

Oracle Linux Manager

Concepts Guide for Release 2.10



F36091-04
January 2023



Oracle Linux Manager Concepts Guide for Release 2.10,

F36091-04

Copyright © 2021, 2023, Oracle and/or its affiliates.

Contents

Preface

Conventions	iv
Documentation Accessibility	iv
Access to Oracle Support for Accessibility	iv
Diversity and Inclusion	iv

1 About Oracle Linux Manager

Overview of Oracle Linux Manager Architecture	1-3
Oracle Linux Manager Concepts	1-5

2 About Software Channels

Differences Between ULN and Oracle Linux Yum Server Channels	2-1
Optimizing the Speed of Channel Synchronization	2-2

3 Provisioning Client Systems in Oracle Linux Manager

4 Maintaining Software Channels in Oracle Linux Manager

5 Overview of Development, Test, Acceptance, and Production Life Cycles

6 Configuring System Groups to Standardize Configurations

7 Applying Errata to Systems and System Groups

Preface

[Oracle Linux Manager: Concepts Guide for Release 2.10](#) describes the basic concepts and features of Oracle Linux Manager 2.10 which is based on the Spacewalk open source project. It provides an overview of how to use Oracle Linux Manager, including recommended best practices for using the software in enterprise environments.

Conventions

The following text conventions are used in this document:

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <https://www.oracle.com/corporate/accessibility/>.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab>.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. 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 ongoing and will take time and external cooperation.

1

About Oracle Linux Manager

A central task for IT administrators is to provision systems and keep them up to date with the latest patches and operating system updates. If the security policy at your site mandates errata as critical, it is crucially important to conduct regular testing and apply patches to reduce the risk of systems compromise or data exposure. As data centers expand, administrators are required to manage even greater numbers of physical servers and virtual machines. As a result, automation becomes a necessity for efficient and cost-effective systems management. Oracle Linux Manager 2.10, based on the Spacewalk open source software, helps automate Oracle Linux systems management, thereby enabling you to control the system software life cycle, from initial installation, through maintenance, software configuration, upgrades, and eventual decommissioning.

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

This guide introduces you to the basic concepts of using Oracle Linux Manager to manage Oracle Linux systems. Beyond the scope of this guide are additional efficiencies that can result from implementing Oracle Linux Manager.

