If you make any changes to `/etc/cobbler/settings`, restart the `cobblerd` service and then run the `cobbler sync` command.

7. To support booting of UEFI-based PXE clients, copy the `/boot/efi/EFI/redhat/grub.efi` file to `/var/lib/tftpboot/grub`.

   ```
   # cp /boot/efi/EFI/redhat/grub.efi /var/lib/tftpboot/grub
   ```

8. Configure the firewall to allow access through DHCP requests.

   For Oracle Linux 6:

   ```
   # iptables -I INPUT -i eth1 -p udp --dport 67:68 --sport 67:68 -j ACCEPT
   # service iptables save
   ```

   In the previous example, the server expects to receive requests on interface `eth1`.

   For Oracle Linux 7:

   ```
   # firewall-cmd --permanent --zone=public --remove-interface=enp0s3
   # firewall-cmd --permanent --zone=internal --add-interface=enp0s3
   # firewall-cmd --permanent --zone=internal --add-port=67/udp
   # firewall-cmd --permanent --zone=internal --add-port=68/udp
   # firewall-cmd --reload
   ```

   In the previous example, the server expects to receive requests on interface `enp0s3` in the `internal` zone.

4.5.2 Adding a PXE Client To Be Provisioned by Spacewalk

Add a PXE client to be provisioned by Spacewalk as follows:

1. List the kickstart profiles in Spacewalk that are usable by Cobbler.

```
# cobbler profile list
   ol6u6-x86_64-devsys:1:SpacewalkDefaultOrganization
   ol6u6-x86_64-server:1:SpacewalkDefaultOrganization
```

2. Use the `cobbler system add` command to define the host name, MAC address, and IP address of the target PXE client and the profile that you want to install, as shown in the following example:

```
# cobbler system add --name=svr1.mydom.com --hostname=svr1.mydom.com --mac=08:00:27:c6:a1:16 \
  --ip=92.168.1.253 --profile=ol6u6-x86_64-server:1:SpacewalkDefaultOrganization
```

   If you are provisioning a client that uses an IP address from a DHCP address pool, you might use a command such as the following:

```
# cobbler system add --name=devsys2 --hostname=devsys2 \
  --profile=ol6u6-x86_64-devsys:1:SpacewalkDefaultOrganization \
  --kopts="ksdevice=eth0"
```

   The `--kopts` option enables you to specify options to be added to the kernel boot line. In this example, `ksdevice=eth0` specifies the network interface that kickstart should use for installation, which prevents the installation pausing to prompt you to choose which network interface to use.

3. By default, GRUB displays a boot menu for UEFI-based clients and prompts you to choose an entry. To prevent GRUB from displaying this menu, edit the `/etc/cobbler/pxe/grubsystem.template` fiile and add `default=0`, `hiddenmenu`, and `timeout=0` entries, as shown in the following example:

```
default=0
hiddenmenu
timeout=0

title $profile_name
    root (nd)
    kernel $kernel_path $kernel_options
    initrd $initrd_path
```

4. Run the `cobbler sync` command:

```
# cobbler sync
task started: YYYY-MM-DD_hhmmss_sync
task started (id=Sync, time=date)
...
generating PXE configuration files
generating: /var/lib/tftpboot/pxelinux.cfg/01-08-00-27-c6-a1-16
generating: /var/lib/tftpboot/grub/01-08-00-27-c6-a1-16
rendering DHCP files
generating /etc/dhcp/dhcpd.conf
...
*** TASK COMPLETE ***
```

   Cobbler creates pxelinux and GRUB boot configuration files for the client in the `/var/lib/tftpboot/pxelinux.cfg` and `/var/lib/tftpboot/grub` files. These files are named for the client's MAC address prefixed by `01-`, which represents the ARP hardware type for Ethernet, and use dashes to separate each byte value instead of colons. These client-specific files are based on the `/etc/cobbler/pxe/pxesystem.template` and `/etc/cobbler/pxe/grubsystem.template` files.

Cobbler also creates generic `pxelinux.cfg/default` and `grub/efidefault` boot configuration files from `/etc/cobbler/pxe/pxeprofile.template` and `/etc/cobbler/pxe/grubprofile.template`.

Cobbler adds an entry for the client to the `/etc/dhcp/dhcpd.conf` file, which is based on `/etc/cobbler/dhcp.template`, as shown in the following example:

```
# group for Cobbler DHCP tag: default
group {
    host generic1 {
        hardware ethernet 08:00:27:c6:a1:16;
        fixed-address 192.168.1.253;
        option host-name "svr1.mydom.com";
        if substring(vendorclass, 0, 9)="PXEClient" {
          if pxetype=00:06 or pxetype=00:07 {
              filename "/grub/grub.efi";
          } else {
              filename "/pxelinux.0";
          }
        }
        next-server swksvr.mydom.com;
    }
}
```

Note

If an error message is generated that indicates that the `dhcpd` service could not be started, run the following command:

```
# systemctl restart dhcpd
```

5. Display the PXE systems that are known to Cobbler.

```
# cobbler system list
   svr1.mydom.com
```

4.5.3 About Boot Loader Configuration Files

A boot-loader configuration file for BIOS-based PXE clients uses pxelinux configuration settings, as shown in the following Oracle Linux 6 example:

```
default ol6u9
prompt 0
timeout 1
label ol6u9
        kernel /images/ol6-x86_64:1:SpacewalkDefaultOrganization/vmlinuz
        ipappend 2
        append initrd=/images/ol6-x86_64:1:SpacewalkDefaultOrganization/initrd.img \
          ksdevice=bootif lang=en_US kssendmac text \
          ks=http://192.168.1.3/cblr/svc/op/ks/system/svr1.mydom.com
```

Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing. The `append` directive and all of its arguments must be on the same line.

To enable the `boot:` prompt to be displayed, change the value of `prompt` to 1. To display the prompt, press `Shift` or `Alt` at the console.

The `default` directive identifies the default boot entry by its `label` value, `ol6u6`.

Pxelinux boots the client by using the default boot entry after `timeout`/10 seconds.

The `kernel` directive defines the name of the kernel executable and the `append` directive defines any parameters that should be appended when loading the kernel, such as the name of the ram-disk image and the location of the kickstart file.

The `ipappend 2` directive specifies that the Installer should use the same network interface as the system used to boot.

For pxelinux, the kernel and ram-disk image file paths are relative to `/var/lib/tftpboot`. The default boot loader configuration file for pxelinux is `/var/lib/tftpboot/pxelinux.cfg/default`

A boot-loader configuration file for UEFI-based PXE clients uses GRUB configuration settings:

```
default=0
hiddenmenu
timeout=0

title ol6u6-x86_64-server:1:SpacewalkDefaultOrganization
    root (nd)
    kernel /images/ol6-x86_64:1:SpacewalkDefaultOrganization/vmlinuz \
      ksdevice=bootif lang=en_US kssendmac text \
      ks=http://192.168.1.3/cblr/svc/op/ks/system/svr1.mydom.com
    initrd /images/ol6-x86_64:1:SpacewalkDefaultOrganization/initrd.img
```

Note the following additional important points about boot loader configuration:

- Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing. The `kernel` directive and all of its arguments must be on the same line.
- The `timeout=0` and `hiddenmenu` directives cause the default kernel to boot immediately without allowing you to press a key to display a menu or modify the configuration of a boot entry. The default kernel is defined as the first entry (`0`), which is the only entry listed in this file.
- The `root` directive defines that the kernel and initial ram-disk image files are available on the network device `(nd)`, indicating that the files are available by using TFTP.
- The `kernel` directive defines the name of the kernel executable and any parameters that should be appended when loading the kernel, such as the location of the installation packages, and how to access these packages. The `initrd` directive specifies the initial ram-disk image file.
- For GRUB, the kernel and ram-disk image file paths are relative to `/var/lib/tftpboot/grub`. The default boot loader configuration file for GRUB is `/var/lib/tftpboot/grub/efidefault`.

To support different types of client, a configuration file can be named for the following:

- Client's UUID (for example, `a8943708-c6f6-51b9-611e-74e6ac80b93d`)
- Client's MAC address prefixed by `01-`, which represents the ARP hardware type for Ethernet, and by using dashes to separate each byte value instead of colons (for example, `01-80-00-27-c6-a1-16`)
- Client's IP address expressed in hexadecimal without any leading 0x (for example, `C0A801FD` represents the IP address `192.168.1.253`)

Cobbler writes client boot configuration files to both `/var/lib/tftpboot/grub` and `/var/lib/tftpboot/pxelinux.cfg` to handle both UEFI or BIOS-based PXE clients.

The boot loader looks for a configuration file in the following order until it finds a matching file name:

- *`UUID`* (for example, `a8943708-c6f6-51b9-611e-74e6ac80b93d`)
- `01-`*`MAC_address`* (for example, `01-80-00-27-c6-a1-16`)

- Full 32 bits of the IP address (for example, `C0A801FD`)
- Most significant 28 bits of the IP address (for example, `C0A801F`)
- Most significant 24 bits of the IP address (for example, `C0A801`)
- Most significant 20 bits of the IP address (for example, `C0A80`)
- Most significant 16 bits of the IP address (for example, `C0A8`)
- Most significant 12 bits of the IP address (for example, `C0A`)
- Most significant 8 bits of the IP address (for example, `C0`)
- Most significant 4 bits of the IP address (for example, `C`)
- `default` (BIOS) or `efidefault` (EFI)

For more information about GRUB, type the `info grub` command to access the GRUB manual.

For more information about pxelinux, see https://wiki.syslinux.org/wiki/index.php?title=PXELINUX.

4.5.4 Configuring DHCP to Support iPXE Clients

iPXE extends the capabilities of PXE in many ways, including the following:

- iPXE clients can boot by using HTTP, iSCSI, AoE, and FCoE
- The boot process can be controlled by using scripts
- DNS lookup is available
- Booting across wide area networks or the Internet is possible

The `gpxelinux.0` boot loader provides some iPXE features, such as DNS lookup and HTTP file transfer, and is available in the `syslinux` package. It does not support iPXE commands or scripts.

You can use `gpxelinux.0` with BIOS-based PXE clients and with UEFI-based PXE clients in legacy mode but not in UEFI mode.

To configure the DHCP service to support iPXE clients, follow these steps:

1. Edit the DHCP server configuration template file `/etc/cobbler/dhcp.template` as follows:
   a. Add the following lines to define the iPXE options for DHCP:

```
option space ipxe;
option ipxe-encap-opts code 175 = encapsulate ipxe;
option ipxe.priority code 1 = signed integer 8;
option ipxe.keep-san code 8 = unsigned integer 8;
option ipxe.skip-san-boot code 9 = unsigned integer 8;
option ipxe.syslogs code 85 = string;
option ipxe.cert code 91 = string;
option ipxe.privkey code 92 = string;
option ipxe.crosscert code 93 = string;
option ipxe.no-pxedhcp code 176 = unsigned integer 8;
option ipxe.bus-id code 177 = string;
option ipxe.bios-drive code 189 = unsigned integer 8;
option ipxe.username code 190 = string;
option ipxe.password code 191 = string;
option ipxe.reverse-username code 192 = string;
option ipxe.reverse-password code 193 = string;
option ipxe.version code 235 = string;
option iscsi-initiator-iqn code 203 = string;
```

```
option ipxe.pxeext code 16 = unsigned integer 8;
option ipxe.iscsi code 17 = unsigned integer 8;
option ipxe.aoe code 18 = unsigned integer 8;
option ipxe.http code 19 = unsigned integer 8;
option ipxe.https code 20 = unsigned integer 8;
option ipxe.tftp code 21 = unsigned integer 8;
option ipxe.ftp code 22 = unsigned integer 8;
option ipxe.dns code 23 = unsigned integer 8;
option ipxe.bzimage code 24 = unsigned integer 8;
option ipxe.multiboot code 25 = unsigned integer 8;
option ipxe.slam code 26 = unsigned integer 8;
option ipxe.srp code 27 = unsigned integer 8;
option ipxe.nbi code 32 = unsigned integer 8;
option ipxe.pxe code 33 = unsigned integer 8;
option ipxe.elf code 34 = unsigned integer 8;
option ipxe.comboot code 35 = unsigned integer 8;
option ipxe.efi code 36 = unsigned integer 8;
option ipxe.fcoe code 37 = unsigned integer 8;
option ipxe.vlan code 38 = unsigned integer 8;
option ipxe.menu code 39 = unsigned integer 8;
option ipxe.sdi code 40 = unsigned integer 8;
option ipxe.nfs code 41 = unsigned integer 8;
```

b. If you do not use a proxy DHCP server, specify the following line to speed up negotiation with the DHCP server:

```
option ipxe.no-pxedhcp 1;
```

c. Add the following line to define the `user-class` option:

```
option user-class code 77 = string;
```

d. Configure the DHCP server to provide the IP addresses of name servers that iPXE clients can use to resolve domain names to IP addresses:

```
option domain-name-servers 192.168.1.1, 192.168.1.4, 192.168.1.8;
```

e. Configure DHCP to specify the `gpxelinux.0` boot loader for non-iPXE clients and the URI of a boot script for iPXE clients:

```
if exists user-class and option user-class = "iPXE" {
      filename "http://web.mydom.com/pxeboot.ipxe";
  } else {
      filename "gpxelinux.0";
  }
```

In the previous example, pure iPXE clients run the HTTP-served boot script `pxeboot.ipxe`.

The following is an example of a boot script for an iPXE client:

```
#!ipxe

dhcp
kernel http://swksvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/pxeboot/vmlinuz
initrd http://swksvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/pxeboot/initrd.img
boot vmlinuz initrd=initrd.img ksdevice=bootif lang=en_US kssendmac text \
```

```
ks=http://192.168.1.3/cblr/svc/op/ks/profile/ol6-x86_64-minimal:1:SpacewalkDefaultOrganization
```

`dhcp` configures the client's network interfaces.

`kernel` downloads the installation kernel.

`initrd` downloads the initial ram-disk image file.

`boot` boots the downloaded installation kernel. Boot line parameters, such as the name of the initial ram-disk file and the location of the kickstart file, are specified as additional arguments.

Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing. The `boot` command and all of its arguments must be on the same line.

For more information, see https://ipxe.org/scripting and https://ipxe.org/cmd.

Non-iPXE clients boot by using `gpelinux.0`. A configuration file for `gpxelinux.0` is named in the same way as for `pxelinux.0` as described in Section 4.5.3, “About Boot Loader Configuration Files”. Unlike `pxelinux.0`, you can use HTTP to access the installation kernel and initial ram-disk image files.

The following is an example of a configuration file for `gpxelinux.0`:

```
prompt 0
default ol6u6
timeout 0

label ol6u9
kernel http://swksvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/pxeboot/vmlinuz
append initrd=http://swksvr.mydom.com/distro-trees/ol6u9-x86_64-server/images/pxeboot/initrd.img \
ksdevice=bootif lang=en_US kssendmac text \
ks=http://192.168.1.3/cblr/svc/op/ks/profile/ol6-x86_64-minimal:1:SpacewalkDefaultOrganization
ipappend 2
```

Do *not* use the `\` line-continuation character. This character is used in the example to denote that the line has been broken for printing. The `append` keyword and all of its arguments must be on the same line.

2. Run the `cobbler sync` command:

```
# cobbler sync
task started: YYYY-MM-DD_hhmmss_sync
task started (id=Sync, time=date)
...
rendering DHCP files
generating /etc/dhcp/dhcpd.conf
...
*** TASK COMPLETE ***
```

The Cobbler service regenerates the `/etc/dhcp/dhcpd.conf` file and reloads the `dhcpd` service.

If you make any further changes to `/etc/cobbler/dhcp.template`, run the `cobbler sync` command. You do not need to run this command if you change the content of the boot loader configuration files.

4.6 Creating a Kickstart Profile in Cobbler

You can also create kickstart profiles in Cobbler outside of Spacewalk. As for Spacewalk, a profile defines how to configure an installation if the target client has to perform a certain role. For example, you might

want to configure a system as a web or database server. To create a profile in Cobbler, you associate a kickstart file with a distribution.

Note

Cobbler-only profiles are not visible from within Spacewalk.

You can use the `cobbler profile list` command as follows to list the profiles that are known to Cobbler:

```
# cobbler profile list
   ol6u9-x86_64
```

To find out which kickstart file a profile uses, run the `cobbler profile report` command:

```
# cobbler profile report ol6u9-x86_64 | grep Kickstart
Kickstart                      : /var/lib/cobbler/kickstarts/sample.ks
Kickstart Metadata             : {}
```

The default `sample.ks` file and other kickstart files that Cobbler provides in `/var/lib/cobbler/kickstarts` are unlikely to be suitable for provisioning clients.

To create a new profile for a distribution, follow these steps:

1. Create the kickstart file to associate with a distribution.

   For example, the following file, named `ol6u6_basic_server.ks`, contains a kickstart definition for a basic Oracle Linux 6 server:

```
# Oracle Linux 6 Basic Server
# Use text-based installation
text
# Install using HTTP from a URL provided by Cobbler
url --url=$tree
# Define localized settings
lang en_US.UTF-8
keyboard us
timezone --utc America/New_York
# Configure network interface settings
network --onboot yes --device eth0 --bootproto dhcp --noipv6
# root password is an SHA-512 hash provided by Cobbler
rootpw  --iscrypted $default_password_crypted
authconfig --enableshadow --passalgo=sha512
# Allow only SSH connections
firewall --service=ssh
# Configure SELinux enforcing mode
selinux --enforcing
# Perform a new installation, removing all existing partitions
# before configuring the new boot loader and disk partitions
install
zerombr
clearpart --drives=sda --all --initlabel
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb quiet"
autopart
# Shutdown and power off the system after installation is finished
# to allow you to change the boot order or make other changes.
poweroff
# Alternatives are halt (default), reboot, and shutdown,
# which might not be suitable for unattended installations

# Package groups and packages to be installed
%packages
...
%end
```

For sample package lists, see Appendix B, *Sample Minimum Package Lists*.

Note

This example requires that you configure an SHA-512 password hash for the `default_password_crypted` setting in `/etc/cobbler/settings`.

The previous example does not take advantage of the power of kickstart templating and snippets for managing large numbers of profiles and systems in Cobbler. For more information, see the `cobbler(1)` manual page and http://www.cobblerd.org/.

2. If SELinux is enabled in `enforcing` mode on your system and you create the kickstart file in a directory other than `/var/lib/cobbler/kickstarts`, for example `/var/kickstart`, do the following:

   a. Type the `semanage` command to define the default file type of the directory hierarchy as `cobbler_var_lib_t`:

   ```
   # /usr/sbin/semanage fcontext -a -t cobbler_var_lib_t "/var/kickstart(/.*)?"
   ```

   b. Type the `restorecon` command to apply the file type to the entire directory hierarchy.

   ```
   # /sbin/restorecon -R -v /var/kickstart
   ```

   c. For each kickstart file in the directory, type the `chcon` command to set the SELinux user to `system_u`.

   ```
   # chcon -u system_u /var/kickstart/*.ks
   ```

   If SELinux is enabled in `enforcing` mode on your system, and you create the kickstart file in `/var/lib/cobbler/kickstarts` or in a directory on which you have defined the default file type as `cobbler_var_lib_t`, type the `chcon` command to set the SELinux user of the file to `system_u`:

   ```
   # chcon -u system_u ol6u6_basic_server.ks
   ```

   You can use the `ls -Z` command to display the context:

   ```
   # ls -Z ol6u9_basic_server.ks
   -rw-rw-r--. root root system_u:object_r:cobbler_var_lib_t:s0 ol6u9_basic_server.ks
   ```

   The correct SELinux context for a kickstart file used by Cobbler is `system_u:object_r:cobbler_var_lib_t:s0`.

3. Create the profile by running the `cobbler profile add` command:

   ```
   # cobbler profile add --name=ol6u9_basic_server --distro=ol6u9-x86_64 \
   --kickstart=/var/lib/cobbler/kickstarts/ol6u9_basic_server.ks
   ```

Note

If the command returns the error `kickstart not found` for a file that does exist at the specified path, the file's SELinux context is incorrect. See the previous step for details of how to set the correct SELinux context on a file.

4. Display the profiles that are now known to Cobbler by running the `cobbler profile list` command.

   ```
   # cobbler profile list
      ol6u9-x86_64
   ```

```
   ol6u9_basic_server
```

The `ol6u9-x86_64` profile is unlikely to be usable. If you want to remove a profile, use the `cobbler profile remove` command, as shown in the following example:

```
# cobbler profile remove --name=ol6u9-x86_64
# cobbler profile list
   ol6u9_basic_server
```

Note

Removing a profile also removes any client `system` definitions that you have created from that profile.

You can now define the PXE clients that Cobbler can provision based on the profile that you created. See Section 4.6.1, “Adding a PXE Client To Be Provisioned by Cobbler”.

4.6.1 Adding a PXE Client To Be Provisioned by Cobbler

To add a PXE client to be provisioned by Cobbler, follow these steps:

1. Define the host name, MAC address, and IP address of the target PXE client and the profile that you want to installl by using the `cobbler system add` command:

   ```
   # cobbler system add --name=svr1 --hostname=svr1 --mac=08:00:27:c6:a1:16 \
   --ip=10.0.0.253 --profile=ol6u9_basic_server
   ```

   If you are provisioning a desktop client that uses an IP address from a DHCP address pool, you might use a command such as the following:

   ```
   # cobbler system add --name=devsys2 --hostname=devsys2 --profile=ol6u9_devsys --kopts="ksdevice=eth0"
   ```

   The `--kopts` option enables you to specify options to be added to the kernel boot line. In this example, `ksdevice=eth0` specifies the network interface that kickstart should use for installation, which prevents the installation pausing to prompt you to choose which network interface to use.

2. By default, GRUB displays a boot menu for UEFI-based clients and prompts you to choose an entry. To prevent GRUB from displaying this menu, edit `/etc/cobbler/pxe/grubsystem.template` and add `default=0`, `hiddenmenu`, and `timeout=0` entries, as shown in the following example:

   ```
   default=0
   hiddenmenu
   timeout=0

   title $profile_name
       root (nd)
       kernel $kernel_path $kernel_options
       initrd $initrd_path
   ```

3. Run the `cobbler sync` command.

   ```
   # cobbler sync
   task started: YYYY-MM-DD_hhmmss_sync
   task started (id=Sync, time=date)
   ...
   generating PXE configuration files
   generating: /var/lib/tftpboot/pxelinux.cfg/01-08-00-27-c6-a1-16
   generating: /var/lib/tftpboot/grub/01-08-00-27-c6-a1-16
   rendering DHCP files
   generating /etc/dhcp/dhcpd.conf
   ...
   *** TASK COMPLETE ***
   ```

Cobbler creates pxelinux and GRUB boot configuration files for the client in `/var/lib/tftpboot/pxelinux.cfg` and `/var/lib/tftpboot/grub`. These files are named for the client's MAC address prefixed by `01-`, which represents the ARP hardware type for Ethernet, and use dashes to separate each byte value instead of colons. These client-specific files are based on `/etc/cobbler/pxe/pxesystem.template` and `/etc/cobbler/pxe/grubsystem.template`.

Cobbler also creates generic `pxelinux.cfg/default` and `grub/efidefault` boot configuration files from `/etc/cobbler/pxe/pxeprofile.template` and `/etc/cobbler/pxe/grubprofile.template`.

Cobbler adds an entry for the client to `/etc/dhcp/dhcpd.conf`, which is based on `/etc/cobbler/dhcp.template`, as shown in the following example:

```
# group for Cobbler DHCP tag: default
group {
    host generic1 {
        hardware ethernet 08:00:27:c6:a1:16;
        fixed-address 10.0.0.253;
        option host-name "svr1";
        if substring(vendorclass, 0, 9)="PXEClient" {
          if pxetype=00:06 or pxetype=00:07 {
              filename "/grub/grub.efi";
          } else {
              filename "/pxelinux.0";
          }
        }
        next-server 10.0.0.6;
    }
}
```

4. Display the systems that are known to Cobbler by running the `cobbler system list` command:

```
# cobbler system list
   svr1
   svr2
```

4.6.2 Removing a PXE Client Definition From Cobbler

To remove a PXE Client definition from Cobbler, follow these steps:

1. Display the systems that are known to Cobbler by typing the `cobbler system list` command:

```
# cobbler system list
   svr1
   svr2
```

2. Specify the name of the system that you want to remove by typing the `cobbler system remove`, as shown in the following example where `svr2` is the system that you want to remove:

```
# cobbler system remove --name=svr2
```

3. Run the `cobbler sync` to update the Cobbler configuration.

```
# cobbler sync
task started: YYYY-MM-DD_hhmmss_sync
task started (id=Sync, time=date)
...
generating PXE configuration files
rendering DHCP files
generating /etc/dhcp/dhcpd.conf
...
*** TASK COMPLETE ***
```

4. Verify that `svr2` has been removed as follows:

```
# cobbler system list
   svr1
```

4.7 Provisioning KVM Hosts by Using Spacewalk

The following procedure describes how to use Spacewalk to provision a KVM host. Before performing this procedure, ensure that you have done the following:

- You are familiar with how to set up and use activation keys and kickstart profiles in Spacewalk and how to configure Cobbler, DHCP, and boot loaders to support network installation of client systems.

  See Chapter 3, *Working With Activation Keys in Spacewalk*, Section 4.4, “Working With Kickstart Profiles”, and Section 4.5, “Installing Client Systems by Using Kickstart”.

- You have set up a base channel and kickstartable tree for the Oracle Linux distribution that you want to install on the KVM host.

- The system that you configure as a KVM host must have VT-x acceleration enabled in the BIOS or UEFI firmware and be able to forward this capability to any KVM guests. Suitable systems are bare-metal systems with VT-x enabled and Oracle VM virtual machines that have been configured with this capability. Oracle VirtualBox virtual machines do not support this functionality and are not suitable.

To provision a KVM host, follow these steps:

1. In Spacewalk, create an activation key that is specific to KVM hosts on the desired platform: Oracle Linux 6 (x86_64) or Oracle Linux 7 (x86_64).

   Enter the key settings as follows:

   Description — Enter a description for the key, for example, `Oracle Linux 7 (x86_64) KVM host`.

   Key — Enter a meaningful label for the activation key, for example, `kvmhost-oraclelinux7-x86_64`.

   Usage — Leave blank to enable unlimited use by clients.

   Base Channels — Select the base channel with which the key is associated: `Oracle Linux 7 (x86_64) Base`.

   Add-on Entitlements — If you are running the KVM hypervisor on Oracle Linux 6 or Oracle Linux 7, you should enable the **Virtualization** entitlement. This entitlement installs additional packages on the target host so that Spacewalk is able to identify and manage virtual guests on that host.

   Universal Default — Select if the key should be used as the default activation key for all newly-registered systems.

   Note

   Oracle strongly recommends that you do not associate any channels with a universal default key. Spacewalk uses the universal default key if a key is not specified so it might

be used by any version of any operating system.

2. In Spacewalk, create a kickstart profile for KVM host systems on the desired platform and do the following:

   a. Associate the activation key that you created in Step 1 with the profile.

   b. Enter the profile settings as follows:

Label	Enter a label for the profile. for example, `kvmhost-ol7u4-x86_64`.
Base Channel	Select the base channel with which the distribution is associated, for example, `Oracle Linux 7 Update 4 (x86_64) Base`.
Kickstartable Tree	Select the kickstart distribution with which the profile is associated, for example, `ol7u4-x86_64-server`.
Virtualization Type	Select the virtualization type as **None**.

   c. Configure the following software packages, which kickstart should install on the host, in addition to the `@Base` and `@Core` packages:

Virtualization packages (required for a KVM host):	• `@virtualization-hypervisor` • `@virtualization-tools`
Virtualization packages (recommended):	• `qemu-kvm-tools` (provides debugging and diagnostic utilities.) • `virt-manager` (provides a graphical virtual machine manager that you can use with KVM.) • `virt-viewer` (provides a graphical console client for connecting to virtual machines)
Graphical desktop packages (required to use the Virtual Machine Manager):	• `@^graphical-server-environment` (provides a full graphical server environment) • `@fonts` • `@gnome-desktop` (select alternate desktop environment such as KDE if preferred) • `@x11`
Spacewalk client packages (recommended):	• `rhncfg` • `rhncfg-actions` • `rhncfg-client`
Suggested optional packages:	• `@input-methods` (Only include if the `graphical-server-environment` is enabled. Otherwise, not recommended.)

- `@internet-browser` (Only include if the `graphical-server-environment` is enabled. Otherwise, not recommended.)
- `@multimedia` (Only include if the `graphical-server-environment` is enabled. Otherwise, not recommended.)
- `kexec-tools`
- `osad` (enables you to apply updates and actions to a client system immediately from the Spacewalk server)

d. In the kickstart profile, configure any kickstart advanced options that you require, such as `keyboard`, `lang`, or `network`.

e. Set up the `%pre` or `%post` sections for any pre or post-installation configuration that you want kickstart to perform.

For example, you can enable configuration file management and remote commands by including the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages and configuring `rhn-actions-control` to run in the post-installation shell:

```
rhn-actions-control --enable-all
```

3. Configure Cobbler or DHCP to provide IP and TFTP settings so that the guest being installed can access the appropriate boot loader to continue the provisioning process.
4. Configure the boot-loader configuration file that the boot loader uses to locate the installation kernel, the ram-disk image, and the kickstart file that is served by Spacewalk.
5. After setting up the Spacewalk kickstart profile, Cobbler, DHCP, and boot-loader configuration, boot the target host system from the network, which starts the installation process.

4.8 Provisioning KVM Guests by Using Spacewalk

The following procedure describes how to use Spacewalk to provision KVM guests.

Before you begin, ensure that you have done the following:

- Familiarize yourself with how to set up and use activation keys and kickstart profiles in Spacewalk and how to configure Cobbler, DHCP, and boot loaders to support network installation of client systems.

  See Chapter 3, *Working With Activation Keys in Spacewalk*, Section 4.4, “Working With Kickstart Profiles”, and Section 4.5, “Installing Client Systems by Using Kickstart”.

- Familiarize yourself with how to use KVM to configure a KVM guest, for example by using the graphical Virtual Machine Manager.

Note

To use KVM to configure a KVM guest, you must install the `spacewalk-koan` package on the KVM host.

- You have set up a base channel and kickstartable tree for the Oracle Linux distribution that you want to install on the KVM guest.

To provision a KVM guest, follow these steps:

1. In Spacewalk, create an activation key that is specific to KVM hosts on the desired platform: Oracle Linux 6 (x86_64) or Oracle Linux 7 (x86_64). Enter the key settings as follows:

   Description — Enter a description for the key for example, `Oracle Linux 7 (x86_64) KVM guest`.

   Key — Enter a meaningful label for the activation key, for example, `kvmguest-oraclelinux7-x86_64`.

   Usage — Leave blank to enable unlimited use by clients.

   Base Channels — Select the base channel with which the key is associated:

   `Oracle Linux 7 (x86_64) Base`.

   Add-on Entitlements

   Note

   No add-on entitlements are required for a virtual guest. The Virtualization entitlement is only required for virtual hosts. Do not select it when creating a virtual guest kickstart profile.

   Universal Default — Select this option if the key should be used as the default activation key for all newly-registered systems.

   Note

   Oracle strongly recommends that you do not associate any channels with a universal default key. Spacewalk uses the universal default key if a key is not specified so it might be used by any version of any operating system.

2. In Spacewalk, create a kickstart profile for KVM host systems on the desired platform:

   a. Associate the activation key that you created with the profile.

   b. Enter the profile settings as follows:

      Label — Enter a label for the profile, for example, `kvmguest-ol7u4-x86_64`.

      Base Channel — Select the base channel with which the distribution is associated, for example, `Oracle Linux 7 Update 4 (x86_64) Base`.

      Kickstartable Tree — Select the kickstart distribution with which the profile is associated, for example, `ol7u4-x86_64-server`.

      Virtualization Type — For a KVM guest, select the virtualization type as **KVM Virtualized Guest**. KVM supports only HVM guests.

   c. Configure the software packages that kickstart should install on the host in addition to the `@Base` package. The intended function of the guest system determines the set of packages. However,

Oracle recommends the following additional packages for a KVM guest that is also a Spacewalk client:

- `@guest-agents` (agents used when running under a hypervisor)
- `@guest-desktop-agents` (agents used when running as a virtualized desktop)
- `acpid` (enables you to control the power state of the guest from the host)
- `osad` (enables you to apply updates and actions to a client system immediately from the Spacewalk server)
- `rhncfg`
- `rhncfg-actions`
- `rhncfg-client`

d. In the kickstart profile, configure any kickstart advanced options that you require, such as `keyboard`, `lang`, or `network`.

e. Set up the `%pre` or `%post` sections for any pre or post-installation configuration that you want kickstart to perform.

For example, you can enable configuration file management and remote commands by including the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages and configuring `rhn-actions-control` to run in the post-installation shell:

```
rhn-actions-control --enable-all
```

3. If you want to install the guest by using PXE network booting and kickstart, do the following:

a. Configure Cobbler or DHCP to provide IP and TFTP settings so that the guest being installed can access the appropriate boot loader to continue the provisioning process.

b. Configure the configuration file that the boot loader uses to locate the installation kernel, the ram-disk image, and the kickstart file served by Spacewalk.

c. After setting up the Spacewalk kickstart profile, Cobbler, DHCP, and boot-loader configuration, boot the target guest system from the network to start the installation process.

If you want to install the guest by using a network installation, use a boot image that is made available over HTTP by the Spacewalk server. You can use a full ISO image, a UEK boot image, or a RHCK boot image in conjunction with kickstart, depending on your requirements.

Chapter 5 Registering Client Systems With Spacewalk

This chapter describes how to register client systems with Spacewalk. Before you register a system with Spacewalk, you should create an activation key to use with client systems, as described in Chapter 3, *Working With Activation Keys in Spacewalk*. It is possible to register a system without an activation key by providing a user name and password, but Spacewalk does not perform channel subscription or package installation in this case. Rather than using a default activation key, Oracle recommends using an activation key that is specific to the Oracle Linux release and system architecture.

Spacewalk registration is usually performed by Spacewalk's provisioning service. For existing or manually installed systems, you can alternatively use the `rhnreg_ks` command to register the system with Spacewalk.

Note

Do not register a Spacewalk server or client with ULN. Oracle recommends that you register the Spacewalk server as a client of itself after you have set up the software channels. Include the Spacewalk server channel in the list of software channels to which the server is subscribed.

5.1 Registering a Client System by Using Kickstart

If you install a system by using a kickstart file that is generated from a Spacewalk profile, Spacewalk automatically registers the system as a Spacewalk client if the following conditions are met:

- The Spacewalk Client channel must be selected on the Modify Operating System page under the profile's **Kickstart Details** tab.
- An activation key is associated with the profile on the Kickstart Details page under the profile's **Activation Keys** tab.

5.2 Installing the Spacewalk Client Software and Registering a Client System

Note

Starting with Oracle Linux 7 Update 1 and Oracle Linux 6 Update 9, you do not need to install the Spacewalk Client software for Relesae 2.7 before you register an Oracle Linux 7 or Oracle Linux 6 server with Spacewalk. See Section 5.3, “Registering a Client System by Using rhnreg_ks Without First Installing the Spacewalk Client Software”.

To install the Spacewalk Client software on an Oracle Linux server and register the server as a Spacewalk client:

1. Enable access to the repositories that contain the Spacewalk Server and any dependent packages on the Oracle Linux yum server at https://yum.oracle.com.

   For Oracle Linux 6, do the following:

   a. Ensure that your system is up to date and that you have transitioned to use the modular yum repository configuration by installing the `oraclelinux-release-el6` package and running the `/usr/bin/ol_yum_configure.sh` script.

```
# yum install oraclelinux-release-el6
# /usr/bin/ol_yum_configure.sh
```

b. Install the `oracle-spacewalk-client-release-el6` release package to install appropriate yum repository configuration.

```
# yum install oracle-spacewalk-client-release-el6
```

The `ol6_spacewalk27_client` repository is enabled by default in the repository configuration file.

For Oracle Linux 7, do the following:

a. Ensure that your system is up to date and that you have transitioned to use the modular yum repository configuration by installing the `oraclelinux-release-el7` package and running the `/usr/bin/ol_yum_configure.sh` script.

```
# yum install oraclelinux-release-el7
# /usr/bin/ol_yum_configure.sh
```

b. Install the `oracle-linux-manager-client-release-el7` release package to install appropriate yum repository configuration. This package reflects the new naming convention but includes the proper information for Spacewalk 2.7.

```
# yum install oracle-linux-manager-client-release-el7
```

Then enable the correct Spacewalk repositories for Spacewalk 2.7.

```
# yum-config-manager --disable ol7_oracle-linux-manager210_client
# yum-config-manager --enable ol7_spacewalk27_client
```

2. Install the Spacewalk Client software.

```
# yum install rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin
```

This command replaces the existing packages and deletes any previous registration with ULN.

3. Download the CA certificate file (`RHN-ORG-TRUSTED-SSL-CERT`) to the server.

In a browser, navigate to **`http://swksvr_FQDN/pub`**, where *`swksvr_FQDN`* is the fully qualified domain name of the Spacewalk server, and download the CA certificate file (`RHN-ORG-TRUSTED-SSL-CERT`) to the `/usr/share/rhn/` directory.

You can download the file by using the `wget` command, for example:

```
# wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT \
  http://swksvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT
```

Alternatively, you can install the package that is automatically generated when you install SSL certificates, for example:

```
# yum install http://swksvr_FQDN/pub/rhn-org-trusted-ssl-cert-1.0-1.noarch.rpm
```

Note that you might need to specify a different URL if you replaced the SSL certificates after installing and configuring the Spacewalk server software.

4. Register the system with Spacewalk by using the `rhnreg_ks` command, using the `--sslCACert` option to specify the certificate:

```
# rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT \
```

```
  --serverUrl=https://swksvr_FQDN/XMLRPC --activationkey=activation_key
```

Make sure to specify the Spacewalk server or proxy by its fully qualified domain name.