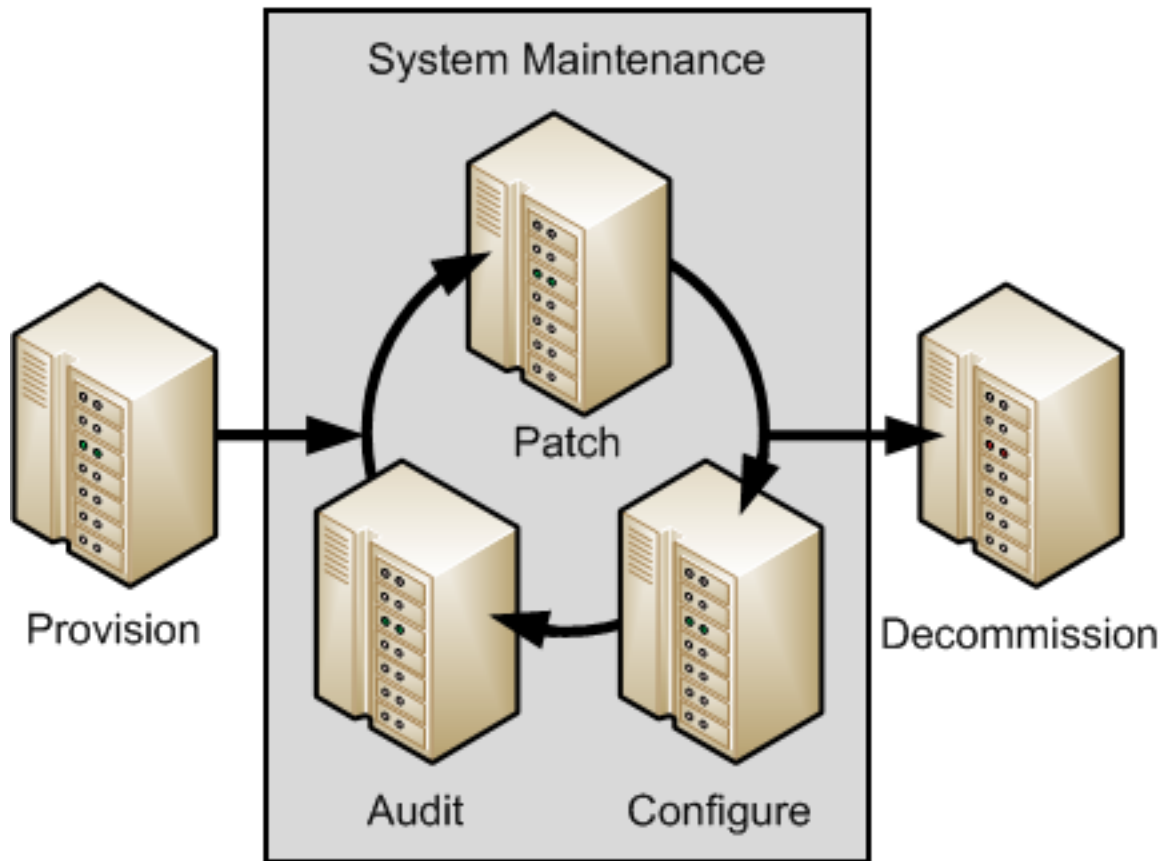
For example, you can use Oracle Linux Manager to do the following:

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

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

As illustrated in the following figure, during the typical life cycle of a system, you can use Oracle Linux Manager to simplify several system management tasks, such as installing bare metal systems and virtual guests, applying patches and software updates, configuring software, and auditing system security.

Figure 1-1 Life Cycle of a System



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

Oracle Linux Premier Support and Oracle Linux Basic Support contracts provide support for Oracle Linux Manager and include a limited-use license for Oracle Database Enterprise Edition for use with Oracle Linux Manager. You can use Oracle Linux Manager to automate the management of Oracle Linux 6, Oracle Linux 7, and Oracle Linux 8 systems. When deployed on Oracle Linux, Oracle Linux Manager can additionally manage other Linux releases, such as CentOS, Debian, Fedora, and SLES. However, note that Oracle does not support Oracle Linux Manager for management of these systems.

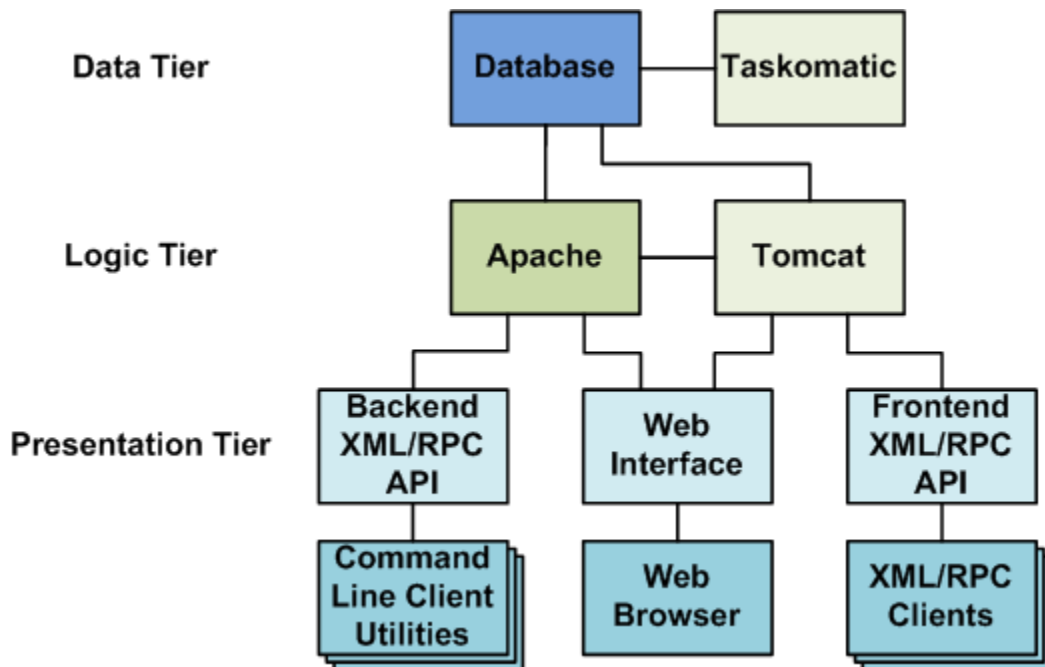
Oracle Linux Manager provides automated management capabilities at no additional cost. Oracle support for Oracle Linux Manager is especially valuable when transitioning to Oracle Linux, for those who are already familiar with Red Hat Satellite 5 releases or SUSE Manager. However, your IT organization might prefer to implement Oracle Enterprise Manager instead because it provides a comprehensive management solution that extends beyond the management of purely Linux-derived systems. The Oracle Enterprise Manager product family supports the management of

the entire Oracle software stack, including hardware, hypervisors, operating systems and Oracle Database, Oracle middleware, and Oracle software applications. For more information, visit the Oracle Enterprise Manager product page at <https://www.oracle.com/enterprise-manager/>.

Overview of Oracle Linux Manager Architecture

The following figure depicts an internal, three-tiered Oracle Linux Manager server architecture.

Figure 1-2 Overview of the Internal Three-Tiered Oracle Linux Manager Server Architecture



The server architecture includes the following three tiers:

- **Data tier:** Contains the database and Taskomatic. Oracle Linux Manager uses the Taskomatic daemon to perform asynchronous scheduled tasks, such as resynchronizing software channels, applying software and configuration updates to clients, and notifying you when new errata become available.
- **Logic tier:** Contains the Apache and Tomcat web servers, which process data for use by the presentation tier.
- **Presentation tier:** Contains the back-end and front-end XML/RPC APIs, which provide programming interfaces for the command-line client utilities and other XML/RPC clients, and the web interface, which can be accessed by using a web browser.

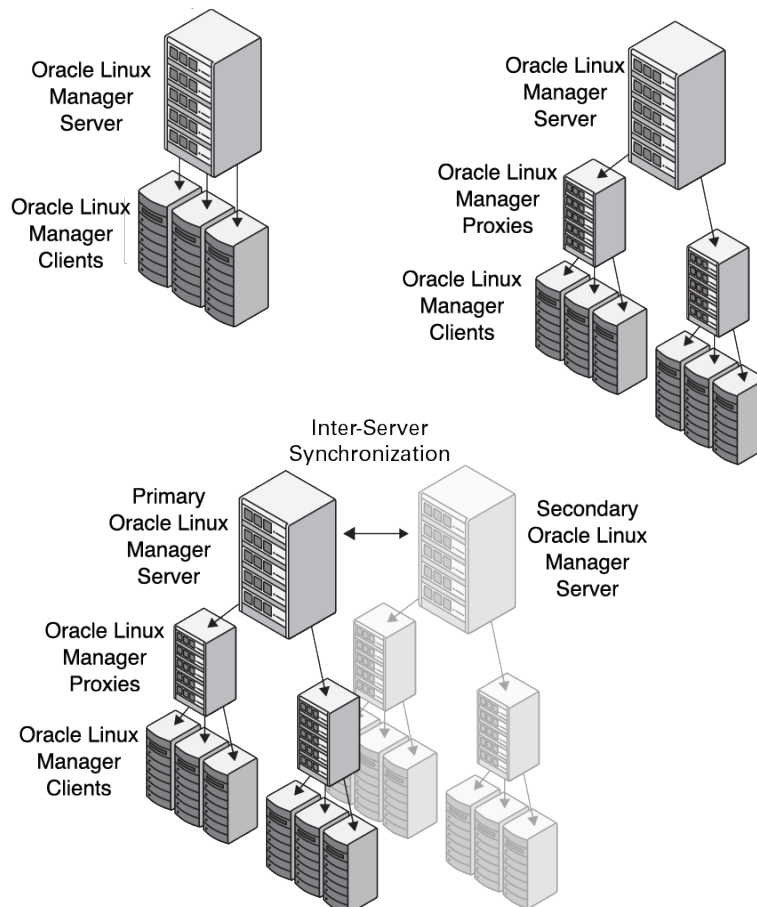
Oracle Linux Manager uses a distributed client-server architecture, in which registered client systems subscribe to the software channels that an Oracle Linux Manager server hosts. You can distribute server functionality across as many systems as are required to meet your organization's needs. This capability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