If you need to re-register a Spacewalk client with a Spacewalk server, additionally specify the `--force` option.

5. Disable access to the Spacewalk Client repository in the Oracle Linux yum server repository configuration file or delete `/etc/yum.repos.d/spacewalk27-client.repo`.

5.3 Registering a Client System by Using rhnreg_ks Without First Installing the Spacewalk Client Software

Starting with Oracle Linux 7 Update 1 and Oracle Linux 6 Update 9, you do not need to install the Spacewalk Client software before you register an Oracle Linux 7 or Oracle Linux 6 server with Spacewalk.

To register an Oracle Linux server as a Spacewalk client:

1. Download the CA certificate file `RHN-ORG-TRUSTED-SSL-CERT` to the server.

   In a browser tab, navigate to **`http://swksvr_FQDN/pub`**, where *`swksvr_FQDN`* is the fully qualified domain name of the Spacewalk server, and download the CA certificate file `RHN-ORG-TRUSTED-SSL-CERT` to `/usr/share/rhn/`.

   Alternatively, you can use `wget` from the command line, for example:

```
# wget -q -O /usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT \
  http://swksvr_FQDN/pub/RHN-ORG-TRUSTED-SSL-CERT
```

2. Register the system with Spacewalk by using the `rhnreg_ks` command with the `--sslCACert` option to specify the certificate.

```
# rhnreg_ks --sslCACert=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT \
  --serverUrl=https://swksvr_FQDN/XMLRPC --activationkey=activation_key
```

   Specify the Spacewalk server or proxy by its fully qualified domain name.

   If you need to re-register a Spacewalk client with a Spacewalk server, additionally specify the `--force` option.

3. To install the Spacewalk Client software after registration, subscribe the server to a Spacewalk client software channel and use the `yum` command to install the packages:

```
# yum install rhn-client-tools rhn-check rhn-setup rhnsd m2crypto yum-rhn-plugin
```

Note

Oracle recommends that you install the Spacewalk Client software after registration to support all of the features provided by Spacewalk, which include provisioning and auditing.

Chapter 6 Configuring Client Systems for Immediate Updates

By default, the `rhnsd` daemon on a client system connects to the Spacewalk server every four hours and performs any updates or actions that you have scheduled. If you install the OSA daemon, you can apply updates and actions to client systems immediately from the Spacewalk server.

6.1 Enabling the OSA Daemon in a Kickstart Profile by Using the Spacewalk Web Interface

Note

The following procedure applies if you use Spacewalk to generate the kickstart file.

Configure a kickstart profile to install and enable the OSA daemon on a client system as follows:

1. Go to **Systems**, select **Kickstart** and then **Profiles**.
2. Select the profile, and then select **Software** to display the Package Groups page.
3. On the Package Groups page, include `osad` in the list of packages to install.
4. Select **System Details** to display the Details page.
5. On the Details page, select **Advanced Options**, enable the `services` option and add the following entry:

   ```
   --enabled=osad
   ```

   Spacewalk adds the following option to the generated kickstart file:

   ```
   services --enabled=osad
   ```

   The `osad` service starts automatically at the default run level when the target client system reboots following installation.
6. Click **Update Kickstart Distribution** to save your changes.

6.2 Enabling the OSA Daemon in a Kickstart File

Note

The following procedure applies if you upload a kickstart file into a profile.

If you want to be able to apply updates and actions to a client system immediately from the Spacewalk server:

- Include the `osad` package for installation.
- Include the following kickstart option to enable the `osad` service:

  ```
  services --enabled=osad
  ```

6.3 Enabling the OSA Daemon Manually

Install and enable the OSA daemon manually as follows:

1. Log in as `root` on the client system.
2. Use the `yum` command to install the `osad` package:

```
# yum install osad
```

3. Enable and start the `osad` service:
   - On an Oracle Linux 5 or Oracle Linux 6 client system:

```
# chkconfig osad on
# service osad start
```

   - On an Oracle Linux 7 client system:

```
# systemctl enable osad
# systemctl start osad
```

4. Check that the `osa` daemon is online.
5. If the `osad` service does not start and displays the error `SSLDisabledError`, edit `/etc/sysconfig/rhn/up2date`, verify the following for the `serverURL` entry:
   a. URL uses `https://`
   b. URL uses the fully qualified domain name of the Spacewalk server or proxy, as shown in the following example:

```
serverURL=https://swksvr.mydom.com/XMLRPC
```

   If one or both of these settings are not correct, you will need to re-register the client. See Section 5.3, “Registering a Client System by Using rhnreg_ks Without First Installing the Spacewalk Client Software” for instructions.

6.4 Replacing the jabberd/osa Database

You might need to replace the `jabberd`/`osa` database on a Spacewalk server or proxy if you encounter any of the following errors:

- OSA status shows "offline as of unknown" for client servers.
- `osa-dispatcher` errors in `/var/log/mesasges` on the Spacewarlk server or proxy.
- "db: corruption detected! close all jabberd processes and run db_recover" message in `/var/log/messages` on the Spacewalk server or proxy.

This problem occurs because the default Berkeley database format does not support transactions, and as a result, can become damaged when too many clients attempt to update at the same time. Switching to SQLite provides transactional support for the `jabberd` database and can handle significantly more downstream clients.

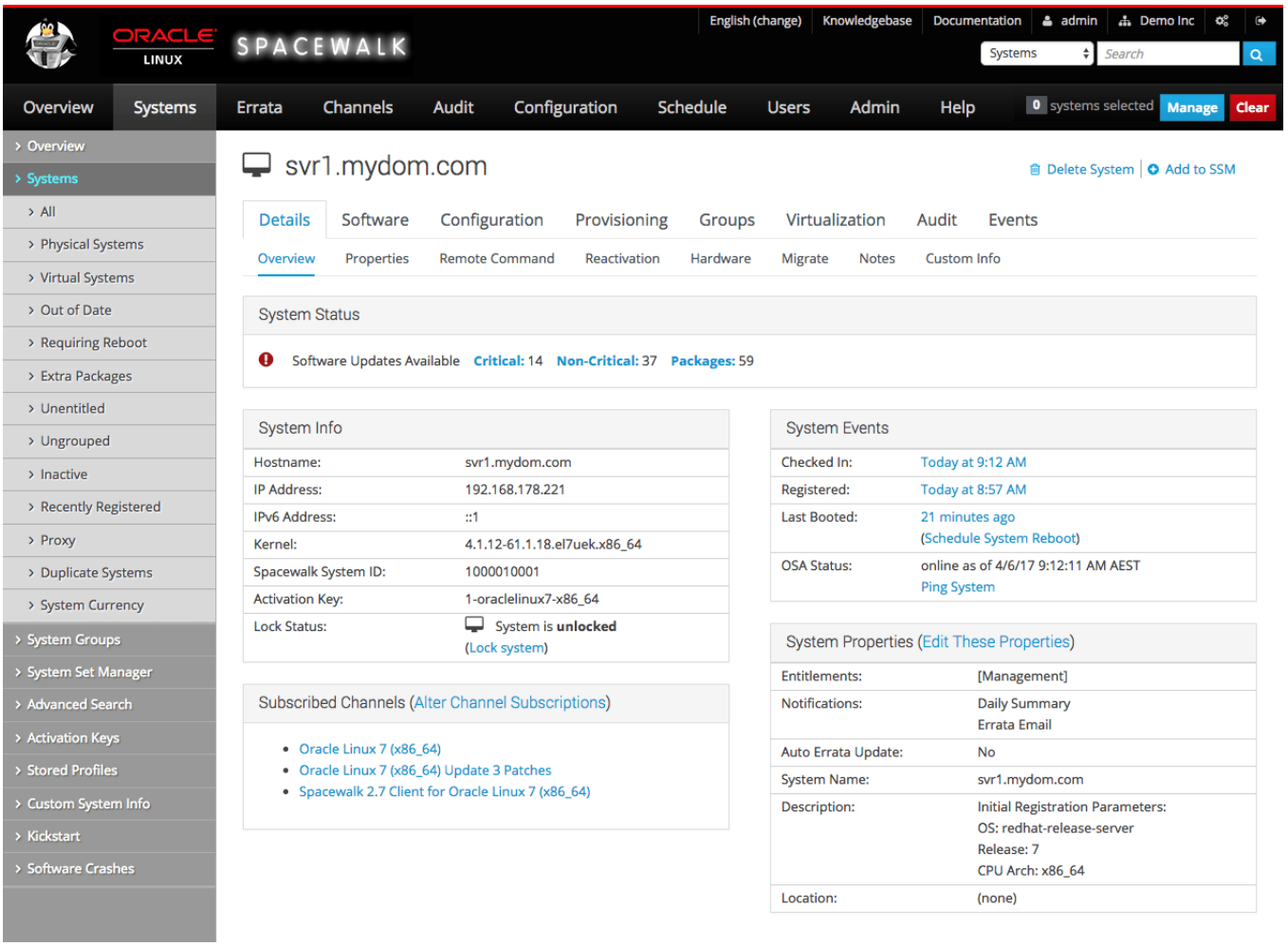
The workaround for this issue is to replace the database. See *Replacing the jabberd/osa Database on Spacewalk Servers or Spacewalk Proxies* in *Spacewalk for Oracle® Linux: Installation Guide for Release 2.7* for instructions.

6.5 Querying the Status of Client Systems

You can use either the Spacewalk web interface or the `spacecmd` command to query the status of client systems.

6.5.1 Querying the Status of a Client System by Using the Spacewalk Web Interface

Figure 6.1 System Status Page



To verify the status of an active client system:

1. Go to **Systems** and select the client system from the list.

   The Overview page displays the following information for the client system:

   - The System Status pane shows how many critical errata updates, non-critical errata updates, and packages are available to install on the client system. Select **Critical**, **Non-Critical**, or **Packages** to view and optionally install the available errata and packages.
   - The System Info pane shows the host name, IP addresses, kernel version, Spacewalk system ID, activation key, and whether the system is locked.
   - The Subscribed Channels pane shows the base and child channels to which the client system is subscribed.
   - The System Events pane shows when the client system last checked in, when it was registered, when it last booted, and when the OSA daemon last started.

- The System Properties pane shows the entitlements, notifications, automatic errata update status, system name, summary of the installed operating system, and location.

2. If the OSA status is shown as `online as of unknown`, select **Ping System**.

3. Wait a few seconds and then reload the page. The status should update and display when the OSA daemon was last started.

6.5.2 Querying the Status of a Client System by Using the spacecmd Command

Verify the status of a client system, use the `system_details` command as follows:

```
spacecmd {SSM:0}> system_details swksvr.mydom.com
Name:          swksvr.mydom.com
System ID:     1000010000
Locked:        False
Registered:    20170405T13:05:50
Last Checkin:  20170419T06:51:47
OSA Status:    online
Last Boot:     20170405T12:05:04

Hostname:      swksvr.mydom.com
IP Address:    192.168.178.220
Kernel:        4.1.12-61.1.33.el7uek.x86_64

Activation Keys
---------------
1-oraclelinux7-x86_64

Software Channels
-----------------
ol7-x86_64-u4-base
  |-- ol7-x86_64-u4-patch

Entitlements
------------
enterprise_entitled

System Groups
-------------
group1
```

Chapter 7 Configuring Client Systems for Remote Management

You can install the remote configuration client packages on the remote system, which enable you to perform remote configuration of client system from Spacewalk,

7.1 Enabling Remote Configuration in a Kickstart Profile by Using the Spacewalk Web Interface

Note

The following procedure applies if you use Spacewalk to generate the kickstart file.

To configure a kickstart profile to install and enable the remote configuration client software on a client system:

1. Go to **Systems**, select **Kickstart** and then **Profiles**.
2. Select the profile, and then select **Software** to display the Package Groups page.
3. On the Package Groups page, include `rhncfg`, `rhncfg-actions`, and `rhncfg-client` in the list of packages to install.
4. Select **System Details** to display the Details page.
5. On the Details page, select the **Enable Spacewalk Configuration Management** and **Enable Spacewalk Remote Commands** check boxes.
6. Click **Update Kickstart Distribution** to save your changes.

7.2 Enabling Remote Configuration in a Kickstart File

Note

The following applies if you upload a kickstart file into a profile.

If you want to be able to deploy configuration files and run commands remotely from the Spacewalk web interface:

- Include the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages for installation.
- Configure `rhn-actions-control` to run on the client system in the post-installation shell:

```
%post
rhn-actions-control --enable-all
%end
```

For more information, see the `rhn-actions-control(8)` manual page.

7.3 Enabling Remote Configuration Manually

Install and configure remote configuration manually as follows:

1. Log in as `root` on the client system.
2. Use the `yum` command to install the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages:

```
# yum install rhncfg rhncfg-actions rhncfg-client
```

3. Use the `rhn-actions-control` command to configure the remote actions that the client permits:

```
# rhn-actions-control --enable-all
```

The `--report` option lists the permitted remote actions, for example:

```
# rhn-actions-control --report
deploy is enabled
diff is enabled
upload is enabled
mtime_upload is enabled
run is enabled
```

For more information, see the rhn-actions-control(8) manual page.

7.4 Enabling Remote Configuration for Client Systems by Using the Spacewalk Web Interface

Note

If you want Spacewalk to install the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages automatically from a software channel, the channel label must contain the string `rhn-tools`, for example `ol7-spacewalk27-client-rhn-tools`.

To install and configure remote configuration for existing non-managed client systems, follow these steps:

1. In the Spacewalk web interface, do the following:
   a. Enable the software channel that contains the `rhncfg`, `rhncfg-actions`, and `rhncfg-client` packages for the client. See Chapter 9, *Updating Client Systems*.
   b. Go to **Configuration**, **Systems**, and then **Target Systems**.
   c. On the Target Systems page, select the client systems from the systems that are listed and click **Enable Spacewalk Configuration Management**.
2. Configure configuration management by running the following commands on each client system:
   a. Check for any queued pending actions.

```
# rhn_check
```

   b. Configure the remote actions that the client permits.

```
# rhn-actions-control --enable-all
```

The `--report` option lists the permitted remote actions.

```
# rhn-actions-control --report
deploy is enabled
diff is enabled
upload is enabled
mtime_upload is enabled
run is enabled
```

For more information, see the `rhn-actions-control(8)` man page.

Chapter 8 Configuring System Groups to Manage Client Systems

You can create system groups to perform the same actions on multiple client systems. Typically, a system group contains systems that have a common installation base, architecture, and profile, for example Oracle Linux 6 (x86_64) servers.

If you manage large numbers of systems, creating system groups is a effective way of applying errata, installing or upgrading packages, changing channel subscriptions, deploying configuration files, and reconfiguring kickstart provisioning with a minimum of effort.

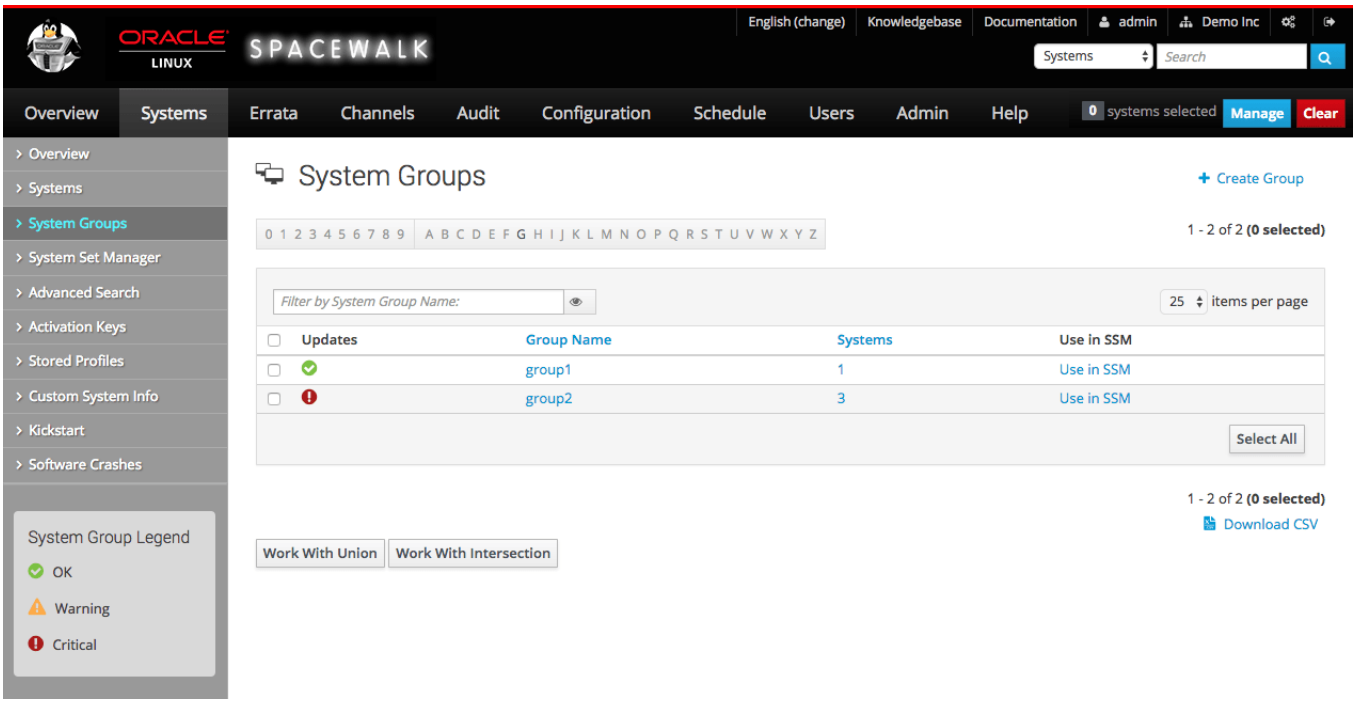
Spacewalk provides the *System Set Manager*, which maintains a current, working system group, or *system set*, to which you can add or remove systems and system groups. You can perform actions on the systems in the system set or you can save the system set as a new system group.

Note

If a system is present in the system set, the Spacewalk web interface has a check mark in its associated check box on the Systems page. You can select or deselect system check boxes to add or remove systems from a system set.

8.1 Working With System Groups by Using the Spacewalk Web Interface

Figure 8.1 System Groups Page



Select **Systems** and then **System Groups**:

- To create a system group:

  1. Click **+ create new group**.

2. On the Create System Group page, enter a name and description for the system group.
3. Click **Create Group**.

- To add client systems to a system group:
  1. Click the system group name.
  2. Select the **Target Systems** tab.
  3. On the Target Systems page, select the check boxes for the systems that you want to add to the group and click **Add Systems**.
- To work with a system group:
  1. Click the system group name.
  2. On the Details page, click **Work With Group**.

     Spacewalk loads the group into the System Set Manager.

     The Selected Systems List page under System Set Manager displays the member systems of the system group. Any actions that you take on the tabs under System Set Manager apply only to these systems.
- To work with the union or intersection of two or more system groups:
  1. Select the check boxes next to the system groups.
  2. Click either **Work With Union** or **Work With Intersection**.
     - **Work With Union** creates a union group that includes all member systems of the selected groups.
     - **Work With Intersection** creates an intersection group that includes only systems that are members of all of the selected groups. If no systems are members of all of the groups, the intersection group does not have any members.
     - The Selected Systems List page under System Set Manager displays the member systems of the union or intersection group. Any actions that you take on the tabs under System Set Manager apply only to these systems.
     - To save a union or intersection group as a new system group, select the Groups tab, click **+ create new group**, enter a name and description for the system group, and click **Create Group**.
- To remove client systems from a system group:
  1. Click the system group name.
  2. Select the **Systems** tab.
  3. On the Systems page, select the check boxes of the systems that you want to remove from the group and click **Remove Systems**.
- To delete a system group:
  1. Click the system group name.
  2. Click **delete group** and then click **Confirm Deletion**.

8.2 Working With System Groups by Using the spacecmd Command

Create a system group by using the `group_create` command as follows:

```
spacecmd {SSM:0}> group_create group3 "Example system group 3"
```

To list system groups, use the `group_list` command:

```
spacecmd {SSM:0}> group_list
group1
group2
group3
```

To add client systems to a system group, use the `group_addsystems` command:

```
spacecmd {SSM:0}> group_addsystems group3 svr1.mydom.com
```

You can also specify systems by the software channels to which they are subscribed or the results of a system search, as shown in the following example:

```
spacecmd {SSM:0}> group_addsystems group3 channel:ol6-x86_64
spacecmd {SSM:0}> group_addsystems group3 ip:192.168.1
```

See Section 8.3, “Searching for Systems by Using the spacecmd Command”.

To display the details of a system group, use the `group_details` command:

```
spacecmd {SSM:0}> group_details group3
Name               group3
Description:       Example system group 3
Number of Systems: 1

Members
-------
svr1.mydom.com
```

To work with a system group, specify it by using `group:`*`group_name`* to a `spacecmd` command:

```
spacecmd {SSM:0}> system_listerrata group:group2
System: svr1.mydom.com

Security Errata
---------------
ELSA-2017-1095 Important: bind security update 4/19/17
ELSA-2017-0907 Moderate: util-linux security and bug fix update 4/12/17
ELSA-2017-0906 Moderate: httpd security and bug fix update 4/12/17
ELSA-2017-0933 Important: kernel security, bug fix, and 4/12/17
...
```

To create a union of two or more system groups, create an empty group and specify the groups to the `group_addsystems` command:

```
spacecmd {SSM:0}> group_create group4 "Example system group 4"
spacecmd {SSM:0}> group_addsystems group4 group:group1 group:group2
```

To create an intersection of two or more system groups, clear the contents of the system set in the System Set Manager, use the `ssm_intersect` command to create the intersection as the new system set, create an empty group and specify the system set as `ssm` to the `group_addsystems` command:

```
spacecmd {SSM:0}> ssm_clear
spacecmd {SSM:0}> ssm_intersect group:group1 group:group2
spacecmd {SSM:2}> group_create group5 "Example system group 5"
spacecmd {SSM:2}> group_addsystems group5 ssm
```

```
spacecmd {SSM:2}> ssm_clear
spacecmd {SSM:0}>
```

Note

`{SSM:N}` shows the number of systems that are members of the system set.

To remove client systems from a system group, use the `group_removesystems` command:

```
spacecmd {SSM:0}> group_removesystems group3 svr1.mydom.com
Systems
-------
svr1.mydom.com

Remove these systems [y/N]: y
```

To delete a system group, use the `group_delete` command:

```
spacecmd {SSM:0}> group_delete group3
group3

Delete these groups [y/N]: y
```

8.3 Searching for Systems by Using the spacecmd Command

Search for systems by using the `system_search` command as follows:

```
spacecmd {SSM:0}> system_search criterion:value
```

You can search on the following criteria:

`device`	System device name, for example, `"xen platform device"`.
`driver`	System driver name, for example, `ata_piix`.
`hostname`	FQDN of the system, for example, `svr1.mydom.com`.
`id`	System ID in Spacewalk, for example, `1000010100`.
`ip`	IP address, for example, `192.168.1`.
`name`	System name in Spacewalk, for example, `svr1.mydom.com`.
`uuid`	System UUID, for example, `0004fb0000060000a4d43e4f737f4f5d`.
`vendor`	System vendor name, for example, `GenuineIntel`.

For example, you would search for systems that have an IP address that contains `192.168.1` as follows:

```
spacecmd {SSM:0}> system_search ip:192.168.1
svr1.mydom.com    192.168.1.201
svr2.mydom.com    192.168.1.202
...
```

You can also use a `search` query instead of a system name with `spacecmd` commands, as shown in this example:

```
spacecmd {SSM:0}> group_addsystems group3 search:ip:192.168.1
```

To search for systems that subscribe to a software channel, use the `softwarechannel_listsystems` command:

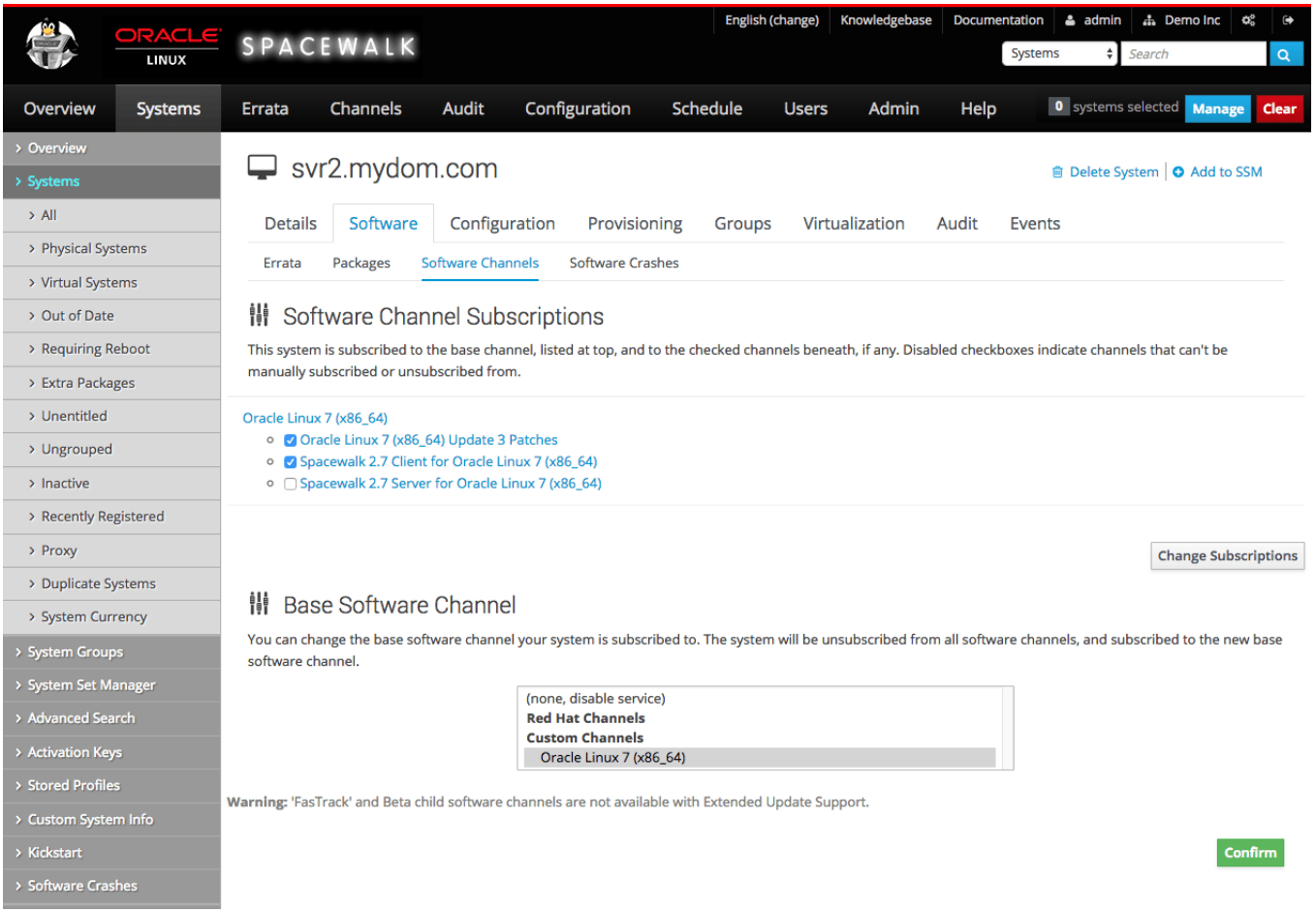
```
spacecmd {SSM:0}> softwarechannel_listsystems ol6-x86_64
svr1.mydom.com
svr2.mydom.com
...
```

Chapter 9 Updating Client Systems

You can use either the Spacewalk web interface or the `spacecmd` command on the Spacewalk server to subscribe client systems to software channels and update client systems by using these channels. Alternatively, you can use the `spacewalk-channel` command on an individual Spacewalk client.

9.1 Subscribing Client Systems to Software Channels by Using the Spacewalk Web Interface

Figure 9.1 Software Channel Subscriptions Page



To subscribe systems to software channels:

1. Go to **Systems** and click the system name.
2. Select **Software** and then select the **Software Channels** tab.
3. Change the child or base software channels:
   - To change the child software channels to which a system is subscribed:
     a. In the Software Channel Subscriptions section, select or deselect the check boxes next to the child software channels to which you want to want to subscribe or unsubscribe the client.

b. Click **Change Subscriptions**.

- To change the base software channel to which a system is subscribed:

  a. In the Base Software Channel section, select the new base software channel.

  b. Click **Confirm**.

  c. On the Confirm Base Software Channel page, click **Modify Base Software Change**.

Note

Changing the base software channel unsubscribes a system from all other software channels.

9.2 Subscribing Client Systems to Software Channels by Using the spacecmd Command

To list the base and child software channels to which a system is subscribed, use the `system_listbasechannel` and `system_listchildchannels` commands, as shown in the following example:

```
spacecmd {SSM:0}> system_listbasechannel svr1.mydom.com
ol6-x86_64
spacecmd {SSM:0}> system_listchildchannels svr1.mydom.com
ksplice-ol6-x86_64
ol6_x86_64_addons
ol6_x86_64_spacewalk27_client
ol6_x86_64_uekr3_latest
```

To list the available child channels of a base channel, use the `softwarechannel_listchildchannels` command:

```
spacecmd {SSM:0}> softwarechannel_listchildchannels oraclelinux6-x86_64
ksplice-ol6-x86_64
ol6_x86_64_addons
ol6_x86_64_playground
ol6_x86_64_spacewalk27_client
ol6_x86_64_spacewalk27_server
ol6_x86_64_uekr3_latest
```

To add or remove child channels, use the `system_addchildchannels` and `system_removechildchannels` commands, as shown in this example:

```
spacecmd {SSM:0}> system_removechildchannels svr1.mydom.com ol6_x86_64_addons
Systems
-------
svr1.mydom.com

Removing Channels
-----------------
ol6_x86_64_addons

Is this ok [y/N]: y

spacecmd {SSM:0}> system_addchildchannels svr2.mydom.com ol6_x86_64_playground
Systems
-------
svr2.mydom.com
```

```
Adding Channels
---------------
ol6_x86_64_playground

Is this ok [y/N]: y
```

To list the available base channels, use the `softwarechannel_listbasechannels` command:

```
spacecmd {SSM:0}> softwarechannel_listbasechannels
oraclelinux7u3-x86_64
oraclelinux7u4-x86_64
```

To change the base channel to which a system is subscribed, use the `system_setbasechannel` command:

```
spacecmd {SSM:0}> system_setbasechannel svr5.mydom.com oraclelinux7u4-x86_64
System:           svr5.mydom.com
Old Base Channel: oraclelinux7u3-x86_64
New Base Channel: oraclelinux7u4-x86_64

Is this ok [y/N]: y
```

Note

Changing the base software channel unsubscribes a system from all other software channels.

You can change the subscribed channels for multiple systems by specifying the following arguments in place of a system name:

`channel:`*`channel_name`*

Matches systems that are subscribed to the specified software channel.

`group:`*`group_name`*

Specifies the systems in the named system group, for example, the following:

```
spacecmd {SSM:0}> system_removechildchannels group:group3 ol6_x86_64_playground
Systems
-------
svr1.mydom.com
svr2.mydom.com

Removing Channels
-----------------
ol6_x86_64_playground

Is this ok [y/N]: y
```

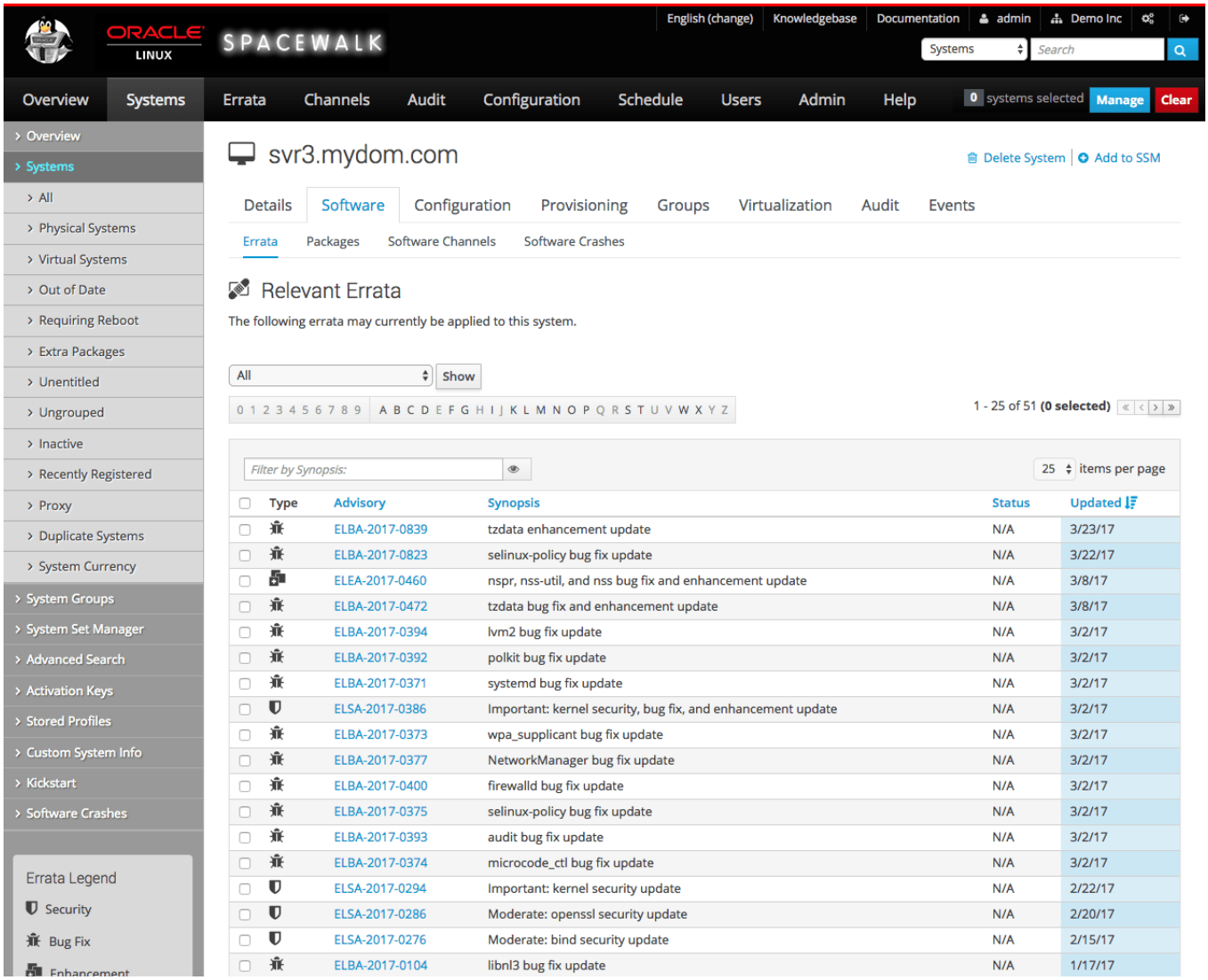
`search:`*`criterion`*`:`*`value`*

Matches systems that match a search criterion. See Section 8.3, “Searching for Systems by Using the spacecmd Command”.

`ssm`: Specifies the systems that are currently in the system set.

9.3 Listing and Applying Available Security Updates and Errata by Using the Spacewalk Web Interface

Figure 9.2 Relevant Errata Page



To list the available security updates and other errata for systems or system groups:

1. For systems:

   a. Go to **Systems** and click the system name.

   b. Select **Software** and then select the **Errata** tab.

      Alternatively, click **Critical** or **Non-Critical** in the System Status pane to display the Relevant Errata page with security advisory or non-critical errata selected for display.

   For system groups:

   a. Go to **System Groups** and click the system group name.

b. On the **Details** page, click **work with group**.

   Spacewalk loads the group into the System Set Manager.

c. In the System Set Manager, select the **Errata** tab.

2. On the Relevant Errata List page, select **All**, **Non-Critical**, **Bug Fix Advisory**, **Product Enhancement Advisory**, or **Security Advisory** from the pull-down list and click **Show**.

   - You can filter the list on the Synopsis value or sort the list by clicking **Advisory**, **Synopsis**, **Status**, **Affected** (system groups only), or **Updated**.
   - To see more details about an erratum listed under **Advisory**, select its name.

     The CVEs section lists the CVEs that are fixed by an erratum. Click on a CVE name for more details.
   - To display the packages that are affected by an erratum, select the **Packages** tab.
   - To display the systems to which you can apply the erratum, select the **Affected Systems** tab.

3. To apply errata to systems or system groups:

   a. Select the check boxes for the errata that you want to apply, or click **Select All** to select all of the listed errata.

   b. Click **Apply Errata**.

   c. On the Relevant Errata Confirm page, change the schedule if required, and click **Confirm**.

      The page updates to include a link to the scheduled action.

      If you have not edited the schedule and you have enabled the OSA daemon on the client, the OSA daemon usually installs the errata packages immediately. Otherwise, `rhnsd` applies the errata when it next runs on the client.

   d. Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the status and details of the errata update on the client.