For example, a simple deployment might have a single primary server to serve a pool of client systems, along with a proxy server to offload processing from the primary so that you can support more client systems. In larger deployments, you can configure multiple Oracle Linux Manager primary servers and proxy servers to improve provisioning and software download speeds. In a distributed configuration, clients are likely to have higher network bandwidth access to servers that are within close proximity.

Because Oracle Linux Manager architecture is flexible and scalable, many different deployment configurations are feasible. The following scenarios, which are also illustrated in [Figure 1-3](#), are possible:

- Simple deployment with a single Oracle Linux Manager server that has several clients.
- More complex deployment with one Oracle Linux Manager server and two Oracle Linux Manager proxies, where each proxy has several clients.
- Deployment with two Oracle Linux Manager servers, each at different sites, where the server at one site acts as the primary server, while the server at the other site acts as a secondary or worker server. Inter-Server Synchronization (ISS) is used to manage channel content, channel permissions, and organizational trust settings between the two servers.

Figure 1-3 Deployment Configurations for Oracle Linux Manager



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

You register client systems, whether physical servers or virtual guests, with a *Oracle Linux Manager Server Proxy* so that they can be subscribed to software channels. The clients can then obtain packages from Oracle Linux Manager server. You can also use Oracle Linux Manager with kickstart to automate an Oracle Linux installation by using software packages from a network installation server. If you use Oracle Linux Manager with kickstart to provision new client systems, you can configure Oracle Linux Manager server to automatically register these client systems. A new client system is automatically registered as an Oracle Linux Manager client if you associate an activation key with its kickstart profile and configure kickstart to install Oracle Linux Manager Client software on the system.

You can use Oracle Linux Manager to administer a registered kickstarted system immediately, which simplifies subsequent patching, configuration management, and security auditing. You can also register previously installed legacy systems with an Oracle Linux Manager server to bring these systems under Oracle Linux Manager's control.

Oracle Linux Manager Concepts

This chapter describes some basic Oracle Linux Manager concepts. For more information about these concepts, as well as step-by-step instructions and examples, see the [Oracle® Linux Manager & Spacewalk for Oracle® Linux Documentation](#).

Activation Key

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

Software Channel

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

Note:

Oracle also uses the term *channel* to refer to the software distribution channels that the Unbreakable Linux Network (ULN) provides. A *channel* is the subscription mechanism by which clients can obtain software packages, patches, and updates. You do not necessarily need to associate an upstream repository with a software channel. For example, you might want to instead add custom-built, local packages and errata by using the web interface or by using the `rhnpush` command.

Entitlement

Enables additional functionality that is supported by Oracle Linux Manager. Note that starting with Spacewalk 2.6, most entitlements were removed. The only entitlement that is currently

supported is the Virtualization entitlement, which installs additional packages on the target host so that Oracle Linux Manager is able to manage virtual guests on that host.