9.4 Listing and Applying Available Security Updates and Errata by Using the spacecmd Command

To list the security, bug fix, and product-enhancement advisory errata that are available for a client system, use the `system_listerrata` command:

```
spacecmd {SSM:0}> system_listerrata svr1.mydom.com
Security Errata
---------------
ELSA-2017-1095 Important: bind security update 4/19/17
ELSA-2017-0907 Moderate: util-linux security and bug fix update 4/12/17
ELSA-2017-0906 Moderate: httpd security and bug fix update 4/12/17
ELSA-2017-0933 Important: kernel security, bug fix, and 4/12/17

Bug Fix Errata
--------------
ELBA-2017-0992 lvm2 bug fix update 4/18/17
ELBA-2017-0928 NetworkManager bug fix update 4/13/17
ELBA-2017-0910 libtool bug fix update 4/12/17
ELBA-2017-0915 openssh bug fix update 4/12/17
```

```
ELBA-2017-0913 ca-certificates bug fix and enhancement update 4/12/17
ELBA-2017-0921 grubby bug fix update 4/12/17
ELBA-2017-0923 squid bug fix update 4/12/17
ELBA-2017-0905 irqbalance bug fix update 4/12/17
ELBA-2017-0909 systemd bug fix update 4/12/17
ELBA-2017-0904 dmidecode bug fix update 4/12/17
ELBA-2017-0911 initscripts bug fix update 4/12/17
```

To find out more details about an erratum, use the `errata_details` command:

```
spacecmd {SSM:0}> errata_details ELSA-2017-1095
Name: ELSA-2017-1095
Product: Oracle Linux 7
Type: Security Advisory
Issue Date: 4/19/17

Topic
-----


Description
-----------
[32:9.9.4-38.3] - Fix CVE-2017-3136 (ISC change 4575) - Fix
CVE-2017-3137 (ISC change 4578)

CVEs
----
CVE-2017-3136
CVE-2017-3137

Solution
--------
This update is available via the Unbreakable Linux Network (ULN) and
the Oracle Yum Server Server. Details on how to use ULN or http
://yum.oracle.com to apply this update are available at
https://linux.oracle.com/applying_updates.html.

References
----------
https://linux.oracle.com/errata/ELSA-2017-1095.html

Affected Channels
-----------------
ol7-x86_64-u3-patch

Affected Systems
----------------
3

Affected Packages
-----------------
bind-9.9.4-38.el7_3.3:32.x86_64
bind-chroot-9.9.4-38.el7_3.3:32.x86_64
bind-libs-9.9.4-38.el7_3.3:32.i686
bind-libs-9.9.4-38.el7_3.3:32.x86_64
bind-libs-lite-9.9.4-38.el7_3.3:32.i686
bind-libs-lite-9.9.4-38.el7_3.3:32.x86_64
bind-license-9.9.4-38.el7_3.3:32.noarch
bind-pkcs11-9.9.4-38.el7_3.3:32.x86_64
bind-pkcs11-libs-9.9.4-38.el7_3.3:32.i686
bind-pkcs11-libs-9.9.4-38.el7_3.3:32.x86_64
bind-pkcs11-utils-9.9.4-38.el7_3.3:32.x86_64
bind-utils-9.9.4-38.el7_3.3:32.x86_64
```

To find the errata that fix a CVE, use the `errata_findbycve` command:

```
spacecmd {SSM:0}> errata_findbycve CVE-2017-3136
```

```
CVE-2017-3136:
ELSA-2017-1095
```

To list the systems to which you could apply an erratum, use the `errata_listaffectedsystems` command:

```
spacecmd {SSM:0}> errata_listaffectedsystems ELSA-2017-1095
ELSA-2017-1095:
svr1.mydom.com
svr2.mydom.com
svr3.mydom.com
```

To apply an erratum to a system, use the `system_applyerrata` command:

```
spacecmd {SSM:0}> system_applyerrata svr1.mydom.com ELSA-2017-1095
Start Time [now]:
Errata Systems
-------------- -------
ELSA-2017-1095 1

Start Time: 20170421T10:01:40

Apply these errata [y/N]: y
INFO: Scheduled 1 system(s) for ELSA-2017-1095
```

You can apply errata to multiple systems by specifying the following arguments in place of a system name:

`channel:`*`channel_name`*

Matches systems that are subscribed to the specified software channel.

`group:`*`group_name`*

Specifies the systems in the named system group.

`search:`*`criterion`*`:`*`value`*

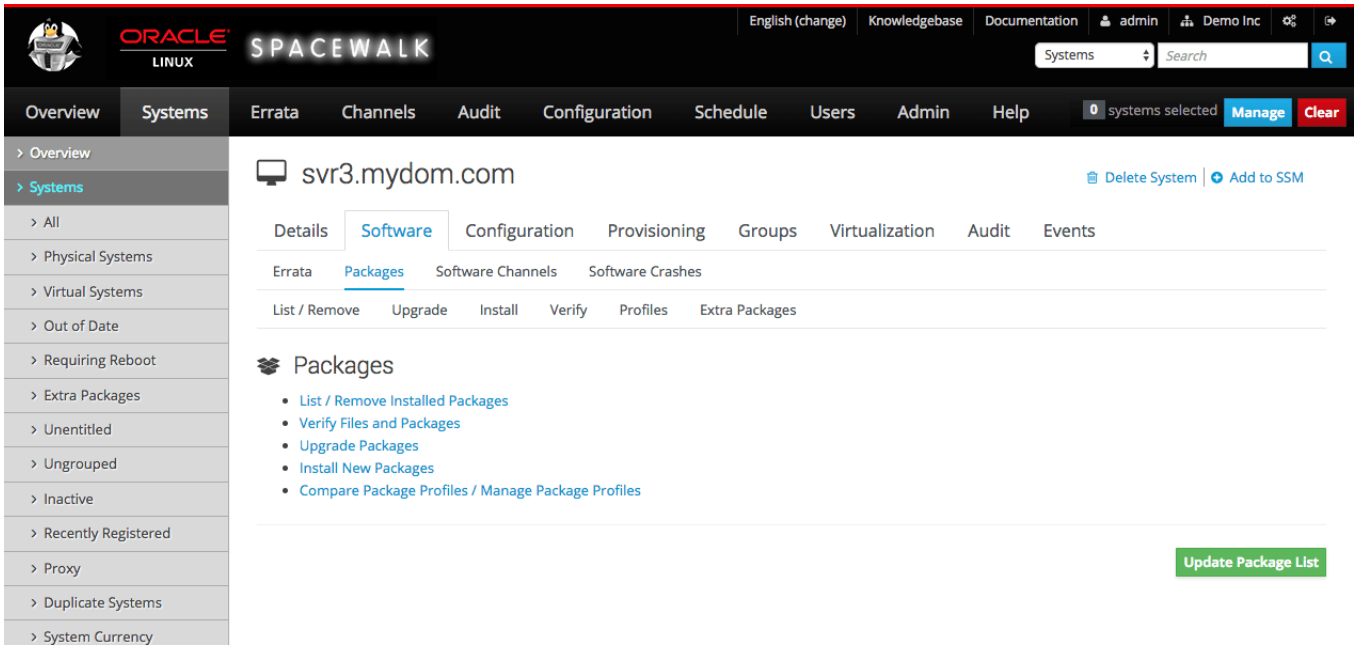
Matches systems that match a search criterion. See Section 8.3, “Searching for Systems by Using the spacecmd Command”.

`ssm`: Specifies the systems that are currently in the system set, for example, the following:

```
spacecmd {SSM:0}> ssm_add svr2.mydom.com svr3.mydom.com
spacecmd {SSM:2}> system_applyerrata ssm ELSA-2017-1095
Start Time [now]:
Errata Systems
-------------- -------
ELSA-2017-1095 2

Start Time: 20170421T10:02:23

Apply these errata [y/N]: y
INFO: Scheduled 2 system(s) for ELSA-2017-1095
```

9.5 Managing Packages for Systems by Using the Spacewalk Web Interface

Figure 9.3 Packages Page



To manage packages for a system:

1. Go to **Systems** and click the system name.
2. Select **Software**.
3. On the Packages page, select the tab or link for the package operation that you want to perform:

   Extra Packages The Extra Packages page displays packages that are installed on a system, but which are not present in any of the subscribed channels.

   Note

   If you registered an existing system, such as the Spacewalk server itself, as a client, it is possible that some of the installed packages are not present in any subscribed channel.

   If the Spacewalk server is a client of itself, Oracle recommends that you synchronize the Spacewalk Server repository and associate it with the server so that the server receives Spacewalk Server software updates.

   If one or more packages should not have been installed on a system:

   a. Select the packages that you want to remove and click **Remove Packages**.

   b. On the Confirm Package Removal page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually removes the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package removals.

Install

a. On the Installable Packages page, select the packages that you want to install and click **Install Selected Packages**.

Tip

Use the package filter to locate a package.

To see more information about a package, click its name.

The Details page for the package lists any errata that include the package. To find out more information about an erratum, click its name.

The Details page for the erratum lists the CVEs that the erratum fixes. To find out more information about a CVE, click its name.

b. On the Confirm Package Install page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually installs the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package installations.

List/Remove

a. On the Removable Packages page, select the packages that you want to remove and click **Remove Packages**.

b. On the Confirm Package Removal page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually removes the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package removals.

Profiles

On the Profiles page, you can do the following:

- Create a package profile from the set of packages that are currently installed on the system:

  a. Click **Create System Profile**.

  b. On the Create Stored Profile page, enter a name and description for the profile and then click **Create Profile**.

- Compare the packages installed on this system with a stored package profile for this system or for another system:

  - In the Compare to Stored Profile section, select the profile name from the pull-down list and click **Compare**.

- Compare the packages installed on this system with those installed on another system:

  - In the Compare to System section, select the system name from the pull-down list and click **Compare**.

Upgrade

a. On the Upgradable Packages page, select the packages that you want to upgrade and click **Upgrade Packages**.

b. On the Confirm Package Upgrade page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually upgrades the packages immediately.

Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package upgrades.

Verify

a. On the Verifiable Packages page, select the packages that you want to verify and click **Verify Selected Packages**.

b. On the Confirm Package Verification page, change the schedule if required, and click **Confirm**.

   The page updates to include a link to the scheduled action.

   If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually verifies the packages immediately.

   Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details and status of the package verifications.

9.6 Managing Packages for Systems by Using the spacecmd Command

Create a package profile from the set of packages that are currently installed on a system by using the `system_createpackageprofile` command, for example:

```
spacecmd {SSM:0}> system_createpackageprofile svr1.mydom.com -n svr1-profile1 -d "svr1 profile 1"
INFO: Created package profile 'svr1-profile1'
```

To compare the packages installed on this system with a stored package profile for this system or for another system, use the `system_comparepackageprofile` command as follows:

```
spacecmd {SSM:0}> spacecmd {SSM:0}> system_comparepackageprofile svr2.mydom.com svr1-profile1
svr2.mydom.com:
Package  This System  Other System  Difference
-------  -----------  ------------  ----------
zsh      4.3.10-9.el6 None          Only here
```

To compare the packages installed on this system with those installed on another system, use the `system_comparepackageprofile` command:

```
spacecmd {SSM:0}> system_comparepackages svr1.mydom.com svr2.mydom.com
svr2.mydom.com:
Package  This System      Other System  Difference
-------  -----------      ------------  ----------
ypbind   1.20.4-30.el6:3  None          Only here
zsh      None             4.3.10-9.el6  Only there
```

To display the details of an installable package, use the `package_details` command:

```
spacecmd {SSM:0}> package_details zsh
...
Name:    zsh
Version: 4.3.10
Release: 9.el6
Epoch:
Arch:    x86_64

File:    zsh-4.3.10-9.el6.x86_64.rpm
Path:    redhat/1/f5a/zsh/4.3.10-9.el6/x86_64/f5a...59c/zsh-4.3.10-9.el6.x86_64.rpm
Size:    2238632
MD5:     None

Installed Systems: 1

Description
-----------
```

```
The zsh shell is a command interpreter usable as an interactive login
shell and as a shell script command processor.  Zsh resembles the ksh
shell (the Korn shell), but includes many enhancements.  Zsh supports
command line editing, built-in spelling correction, programmable
command completion, shell functions (with autoloading), a history
mechanism, and more.

Available From Channels
-----------------------
oraclelinux6-x86_64
...
```

To install a package on a system, use the `system_installpackage` command:

```
spacecmd {SSM:0}> system_installpackage svr1.mydom.com zsh
svr1.mydom.com:
** Generating package cache **
zsh-4.3.10-9.el6.x86_64

Install these packages [y/N]: y
INFO: Scheduled 1 system(s)
spacecmd {SSM:0}> schedule_list
ID      Date                C    F    P     Action
--      ----                ---  ---  ---   ------
401     20150618T15:22:51     0    0    1    Package Install
...
spacecmd {SSM:0}> schedule_details 401
ID:        401
Action:    Package Install
User:      swadmin
Date:      20150618T15:22:51

Completed:   0
Failed:      0
Pending:     1

Pending Systems
---------------
svr1.mydom.com
```

To list the package upgrades that are available for a system, use the `system_listupgrades` command:

```
spacecmd {SSM:0}> system_listupgrades svr1.mydom.com
bash-4.1.2-29.el6.0.1.x86_64
wget-1.12-5.el6_6.1.x86_64
```

To upgrade the packages on a system, use the `system_upgradepackage` command:

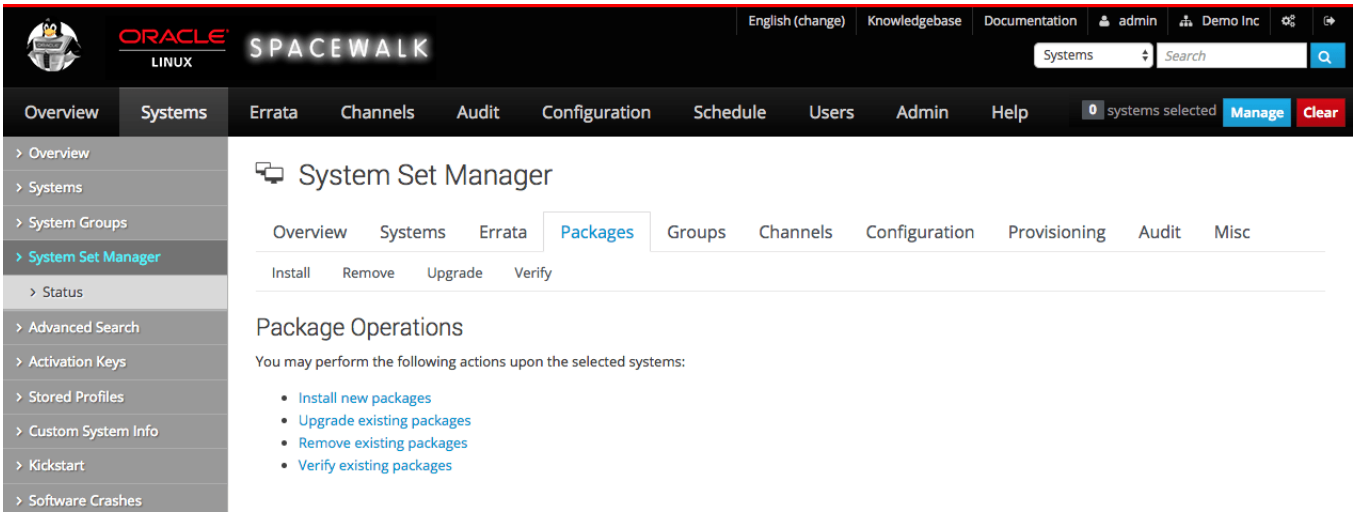
```
spacecmd {SSM:0}> system_upgradepackage svr1.mydom.com *
svr1.mydom.com:
bash-4.1.2-29.el6.0.1.x86_64
wget-1.12-5.el6_6.1.x86_64

Install these packages [y/N]: y
INFO: Scheduled 1 system(s)
```

To remove a package from a system, use the `system_removepackage` command:

```
spacecmd {SSM:0}> system_removepackage svr1.mydom.com busybox*
svr1.mydom.com:
busybox-1.15.1-20.el6:1.x86_64

Remove these packages [y/N]: y
INFO: Action ID: 403
INFO: Scheduled 1 system(s)
```

9.7 Managing Packages for System Groups by Using the Spacewalk Web Interface

Figure 9.4 Package Operations Page



To manage packages for system groups, follow these steps:

1. Go to **System Groups** and click the system group name.
2. On the **Details** page, click **work with group**.

   Spacewalk loads the group into the System Set Manager.
3. In the System Set Manager, select the **Packages** tab.
4. On the Package Operations page, select the tab or link for the package operation that you want to perform:

   Install

   a. On the Select Channel page, select the channel that contains the packages that you want to install on the systems in the system group.

   b. On the Select Packages to Install page, select the packages that you want to install and click **Install Selected Packages**.

   Tip

   Use the package filter to locate a package.

   To see more information about a package, click its name.

   The Details page for the package lists any errata that include the package. To find out more information about an erratum, click its name.

   The Details page for the erratum lists the CVEs that the erratum fixes. To find out more information about a CVE, click its name.

   c. On the Confirm Package Install page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually installs the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package installations.

Remove

a. On the Package Removal page, select the packages that you want to remove and click **Remove Selected Packages**.

b. On the Confirm Package Removal page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually removes the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package removals.

Upgrade

a. On the Select Packages to Upgrade page, select the packages that you want to upgrade and click **Upgrade Selected Packages**.

b. On the Confirm Package Upgrade page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually upgrades the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package upgrades.

Verify

a. On the Verifiable Packages page, select the packages that you want to verify and click **Verify Selected Packages**.

b. On the Confirm Package Verification page, change the schedule if required, and click **Confirm**.

The page updates to include a link to the scheduled action.

If you have not edited the schedule and you have enabled the OSA daemon on the clients, the OSA daemon usually verifies the packages immediately.

The Tasks Log page in the System Set Manager shows the status of the package verifications.

9.8 Managing Packages for System Groups by Using the spacecmd Command

Compare the packages that are installed on the systems in a system group with a stored package profile by using the `system_comparepackageprofile` command, as shown in the following example:

```
spacecmd {SSM:0}> system_comparepackageprofile group:group1 svr1-profile1
svr3.mydom.com:
Package  This System  Other System  Difference
-------  -----------  ------------  ----------
zsh      None         4.3.10-9.el6  Only there

##############################

svr4.mydom.com:
Package  This System  Other System  Difference
-------  -----------  ------------  ----------
zsh      None         4.3.10-9.el6  Only there
```

To install a package on the systems in a system group, use the `system_installpackage` command as follows:

```
spacecmd {SSM:0}> system_installpackage group:group1 zsh
svr3.mydom.com:
zsh-4.3.10-9.el6.x86_64

##############################

svr4.mydom.com:
zsh-4.3.10-9.el6.x86_64

Install these packages [y/N]: y
INFO: Scheduled 2 system(s)
```

To list the package upgrades that are available for the systems in a system group, use the `system_listupgrades` command:

```
spacecmd {SSM:0}> system_listupgrades group:group1
svr3.mydom.com:
bash-4.1.2-29.el6.0.1.x86_64
wget-1.12-5.el6_6.1.x86_64

##############################

svr4.mydom.com:
wget-1.12-5.el6_6.1.x86_64
```

To upgrade the packages on the systems in a system group, use the `system_upgradepackage` command:

```
spacecmd {SSM:0}> system_upgradepackage group:group1 *
svr3.mydom.com:
bash-4.1.2-29.el6.0.1.x86_64
wget-1.12-5.el6_6.1.x86_64

##############################

svr4.mydom.com:
wget-1.12-5.el6_6.1.x86_64

Install these packages [y/N]: y
INFO: Scheduled 2 system(s)
```

To remove a package from the systems in a system group, use the `system_removepackage` command:

```
spacecmd {SSM:0}> system_removepackage svr1.mydom.com busybox*
svr3.mydom.com:
busybox-1.15.1-20.el6:1.x86_64

##############################
```

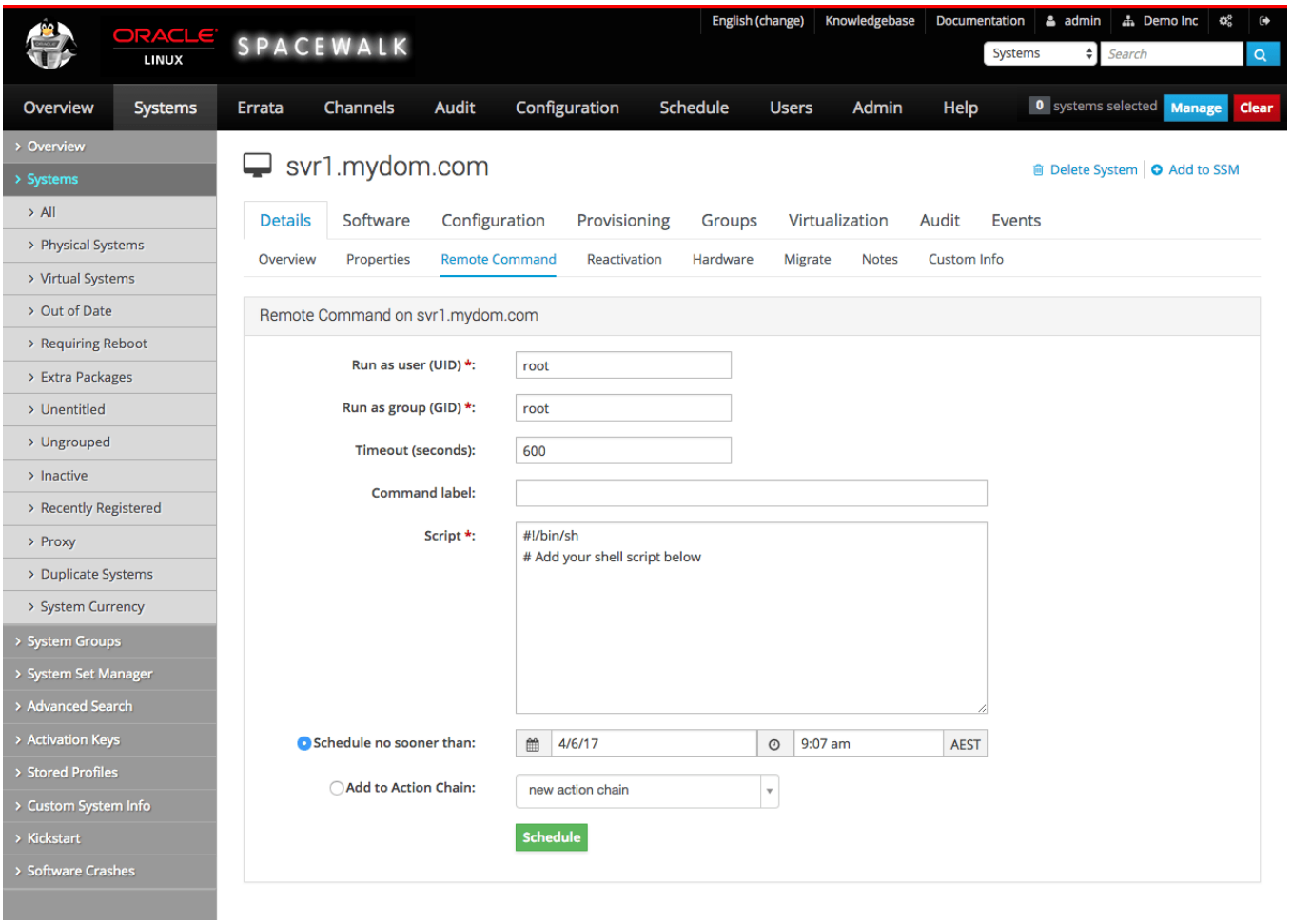
```
svr4.mydom.com:
busybox-1.15.1-20.el6:1.x86_64

Remove these packages [y/N]: y
INFO: Action ID: 407
INFO: Action ID: 408
INFO: Scheduled 2 system(s)
```

Chapter 10 Controlling and Configuring Client Systems

You can use either the Spacewalk web interface or the `spacecmd` command to run command scripts on remote client systems. You can also set up configuration channels, subscribe client systems to these channels, and customize client systems by using the channels to deploy configuration files.

10.1 Running Command Scripts on Remote Client Systems by Using the Spacewalk Web Interface

Figure 10.1 Remote Command Page



Note

The client system must permit the Spacewalk server to run remote commands. See Section 7.1, “Enabling Remote Configuration in a Kickstart Profile by Using the Spacewalk Web Interface” and Section 7.3, “Enabling Remote Configuration Manually”.

To run a command on a remote client:

1. Go to **Systems** and select the client system from the list.
2. Select **Details** and then select the **Remote Command** tab.

3. If required, change the user and group ID of the user that should run the command, the command timeout, and a command label of up to 10 characters.

4. In the **Script** text box, enter the command script that you want to run. The following example runs the `who` command:

```
#!/bin/sh
who
```

5. If required, change the schedule for the command.

6. Click **Schedule** to commit the command script to run according to the schedule that you specify.

   If you have not edited the schedule and you have enabled the OSA daemon on the client, the OSA daemon usually runs the command immediately.

7. Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details of the script and any output if it has already run on the client.

10.2 Running Command Scripts on Remote Client Systems by Using the spacecmd Command

Note

The client system must permit the Spacewalk Server to run remote commands. See Section 7.1, “Enabling Remote Configuration in a Kickstart Profile by Using the Spacewalk Web Interface” and Section 7.3, “Enabling Remote Configuration Manually”.

To run a command on client systems, use the `system_runscript` command, as shown in the following example:

```
spacecmd {SSM:0}> system_runscript group:group3 -s 20150617T0130 -t 60 -f /root/myscript

User:       root
Group:      root
Timeout:    60 seconds
Start Time: 20150617T01:30:00

Script Contents
---------------
#!/bin/sh
yum update

Systems
-------
svr2.mydom.com
svr3.mydom.com

Is this ok [y/N]: y
INFO: Action ID: 343
INFO: Scheduled: 2 system(s)
```

To specify the date and time when an event should start, use the format *YYYYMMDD*[*hhmm*] with the `-s` option. If you do not specify a start time, Spacewalk assumes `0000` (midnight).

The `-t` option defines a timeout for a client to confirm that it has run a command. After this time has elapsed, Spacewalk assumes that the command has failed.

10.3 Working With Scheduled Events

To display a list of completed, failed, and pending events, use the `schedule_list` command:

```
spacecmd {SSM:0}> schedule_list
ID      Date                 C    F    P     Action
--      ----                ---  ---  ---    ------
...
343     20150617T01:30:00     0    0    2    Run an arbitrary script
...
```

The `C`, `F`, and `P` columns show the number of systems on which the event has completed, failed, or is pending. To display only completed, failed, or pending events, use the `schedule_listcompleted`, `schedule_listfailed`, or `schedule_listpending` commands.

To display the details of a pending event, use the `schedule_details` command:

```
spacecmd {SSM:0}> schedule_details 343
ID:        343
Action:    Run an arbitrary script
User:      swadmin
Date:      20150617T01:30:00

Completed:   0
Failed:      0
Pending:     2

Pending Systems
---------------
svr2.mydom.com
svr3.mydom.com
```

To cancel a pending event, use the `schedule_cancel` command:

```
spacecmd {SSM:0}> schedule_cancel 343
INFO: Canceled action 343
Canceled 1 action(s)
```

To re-run a failed event, use the `schedule_reschedule` command:

```
spacecmd {SSM:0}> schedule_reschedule 382
Rescheduled 1 action(s)
```

10.4 Working With Configuration Channels

Note

The client system must permit the Spacewalk server to deploy files. See Section 7.1, “Enabling Remote Configuration in a Kickstart Profile by Using the Spacewalk Web Interface”, Section 7.3, “Enabling Remote Configuration Manually”, and Chapter 3, *Working With Activation Keys in Spacewalk*.

In the same way that a software channel in Spacewalk contains packages for installation on multiple client systems, a configuration channel contains files for configuring client systems. For example, the files might contain configuration information for services, applications, or users.

10.4.1 Using Custom Information Keys

Custom information keys enable you to extract configuration information from clients.

To use a custom information key, follow these steps:

1. Define the custom information key in Spacewalk as described in Section 10.4.2, “Defining Custom Information Keys by Using the Spacewalk Web Interface” and Section 10.4.3, “Defining Custom Information Keys by Using the spacecmd Command”.

   For example, you could define a custom information key named `uptrack-uname` to store the value of the Ksplice effective kernel version.

2. Install the `rhn-custom-info` package on each client with which you want to use the key.

3. Use the `rhn-custom-info` command to make the value of the key available in Spacewalk:

   ```
   # rhn-custom-info uptrack-uname `uptrack-uname -r`
   ```

   This command makes the value returned by `uptrack-uname -r` available as the value of the `uptrack-uname` key. You can then use the macro `rhn.system.custom_info(uptrack-uname)` to extract the value of `uptrack-name` within a configuration file.

10.4.2 Defining Custom Information Keys by Using the Spacewalk Web Interface

To define a custom information key and assign it to a system, follow these steps:

1. Go to **Systems** and select **Custom System Info**.
2. Click **+ create new key**.
3. On the Create Custom Info Key page, enter a key label (for example, `asset_tag`) and description, and then click **Create Key**.
4. Go to **Systems** and click the name of the system for which you want to assign a value to the key.
5. Select the **Custom Info** tab.
6. On the Custom System Information page, click **+ create new value**.
7. On the Edit Custom Info Key page, select the key to which you want to assign a value.

   The page updates to display information about the key and a **Value** text box.

8. Enter the key value in the **Value** text box and click **Update Key**.

   The Custom System Information page displays the key-value pairs that are associated with a system. You can modify a value by selecting the associated **Edit this value** link.

Note

You can also define custom information keys for a system by using the **Custom Info** tab of a system's Kickstart profile.

You can define as many key-value pairs for a system as you require.

10.4.3 Defining Custom Information Keys by Using the spacecmd Command

To create a custom information key, use the `custominfo_createkey` command as follows:

```
spacecmd {SSM:0}> custominfo_createkey admin_user "Email of admin contact"
```

To list the available custom information keys, use the `custominfo_listkeys` command:

```
spacecmd {SSM:0}> custominfo_listkeys
```

```
asset_tag
admin_user
```

To assign a custom information key to a system or system group, use the `system_addcustomvalue`:

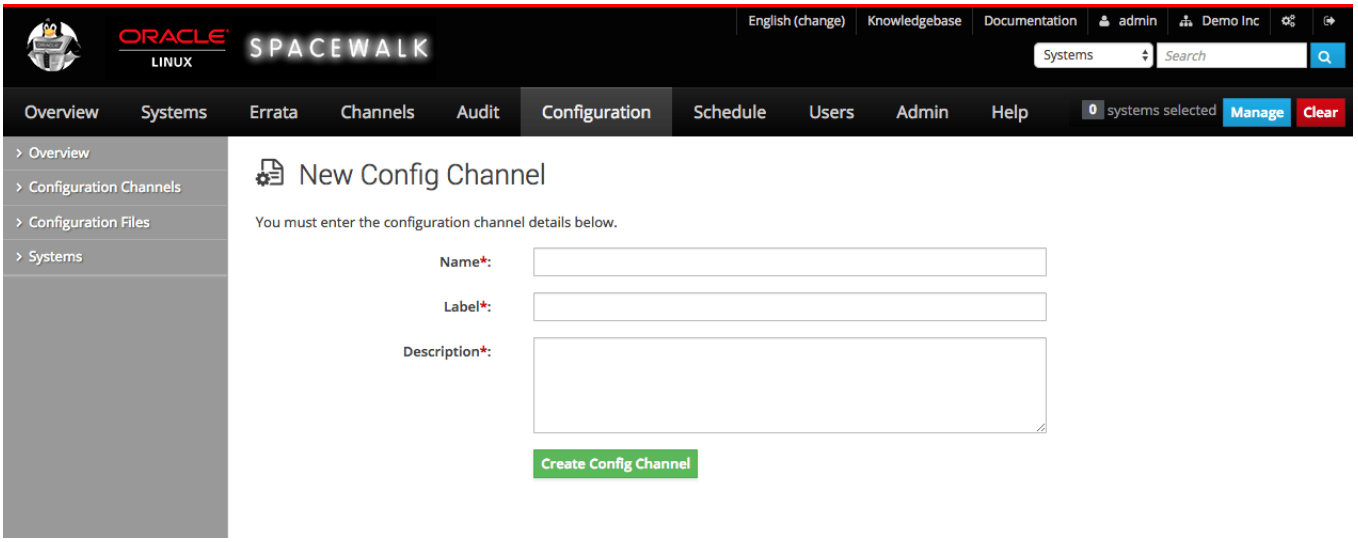
```
spacecmd {SSM:0}> system_addcustomvalue asset_tag "fc01568a" svr3.mydom.com
spacecmd {SSM:0}> system_addcustomvalue admin_user "an.admin@mydom.com" group:group3
```

To list the custom information keys for a system or system group, use the `system_listcustomvalues` command:

```
spacecmd {SSM:0}> system_listcustomvalues svr3.mydom.com
asset_tag = fc01568a
admin_user = an.admin@mydom.com
spacecmd {SSM:0}> system_listcustomvalues group:group3
System: svr3.mydom.com

asset_tag = fc01568a
admin_user = an.admin@mydom.com

##############################

System: svr4.mydom.com

asset_tag = aa10889f
admin_user = an.admin@mydom.com
```

10.4.4 Working With Configuration Channels by Using the Spacewalk Web Interface

Figure 10.2 New Config Channel Page



To create a configuration channel:

1. Go to **Configuration** and select **Configuration Channels**
2. On the Centrally Managed Configuration Channels page, click **+ create new config channel**.
3. Enter a name, label, and description for the channel. The label should be a short representation of the target operating system, architecture (if appropriate), and the purpose of the channel, for example, `ol6_generic_configuration`.
4. Click **Create Config Channel**.

5. To add files to the configuration channel, on the New Channel page, select the **Add Files** tab.
   - To create a text file, directory, or symbolic link:
     a. Select the **Create File** tab.
     b. On the Create New Configuration File page, you can create a text file, directory, or symbolic link.

        For example, to set up a message-of-the-day file that contains configuration information about the client, you might enter the following details:

File Type	Select **Text file**.
Filename/Path	Enter `/etc/motd`.
Ownership	Enter `root` for both the user name and group. (These are the default entries.)
File Permissions Mode	Enter 644. (This is the default mode.)
File Contents	Select the file type as **Shell** from the drop-down list, and enter the file contents in the text field.

        For example, the following file uses macros that Spacewalk replaces with the appropriate values for the system on which the file is deployed:

```
System Information
==================
Client system: {|rhn.system.hostname|}
Spacewalk SID: {|rhn.system.sid|}
Asset tag:     {|rhn.system.custom_info(asset_tag) = 'Asset tag missing'|}
Profile:       {|rhn.system.profile_name|}
Description:   {|rhn.system.description|}
IP address:    {|rhn.system.ip_address(eth0)|}
```

```
MAC address:   {|rhn.system.net_interface.hardware_address(eth0)|}
```

The custom macro `rhn.system.custom_info` substitutes the value of the custom system information key named `asset_tag` as assigned on the Custom Info tab for the system. Otherwise, it inserts the value `Asset tag missing`.

See Section 10.4.2, “Defining Custom Information Keys by Using the Spacewalk Web Interface”.

See Appendix C, *Configuration File Macros*.

c. After entering the details of the file, click **Create Configuration File**.

- To import files:

  a. Select the **Import Files** tab.

  On the Import Configuration File(s) from Another Channel page, you can import configuration files from other configuration channels.

  b. Select the check boxes of the configuration files that you want to import.

  c. Click **Import Configuration File(s)**.

- To upload files:

  a. Select the **Upload File** tab.

  b. On the Upload New Configuration File page, click **Browse...** and select the path of the file to upload.

  c. Select the file type: **Text file** or **Binary file**.

  d. Enter other details for the file, such as ownership and permissions, as required.

  e. Click **Upload Configuration File**.

10.4.5 Working With Configuration Channels by Using the spacecmd Command

To create a configuration channel, use the `configchannel_create` command, as shown in the following example:

```
spacecmd {SSM:0}> configchannel_create
Name: Oracle Linux 6 Server Configuration
Label: ol6-server-config
Description: Generic configuration channel for Oracle Linux 6 servers
```

To add a configuration file to a channel, use the `configchannel_addfile` command:

```
spacecmd {SSM:0}> configchannel_addfile ol6-server-config
Path: /etc/motd
Symlink [y/N]: N
Directory [y/N]: N
Owner [root]: [Enter]
Group [root]: [Enter]
Mode [0644]: [Enter]
SELinux Context [None]: [Enter]
Revision [next]: [Enter]
Read an existing file [y/N]: y
File: /var/config_file_templates/ol6-server/etc/motd
```

```
Path:            /etc/motd
Directory:       False
Owner:           root
Group:           root
Mode:            0644
SELinux Context:

Contents
--------
System Information
==================
Client system: {|rhn.system.hostname|}
Spacewalk SID: {|rhn.system.sid|}
Asset tag:     {|rhn.system.custom_info(asset_tag) = 'Asset tag missing'|}
Profile:       {|rhn.system.profile_name|}
Description:   {|rhn.system.description|}
IP address:    {|rhn.system.ip_address(eth0)|}
MAC address:   {|rhn.system.net_interface.hardware_address(eth0)|}

Is this ok [y/N]: y
```

The custom macro `rhn.system.custom_info` substitutes the value of the custom system information key named `asset_tag` as assigned on the Custom Info tab for the system. Otherwise, it inserts the value `Asset tag missing`.

See Section 10.4.3, “Defining Custom Information Keys by Using the spacecmd Command”.

See Appendix C, *Configuration File Macros*.

To display the details of a configuration channel, use the `configchannel_details` command:

```
spacecmd {SSM:0}> configchannel_details ol6-server-config
Label:       ol6-server-config
Name:        Oracle Linux 6 Server Configuration
Description: Configuration channel for generic Oracle Linux 6 servers

Files
-----
/etc/motd
```

10.4.6 Subscribing Client Systems to Configuration Channels by Using the Spacewalk Web Interface

To subscribe a client system to a configuration channel:

1. Go to **Systems** and click the system name.
2. Select the **Configuration** tab, then the **Manage Configuration Channels** tab, and finally the **Subscribe to Channels** tab.
3. For Step 1: Select Channels for Subscription page, select the check boxes for the channels to which you want to subscribe the system and click **Continue**.
4. For Step 2: Rank Channels for Subscription page, you can optionally change the order of the configuration channels according to priority. Higher-rank entries override lower-rank entries if several entries can modify the same files or directories.
5. To save your changes, click **Update Channel Rankings**.

See Section 10.4.8, “Deploying Configuration Files to Client Systems by Using the Spacewalk Web Interface”.