System Group

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

Organization

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

Repository

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



Note:

Ksplice Offline packages are available from ULN as part of an Oracle Linux Premier Support contract. For more information, see [Oracle Linux: Ksplice User's Guide](#) .

2

About Software Channels

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

Beginning from Oracle Linux 7.1 as well as Oracle Linux 6.9, you do not need to install Oracle Linux Manager client packages in order to register clients. However, Oracle recommends installing Oracle Linux Manager client channel packages to take full advantage of Oracle Linux Manager's management capabilities, including provisioning and auditing. You also need to enable Oracle Linux Manager client channel if you want kickstart to register a client automatically, as kickstart does not recognize the built-in client for this purpose.

An easy way to populate a software channel initially, so that it can be used with kickstart, is to mount an ISO image on the server and use the `rhnpush` command to push packages from the ISO distribution.

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

Note:

Before running the `rhnpush` command, the system that you are using must be registered with Oracle Linux Manager server, even if the command is being run on the same system that is hosting Oracle Linux Manager server. The `rhnpush` command uses the Transport Layer Security/Secure Socket Layer (TLS/SSL) protocols to communicate with Oracle Linux Manager server. If the system is not registered, the `rhnpush` command fails with a certificate error. Client registration updates the configuration to use the correct Oracle Linux Manager SSL certificate. For more information about client registration, see Registering Client Systems in [Oracle Linux Manager: Client Life Cycle Management Guide for Release 2.10](#).

Differences Between ULN and Oracle Linux Yum Server Channels

ULN provides base and patch channels for each update of an Oracle Linux release. The Oracle Linux yum server does *not* provide separate patch channels. Instead, it provides a `olN_latest` repository that includes all of the packages for an entire release. See *Working With Software Channels* in [Oracle Linux Manager: Client Life Cycle Management Guide for Release 2.10](#) for further information.

Optimizing the Speed of Channel Synchronization

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

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

If you specify the `--latest` option with the `spacewalk-repo-sync` command or select the equivalent optional check box in the web interface, the server synchronizes only the latest packages that are available at the time of synchronization. The process does not remove older packages from the channel. If the synchronization interval is large, you might miss a particular version of a package, 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 select these options. Note that using the `--latest` option with a Ksplice channel is an exception to this rule because its packages are always cumulative.

 **Caution:**

DO NOT use the `--latest` option when synchronizing module-enabled channels such as `ol8_AppStream`. The mechanism that underlies this option is not module-aware and if used, will skip required packages.