10.4.7 Subscribing Client Systems to Configuration Channels by Using the spacecmd Command

To list the available configuration channels, use the `configchannel_list` command as follows:

```
spacecmd {SSM:0}> configchannel_list
ol6-dns-server-config
ol6-http-server-config
ol6-nfs-server-config
ol6-server-config
```

To subscribe a system or system group to a configuration channel, use the `system_addconfigchannels` command:

```
spacecmd {SSM:0}> system_addconfigchannels svr3.mydom.com ol6-nfs-server-config -t
spacecmd {SSM:0}> system_addconfigchannels group:group3 ol6-server-config -b
```

To list the configuration channels to which a system or system group is subscribed, use the `system_listconfigchannels` command:

```
spacecmd {SSM:0}> system_listconfigchannels group:group3
System: svr3.mydom.com
ol6-nfs-server-config
ol6-server-config

##############################

System: svr4.mydom.com
ol6-server-config
```

To edit the configuration channels for a system, you can use the `system_setconfigchannelorder` command as follows:

```
spacecmd {SSM:0}> system_setconfigchannelorder svr1.mydom.com
Current Selections
------------------
1. ol6-server-config

a[dd], r[emove], c[lear], d[one]: a

Available Configuration Channels
--------------------------------
ol6-dns-server-config
ol6-http-server-config
ol6-nfs-server-config
ol6-server-config

Channel: ol6-dns-server-config
New Rank: 1

Current Selections
------------------
1. ol6-dns-server-config
2. ol6-server-config

a[dd], r[emove], c[lear], d[one]: d
```

To unsubscribe a system from a configuration channel, use the `system_removeconfigchannels` command:

```
spacecmd {SSM:0}> system_removeconfigchannels svr3.mydom.com ol6-server-config
```

10.4.8 Deploying Configuration Files to Client Systems by Using the Spacewalk Web Interface

Note

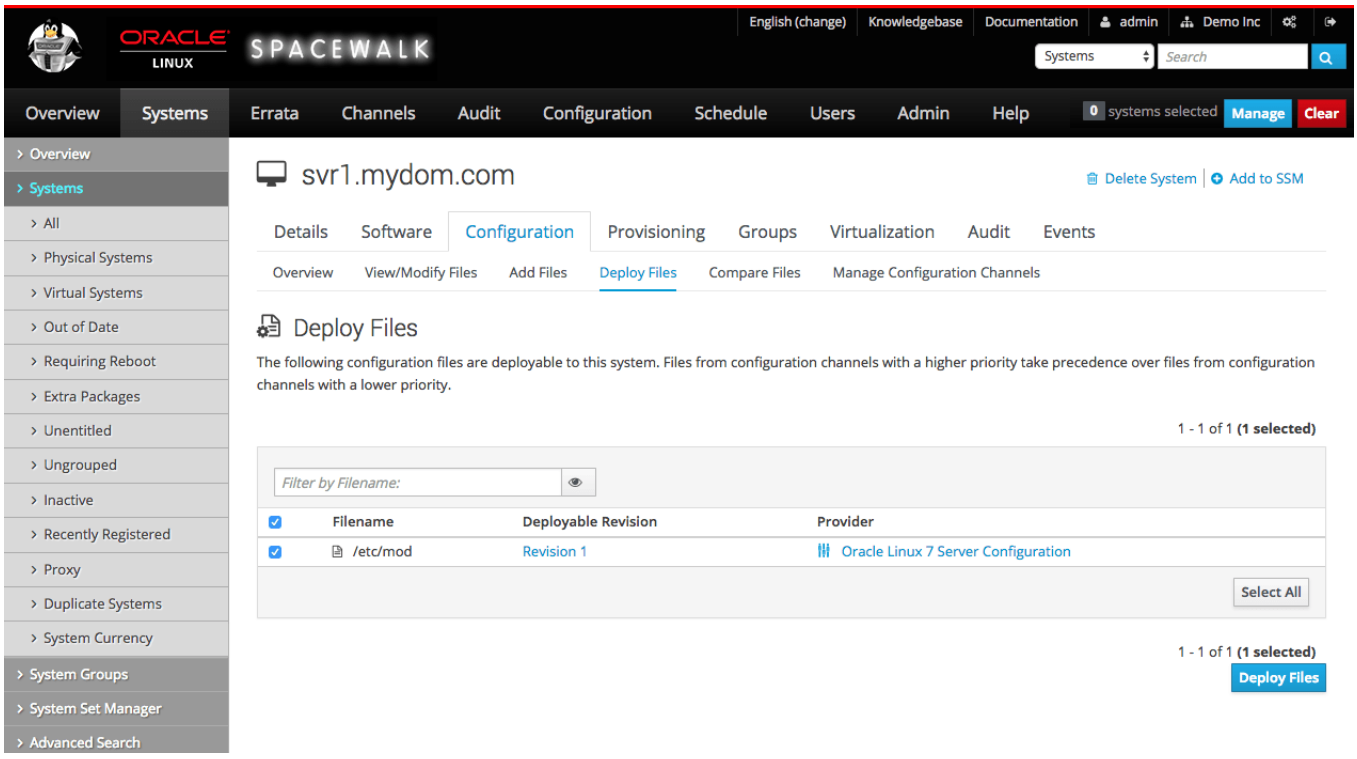
You must have previously subscribed the system to the appropriate configuration channel for the configuration files that you want to deploy. See Section 10.4.6, "Subscribing Client Systems to Configuration Channels by Using the Spacewalk Web Interface".

The first deployment to a client system can fail if the `/var/log/rhncfg-actions` log file does not exist on the client. If the deployment fails, the log file is created automatically, and you can reschedule the deployment event. Alternatively, run the following command on the client system before deploying any files to it:

```
# touch /var/log/rhncfg-actions
```

Oracle recommends using configuration files to deploy SSL and GPG certificates to Spacewalk clients that were not previously provisioned by the Spacewalk server.

Figure 10.3 Deploy Files Page



To deploy a configuration file to a client system:

1. Go to **Systems** and click the system name.
2. Select the **Configuration** tab and then the **Deploy Files** tab.
3. On the Deploy Files page, select the check boxes for the files that you want to deploy, and click **Deploy Files**.
4. On the Confirm Deploy Files page, change the schedule if required, and click **Schedule Deploy**.

5. Select **Events** and then select the **Pending** or **History** tab to view scheduled or completed actions. Click the summary name to display the details of the deployment event.

10.4.9 Deploying Configuration Files to Client Systems by Using the spacecmd Command

Note

You must have previously subscribed the system to the appropriate configuration channel for the configuration files that you want to deploy. See Section 10.4.7, “Subscribing Client Systems to Configuration Channels by Using the spacecmd Command”.

The first deployment to a client system can fail if the `/var/log/rhncfg-actions` log file does not exist on the client. If the deployment fails, the log file is created automatically, and you can reschedule the deployment event. Alternatively, run the following command on the client system before deploying any files to it:

```
# touch /var/log/rhncfg-actions
```

To deploy all configuration files to a system or system group, use the `system_deployconfigfiles` command:

```
spacecmd {SSM:0}> system_deployconfigfiles group:group3
Systems
-------
svr3.mydom.com
svr4.mydom.com

Deploy ALL configuration files to these systems [y/N]: y
INFO: Scheduled deployment for 2 system(s)
```

To display the details of a pending deployment event, use the `schedule_details` command.

See Section 10.3, “Working With Scheduled Events”.

Chapter 11 Performing OpenSCAP Auditing of Client Systems

The `scap-security-guide` package, which is available for Oracle Linux 6 and Oracle Linux 7, provides SCAP Security Guides in eXtensible Configuration Checklist Description Format (XCCDF) that have been updated to include Common Platform Enumeration (CPE) definitions for Oracle Linux.

You can use the SCAP Security Guide or any OpenSCAP compliant XCDDF or Open Vulnerability and Assessment Language (OVAL) files. Oracle provides OVAL files at https://linux.oracle.com/security/.

Note

The client system must permit the Spacewalk Server to run remote commands. See Section 7.1, “Enabling Remote Configuration in a Kickstart Profile by Using the Spacewalk Web Interface” and Section 7.3, “Enabling Remote Configuration Manually”.

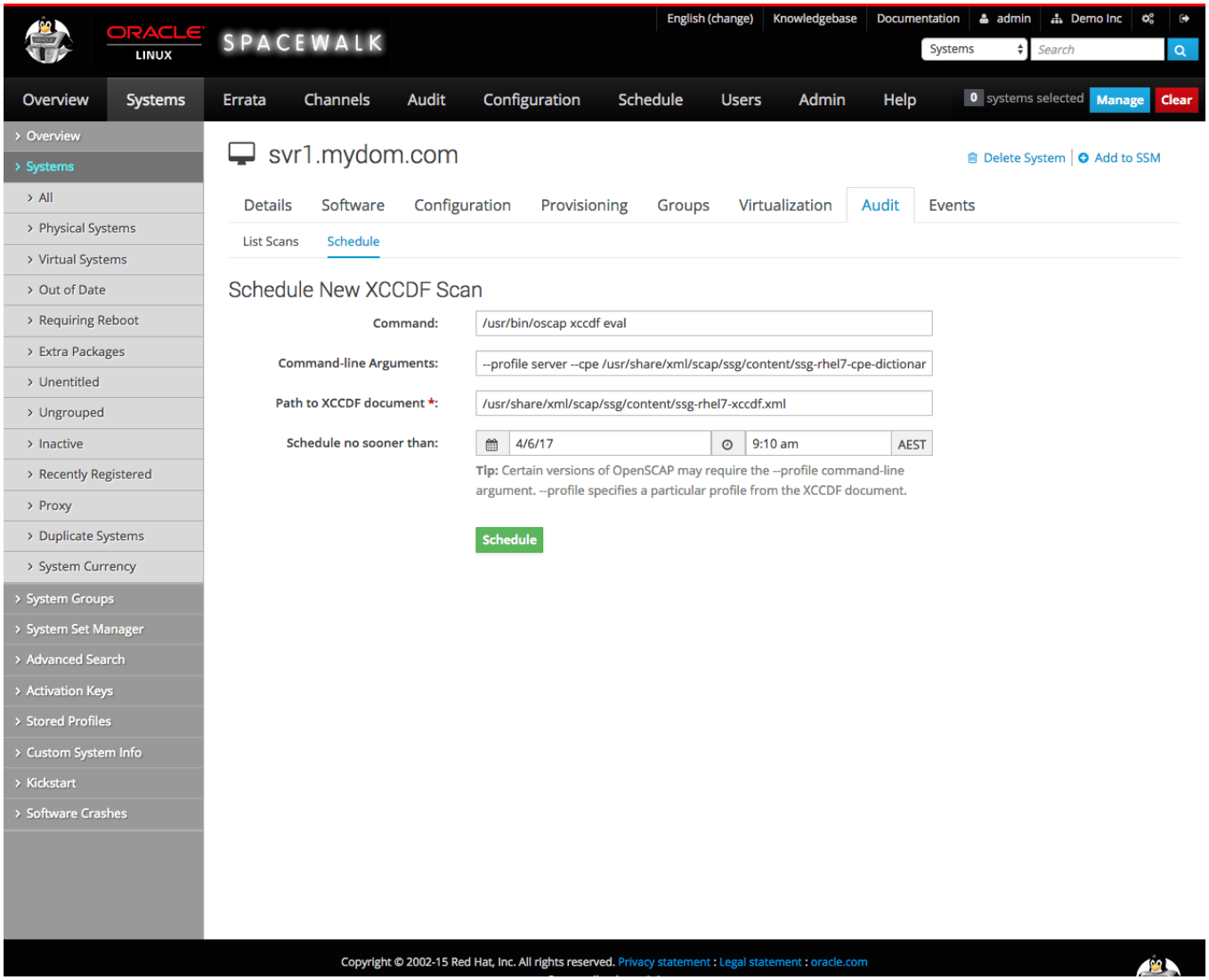
To be able to run OpenSCAP scans on a client system, install the `spacewalk-oscap` package on that system.

For more information about using OpenSCAP compliance checking with Oracle Linux, see *Running OpenSCAP Compliance Checks on Oracle Linux*.

11.1 Performing OpenSCAP Auditing of Client Systems by Using the Spacewalk Web Interface

Typically, you would use the `oscap` command with Spacewalk to perform scans. See *Oracle® Linux 6: Security Guide* and *Oracle® Linux 7: Security Guide* for more information about using this command.

Figure 11.1 Schedule New XCCDF Scan Page



To schedule a scan for a system or system group:

1. For a system:
   - Go to **Systems**, click the system name, select the **Audit** tab, and then select the **Schedule** tab.

   For a system group:

   a. Go to **Systems** and select **System Groups**.

   b. Click the system group name.

   c. On the Details page, click **Work With Group**.

      Spacewalk loads the group into the System Set Manager.

   d. Select the **Audit** tab.
2. On the Schedule New XCCDF Scan page, enter the scan settings in the following fields:

Command-line arguments	Enter any command-line arguments to the command that you are using to perform the scan. For example: `--profile server`.
Path to XCCDF document	Enter the path of the XCCDF checklist file, for example `/usr/share/xml/scap/ssg/content/ssg-rhel6-xccdf.xml`, or downloaded OVAL definition file, for example `com.oracle.elsa-2014.xml`.

3. Change the schedule if required, and click **Schedule**.

   When the scan is complete, a summary of the results of the scan are displayed under the **List Scans** tab. Oracle recommends that you schedule regular scans to check for security regressions.

11.2 Performing OpenSCAP Auditing of Client Systems by Using the spacecmd Command

Note

The `spacecmd` command supports XCCDF scans but not OVAL scans. Instead, you can use Spacewalk's remote command execution facility to run `oscap oval eval` on Spacewalk clients.

See *Oracle® Linux 6: Security Guide* and *Oracle® Linux 7: Security Guide* for more information about using the `oscap` command.

To schedule an XCCDF scan for systems, use the `scap_schedulexccdfscan` command as follows:

```
spacecmd {SSM:0}> scap_schedulexccdfscan '/usr/share/xml/scap/ssg/content/ssg-rhel6-xccdf.xml' \
'profile server' svr1.mydom.com
```

To list scheduled auditing scans, use the `schedule_list` command:

```
spacecmd {SSM:0}> schedule_list
ID      Date                 C    F    P     Action
--      ----                 ---  ---  ---   ------
522     20150625T12:56:01     0    0    1    OpenSCAP xccdf scanning
...
```

See Section 10.3, “Working With Scheduled Events”.

To list the summary results of completed XCCDF scans, use the `scap_listxccdfscans` command:

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

To list the details and results of an XCCDF scan, specified by its scan ID, use the `scap_getxccdfscandetails` and `scap_getxccdfscanruleresults` commands, as shown in the following example:

```
spacecmd {SSM:0}> scap_getxccdfscandetails scan_ID
spacecmd {SSM:0}> scap_getxccdfscanruleresults scan_ID
```

Chapter 12 Configuring Ksplice Offline Clients

On average, the Linux kernel receives security updates and bug fixes about once per month. Traditionally, applying such updates would require you to obtain and install the updated kernel RPMs, to schedule downtime, and to reboot the server into the new kernel with the critical updates. As system setups become more complex with many interdependencies, and access to services and applications must remain as undisrupted as possible, scheduling such reboots becomes more difficult and costly.

Oracle Ksplice provides a way for you to keep your systems secure and highly available by enabling you to update them with the latest kernel, Xen hypervisor updates, and key user space security and bug fix updates. Oracle Ksplice updates the running operating system and Xen hypervisor without requiring a reboot. Your systems remain up to date with OS vulnerability patches and downtime is minimized. A Ksplice update takes effect immediately when it is applied. The update is not the same as an on-disk change that only takes effect after a subsequent reboot.

Oracle creates each Ksplice update from a kernel update that originates either from Oracle or from the Linux kernel community.

The *Ksplice Offline Client* removes the requirement for a server on your intranet to have a direct connection to the Oracle Uptrack server. All available Ksplice updates for each supported kernel version are bundled into an RPM that is specific to that version, and this package is updated every time that a new Ksplice patch becomes available for the kernel.

Note

Ksplice Offline Client is freely available for Oracle Linux customers that subscribe to Oracle Linux Premier Support. If you are an Oracle Linux Basic, Basic Limited, or Network Support subscriber, contact your sales representatives to discuss a potential upgrade of your subscription to a Premier Support plan.

You can configure a Spacewalk server as a mirror of the Ksplice for Oracle Linux channels on ULN. The Spacewalk server does not require access to the Oracle Uptrack server. Instead, you schedule Spacewalk to download the latest Ksplice update packages to a software channel. For older Ksplice updates, an archive channel is available. The `_archive` suffix is usually added to the channel for which it hosts archive packages. See Section 2.1, “About Software Channel Configuration”

After installing Ksplice Offline Client on your Spacewalk client systems, they can install the Ksplice update packages from the Spacewalk server. The clients also do not require access the Oracle Uptrack server.

Note

You cannot use the web interface or the Ksplice Uptrack API to monitor systems that are running Ksplice Offline Client, as these systems are not registered with https://status-ksplice.oracle.com/.

For more information about Ksplice see *Oracle® Linux: Ksplice User's Guide*.

12.1 Supported Kernels

You can use Ksplice Uptrack to bring the following Oracle Linux kernels up to date with the latest important security and bug fix patches:

- All Oracle Linux 6 and Oracle Linux 7 kernels starting with the official release.
- All Oracle Unbreakable Enterprise Kernel versions for Oracle Linux 6, starting with 2.6.32-100.28.9 (released March 16, 2011).

To confirm whether a particular kernel is supported, install the Uptrack client on a system that is running the kernel.

If you have a question about supported kernels, send e-mail to ksplice-support_ww@oracle.com.

12.2 Configuring a Spacewalk Server to Act as a Ksplice Mirror

To configure a Spacewalk server to act as a Ksplice mirror, you configure repositories and associated software channels for the Oracle Linux releases and architectures of the clients on which you want to run Ksplice Offline Client. Each Ksplice channel should be a child of the appropriate base software channel. See Section 2.4, “Working With Repositories” and Section 2.5, “Working With Software Channels”.

The following table shows the channels that are available for Ksplice on Oracle Linux.

Channel Name	Channel Label	Description
Ksplice for Oracle Linux 6 (i386)	`ol6_i386_ksplice`	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on i386 systems.
Ksplice for Oracle Linux 6 (x86_64)	`ol6_x86_64_ksplice`	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 6 on x86-64 systems.
Ksplice for Oracle Linux 7 (x86_64)	`ol7_x86_64_ksplice`	Oracle Ksplice clients, updates, and dependencies for Oracle Linux 7 on x86_64 systems.
Ksplice aware user space packages for Oracle Linux 6 (x86_64)	`ol6_x86_64_userspace_ksplice`	Latest packages for Ksplice aware user space packages for Oracle Linux 6 (x86_64). This channel should only be used with the Ksplice Enhanced client.
Ksplice aware user space packages for Oracle Linux 7 (x86_64).	`ol7_x86_64_userspace_ksplice`	Latest packages for Ksplice aware user space packages for Oracle Linux 7 (x86_64). This channel should only be used with the Ksplice Enhanced client.

For example, you would specify the URL of the Ksplice for Oracle Linux 6 (x86_64) channel on ULN as follows:

```
uln:///ol6_x86_64_ksplice
```

Caution

To reduce the overall space consumed by Ksplice Offline packages, Oracle strongly recommends using repository filters to limit downloaded packages to only those required by your client systems. You can apply a filter either on the Repository configuration in the Manage Repositories page in the web UI or by providing the `-i` or `--include` parameter with the `spacewalk-repo-sync` command line, for example:

```
# spacewalk-repo-sync --channel ol6_x86_64_ksplice -t uln \
  -i uptrack-updates-installed kernel base version,...
```

12.3 Provisioning Client Systems as Ksplice Offline Clients

To provision a client system as a Ksplice offline client, configure its kickstart profile as follows:

- Under **Kickstart Details**, select the **Operating System** tab, ensure that the check box for the Ksplice child software channel is checked, and click **Update Kickstart**.

- Under **Software**, include `uptrack-offline` in the list of packages to install.
- Under **Scripts**, create a post-installation, `nochroot` shell script that installs the Ksplice update packages.

  For Oracle Linux 6 or Oracle Linux 7:

```
yum install uptrack-updates-`uname -r`
```

Install new Ksplice updates as they become available. You can schedule Spacewalk to update the client system or you can set up an `anacron` script on the client itself. For example, you could use the following script with an Oracle Linux 6 or Oracle Linux 7 client:

```
#!/bin/sh
yum install uptrack-updates-`uname -r`
```

The script must be executable and also must be owned by `root`. If you place the script in `/etc/cron.daily` on the client, it runs once every day.

12.4 Installing and Configuring Existing Client Systems as Ksplice Offline Clients

After you have set up Spacewalk to act as a Ksplice mirror, you can configure your other systems to receive `yum` and Ksplice updates.

To configure a system as a Ksplice offline client, follow these steps:

1. Subscribe the client system to the Ksplice software channel that corresponds to the Oracle Linux release and architecture.
2. Install the offline version of the enhanced Ksplice client package:

```
# yum install ksplice-offline
```

3. Insert a configuration directive into `/etc/uptrack/uptrack.conf` to provide the enhanced client with the label of the local user-space channel in your local Yum repo configuration. You do not need to do this if you did not use the `local_` prefix for the channel label and this label matches the label used on ULN exactly. If you used the `local_` prefix or labeled this channel differently, add the following lines and replace *`local_ol6_x86_64_ksplice_userspace`* with whatever you used to label the Ksplice user-space channel:

```
[User]
yum_userspace_ksplice_repo_name = local_ol6_x86_64_ksplice_userspace
```

4. To install offline update packages, you must install the relevant packages for your system. For example, you might install the following packages:

```
# yum install ksplice-updates-glibc ksplice-updates-openssl
```

   When these packages have been installed, the offline version of the enhanced Ksplice client behaves exactly the same as the online version.

5. Update the system to install the Ksplice-aware versions of the user-space libraries:

```
# yum update
```

   To install only the libraries and not update any other packages, limit the update to the `ol6_x86_64_userspace_ksplice` or `ol7_x86_64_userspace_ksplice` channel as appropriate, for example:

```
# yum --disablerepo=* --enablerepo=ol7_x86_64_userspace_ksplice update
```

Alternatively, use the following command:

```
# yum update *glibc *openssl*
```

You may also use this client to perform kernel updates, in the same way that you are able to use the standard uptrack client:

```
# yum install uptrack-updates-`uname -r`
```

6. To enable the automatic installation of updates, change the following entry in `/etc/uptrack/uptrack.conf`:

```
autoinstall = no
```

so that it reads:

```
autoinstall = yes
```

7. Reboot the system so that the system uses the new libraries.

On Oracle Linux 6:

```
# reboot
```

On Oracle Linux 7:

```
# systemctl reboot
```

Appendix A Kickstart Options

Using the Spacewalk web interface, you can specify the following options for a kickstart profile:

`auth`

(Mandatory) Specifies whether the shadow password file is used and the password algorithm. The default setting is `--enableshadow --passalgo=sha256`, which enables the shadow password file and specifies SHA256 as the password algorithm. If you change the password algorithm, the password hash specified for `rootpwd` must have been generated by using the same algorithm or you will not be able to log in to the installed system.

See the `authconfig(8)` manual page for a list of the options that you can specify.

`autopart`

Specifies whether the installation should perform automatic partitioning. If you specify this option, you should also specify `clearpart` and `zerombr`. Use `ignoredisk` to specify the disks that the Installer should or should not use.

`autostep`

```
autostep [ --autoscreenshot ]
```

Specifies that the Installer should step through every screen.

`bootloader`

```
bootloader --location={mbr|none|partition}
         [ --append="boot-loader-kernel-parameters" ]
```

(Mandatory) Specifies whether the boot loader is installed in the MBR or in a disk partition. The default setting is `--location mbr`.

`cdrom`

Specifies that the installation is from the first CD-ROM drive on the system.

`clearpart`

```
clearpart [ --all [ --initlabel ] |
            --linux |
            --list=part1,... |
            --none ]
          [ --drives=drive1,... ]
          [ --initlabel ]
```

Specifies whether to clear any existing partitions. For example, `--drives=sda --all --initlabel` would clear all partitions on the disk device sda and reinitialize the disk lablel.

Caution

The default setting of `--all` clears all partitions on all attached disks.

`cmdline`

Specifies that the installation should be performed in non-interactive, command-line mode.

`device`

```
device {eth|scsi}module_name --opts="module options"
```

Specifies the module name and options for a system device.

`deviceprobe`

Specifies how to probe for devices.

`driverdisk`

```
driverdisk partition [ --type=fstype ]
```

```
driverdisk --source=ftp://image_path

driverdisk --source=http://server/image_path

driverdisk --source=nfs:server:image_path
```

Specifies a driver disk.

`firewall`

```
firewall { --disabled | --enabled } [ --ftp ] [ --http ]
        [ --port=inbound_port1:{tcp|udp},... ]
        [ --smtp ] [ --ssh ]
        [ --trust=network_interface ]...
```

Specifies the configuration of the system firewall. The default value is `--disabled`.

`firstboot`

```
firstboot { --enable | --disable | --reconfig }
```

Specifies how the setup agent starts when the system is first booted. If enabled, the `initial-setup` package must be installed.

`graphical`

Perform a graphical installation. It is not usual to select this option for non-interactive kickstart installations.

`halt`

Specifies that the Installer should halt the system after installation is complete and wait for a key to be pressed on the console before rebooting.

`harddrive`

```
harddrive [ --biospart=BIOS_partition | --partition=partition ]
          [ --dir=install_directory ]
```

Specifies an installation directory on a local hard drive.

`ignoredisk`

```
ignoredisk { --drives=[disk1,...] | --only-use=[disk1,...]
```

Specifies disks that the Installer should or should not use during installation.

`install`

Specifies that the Installer should perform a new installation. This option is specified by default.

`interactive`

Specifies that the installation should be interactive.

`iscsi`

```
iscsi --ipaddr=target_addr --target=target_IQN
    [ --iface=network_interface ] [ --port=target_port ]
    [ --user=target_username --password=target_password ]
    [ --reverse-username=initiator_username
      --reverse-password=initiator_password ]
```

Specifies iSCSI storage to be used during installation.

`iscsiname`

Specifies the iSCSI initiator name for the system.

`key`

Specifies an installation key for package selection and system identification.

`keyboard`

(Mandatory) Specifies the keyboard layout. The default setting is `us`.

`lang`

(Mandatory) Specifies the language to be used for installation and the default locale on the installed system. The default setting is `en_US`.

`logging`

```
logging [ --host=remote_host ]
        [ --level={critical|debug|error|info|warning} ]
        [ --port=remote_port ]
```

Configures installation error logging.

`monitor`

```
monitor [ --hsync=Hfreq ] [ --vsync=Vfreq ]
```

Specifies the monitor's horizontal and vertical synchronization frequency settings.

`mouse`

Deprecated. Do not use.

`multipath`

```
multipath --name=pathname --device=device --rule=rule
```

Specifies a multipath device.

`network`

```
network --bootproto=dhcp [ --device=interface ]
      [ --onboot={no|yes} ]

network --bootproto=static [ --device=network_interface ]
      [ --onboot={no|yes} ]
      [ --noipv4 | --ip=IP_addr --netmask=netmask ]
      [ --noipv6 | --ipv6={auto|dhcp|IPv6_addr/prefix} ]
      [ --gateway=gateway_addr ]
      [ --nameserver=namesvr_addr ]
```

Specifies the configuration of the network interfaces. The default setting is `--bootproto dhcp`, which configures the network interface used for installation to use DHCP to obtain its network settings.

`nfs`

```
nfs --server=NFSserver --dir=install_directory
  [ --opts=mount_options ]
```

Specifies an NFS server and directory path to use for installation.

`poweroff`

Specifies that the Installer should power down the system after installation is complete.

`reboot`

Specifies that the Installer should reboot the system after installation is complete. This option is specified by default. For unattended installations, the `poweroff` option might be preferable.

`rootpw`

```
rootpw { --iscrypted | --plaintext } password
```

(Mandatory) Specifies the `root` password as a hash value or in plain text. This option is specified by default. If you specify a plain text password, select the **Encrypt** check box.

`selinux`

```
selinux { --disabled | --enforcing | --permissive }
```

Specifies the SELinux mode as disabled, enforcing, or permissive. The default setting is `--permissive`.

`services`

```
services [ --disabled=service1,... ]
         [ --enabled=serviceA,... ]
```

Specifies which services to disable or enable at the default run level.

`shutdown`

Specifies that the Installer should shut down the system after installation is complete but not power it down.

`skipx`

Do not install X on the system. This option is specified by default.

`text`

Perform a text-only installation. This option is specified by default as kickstart installations are usually non-interactive.

`timezone`

```
timezome [ --utc ] timezone
```

(Mandatory) Specifies the time zone and whether the hardware clock uses UTC (`--utc`). The default setting is `America/New_York`.

`upgrade`

Specifies that the Installer should perform an upgrade installation.

`url`

```
url --url={file_path |
           ftp://username:password@server/path |
           http://server/path}
```

Specifies the URL of the kickstart file. By default, the kickstart URL is specified as a file path and Spacewalk writes the correct, full URL to the kickstart file, depending on whether the system being installed connects directly to a Spacewalk server or via a Spacewalk proxy. If you enter a full URL instead of a file path, Spacewalk does not modify the URL.

`user`

```
user --name=username [ --groups=group1,... ]
  [ --homedir=directory ]
  [ --password=password ] [ --iscrypted ]
  [ --shell=shell_path ] [ --uid=UID ]
```

Specifies a user to be created on the system.

`vnc`

```
vnc [ --host=hostname ] [ --port=port ]
    [ --password=password ]
```

Specifies parameters for running a VNC server on the system being installed.

`xconfig`

```
xconfig [ --defaultdesktop={GNOME|KDE} ]
        [ --depth={8|16|24|32} ]
        [ --resolution=XxY ]
        [ --startxonboot ]
```

Specifies X Window System parameters.

`zerombr`

Specifies whether to clear the existing disk partitions. This option is specified by default.

`zfcp`

```
zfcp [ --devnum=num ] [ --fcplun=lun ]
     [ --scsiid=id ] [ --scsilun=lun ]
     [ --wwpn=name ]
```

Specifies zFCP parameters for Fibre Channel-attached SCSI devices.

Appendix B Sample Minimum Package Lists

The following packages provide a suitable minimum (*Just Enough OS*) installation to support all Spacewalk management, monitoring configuration, and auditing functions. You can use the `yum` command to install any other packages that are required to configure an Oracle Linux server.

```
@ Base
osad
rhncfg
rhncfg-actions
rhncfg-client
rhncfg-management
```

Appendix C Configuration File Macros

You can use the following standard macros with configuration files:

`rhn.system.custom_info(key_name)`

Value of the key *key_name* associated with the system.

`rhn.system.description`

System description.

`rhn.system.hostname`

System host name.

`rhn.system.ip_address`

Default system IP address.

`rhn.system.net_interface.broadcast(ethN)`

Broadcast address associated with `ethN`.

`rhn.system.net_interface.driver_module(ethN)`

Network interface driver module associated with `ethN`.

`rhn.system.net_interface.hardware_address(ethN)`

MAC address associated with `ethN`.

`rhn.system.net_interface.ip_address(ethN)`

IP address associated with `ethN`.

`rhn.system.net_interface.netmask(ethN)`

Network mask associated with `ethN`.

`rhn.system.profile_name`

Kickstart profile associated with a system.

`rhn.system.sid`

Spacewalk system ID.

Appendix D Spacewalk XML/RPC API

If you are an advanced user, you can use the Spacewalk XML/RPC API to create web interfaces and scripts to perform or automate tasks. More information about the API is available at `https://swksvr_FQDN/rpc/api` on a Spacewalk server.

For example, the following `get-channel-summaries` Python script uses the API to obtain a list of channels, the numbers of packages in each channel, and the number of systems that are subscribed to each channel:

```
#!/usr/bin/python
#
#  get-channel-summaries [--server URL <url>] [--username <user>] [--password <passwd>]

import getopt, struct, sys, xmlrpclib
from array import *

# Insert default values for the Spacewalk server API URL,
# Spacewalk admin user name, and Spacewalk admin password
url = "https://swksvr.mydom.com/rpc/api"
username = "swadmin"
password = "swadmin"

usage1 = "Usage: get-channel-summaries [--serverUrl <url>] \\\n"
usage2 = "         [--username <user>] [--password <passwd>]"

try:
    opts,args = getopt.getopt(sys.argv[1:],"s:u:p:",["serverUrl=","username=","password="])
except getopt.GetoptError as err:
    print(usage1+usage2)
    sys.exit(1)

for o,a in opts:
    if o in ("-s", "--serverUrl"):
        url = a
    elif o in ("-u", "--username"):
        username = a
    elif o in ("-p", "--password"):
        password = a
    else:
        assert False, "Unknown option"

# Connect to Spacewalk
client = xmlrpclib.Server(url,verbose=0)
session = client.auth.login(username,password)

# Get channel list
channels = client.channel.listAllChannels(session)

# Build channel arrays indexed by channel ID
channel_label = {}
channel_packages = {}
channel_systems = {}

for channel in channels:
    channel_label[channel['id']] =  channel['label']
    channel_packages[channel['id']] =  channel['packages']
    channel_systems[channel['id']] =  channel['systems']

# Print output header
fmt1 = '{0:<40s}{1:<10s}{2:<10s}'
print fmt1.format('Channel label','Packages','Systems')
print fmt1.format('-------------','--------','-------')

# Print channel label, package count, and system count -- sorted by label
```

```
fmt2 = '{0:<40s}{1:<10d}{2:<10d}'
for key,value in sorted(channel_label.iteritems(),key=lambda(k,v): (v,k)):
    id = int(key)
    print fmt2.format(value,channel_packages[id],channel_systems[id])

# Disconnect from Spacewalk
client.auth.logout(session)
```

The following is sample output from running this script:

```
Channel label                           Packages  Systems
-------------                           --------  -------
base-channel                            0         0
epel6-channel-label                     68        0
epel6-channel1-label                    3         0
epel6-channel2-label                    1         0
ol5-i386-channel-label                  44        0
ol5-i386-test-label                     34        0
ol5-x64-channel-label                   44        0
ol6-i386-channel-label                  45        0
ol6-x64-channel-label                   45        1
ol7-channel-label                       45        1
oraclelinux6-x86_64                      0        0
oraclelinux6-x86_64-addons               0        0
oraclelinux6-x86_64-spacewalk22-client   0        0
oraclelinux6-x86_64-spacewalk22-server   0        0
oraclelinux6-x86_64-spacewalk24-client   0        0
oraclelinux6-x86_64-spacewalk24-server   0        0
oraclelinux6-x86_64-spacewalk26-client   0        0
oraclelinux6-x86_64-spacewalk26-server   0        0
oraclelinux6-x86_64-spacewalk27-server   0        0
uln-channel-label                        173      0
```

The following example shows the `get-reposync-list` script, which displays the schedules for synchronizing repositories.

```
#!/usr/bin/python
#
#  get-reposync-list [--serverUrl <url>] [--username <user>] [--password <passwd>]

import getopt, struct, sys, xmlrpclib
from array import *

# Insert default values for the Spacewalk server API URL,
# Spacewalk admin user name, and Spacewalk admin password
url = "https://swksvr.mydom.com/rpc/api"
username = "swadmin"
password = "swadmin"

usage1 = "Usage: get-reposync-list [--serverUrl <url>] \\\n"
usage2 = "         [--username <user>] [--password <passwd>]"

try:
    opts,args = getopt.getopt(sys.argv[1:],"s:u:p:",["serverUrl=","username=","password="])
except getopt.GetoptError as err:
    print(usage1+usage2)
    sys.exit(1)

for o,a in opts:
    if o in ("-s", "--serverUrl"):
        url = a
    elif o in ("-u", "--username"):
        username = a
    elif o in ("-p", "--password"):
        password = a
    else:
        assert False, "Unknown option"
```

```
# Connect to Spacewalk
client = xmlrpclib.Server(url,verbose=0)
session = client.auth.login(username,password)

# Get channel list
channels = client.channel.listAllChannels(session)

# Build channel name array indexed by channel ID
channel_label = {}
channel_schedule = {}

for channel in channels:
    id = int(channel['id'])
    channel_label[id] =  channel['label']
    channel_schedule[id] = ''

# Get repository synchronization list
schedules = client.taskomatic.org.listActiveSchedulesByBunch(session,'repo-sync-bunch')

# Construct schedule array indexed by channel ID
for schedule in schedules:
    channel_schedule[int(schedule['data_map']['channel_id'])] = schedule['cron_expr']

# Print output header
fmt = '{0:<40s}{1:<40s}'
print fmt.format('Channel label','Schedule')
print fmt.format('-------------','--------')

# Print channel labels and repository synchronization schedule (if defined)
for key,value in sorted(channel_label.iteritems(),key=lambda(k,v):(v,k)):
    id = int(key)
    sched = channel_schedule[id]
    if (len(sched) > 0):
        print fmt.format(value,sched)
    else:
        print fmt.format(value,"Sync not scheduled")

# Disconnect from Spacewalk
client.auth.logout(session)
```

The following example shows what the output of this command might look like:

```
Channel label                           Schedule
-------------                           --------
oraclelinux6-x86_64                     0 30 0 ? * *
oraclelinux6-x86_64-addons              0 30 2 ? * *
oraclelinux6-x86_64-mysql               0 30 4 ? * *
oraclelinux6-x86_64-playground          0 30 3 ? * *
oraclelinux6-x86_64-spacewalk24-client  0 0 5 ? * *
oraclelinux6-x86_64-spacewalk24-server  0 30 5 ? * *
oraclelinux6-x86_64-spacewalk26-client  0 0 2 ? * *
oraclelinux6-x86_64-spacewalk26-server  0 0 3 ? * *
oraclelinux6-x86_64-spacewalk27-server  0 0 4 ? * *
oraclelinux6-x86_64-uek                 0 0 5 ? * *
oraclelinux6-x86_64-uek-r3              0 30 1 ? * *
...
```