3

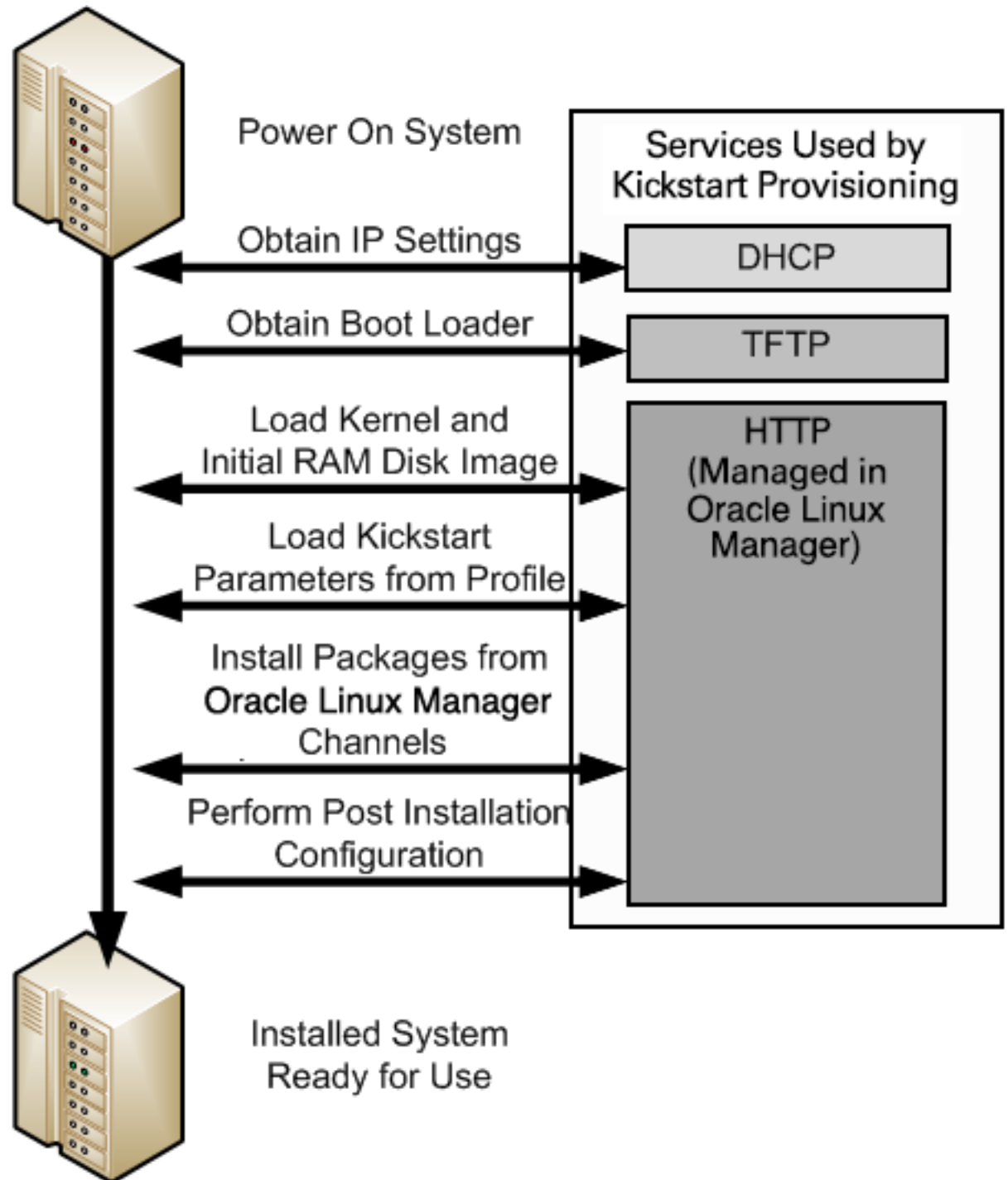
Provisioning Client Systems in Oracle Linux Manager

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

Configuring Oracle Linux Manager and kickstart to install and provision systems involves several steps, including creating a kickstart profile and using a kickstart file to install client systems. For complete instructions, see Provisioning Client Systems in [Oracle Linux Manager: Client Life Cycle Management Guide for Release 2.10](#) .

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

Figure 3-1 Kickstart Provisioning Process for a Client System that Uses DHCP and PXE Boot



The figure illustrates the following process:

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

2. Next, the client downloads the boot loader (`pxelinux`) from the TFTP server.
The boot loader then downloads the installation kernel and initial RAM-disk images from Oracle Linux Manager server.
3. The installation kernel runs Anaconda, which downloads the kickstart parameters, including the kickstart file, from Oracle Linux Manager server and uses this information to install packages from Oracle Linux Manager software channels, and perform any required system configuration.

If you are familiar with Cobbler, you might prefer to use the `cobbler` command to configure and manage kickstart profiles.

4

Maintaining Software Channels in Oracle Linux Manager

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

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

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

The following are a few methods that you can use to configure channels:

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

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

- Create an Oracle Linux Ksplice channel.

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

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

Some organizations have application requirements or policies that require some systems to remain at a particular update level of an Oracle Linux release. For example, perhaps certain production machines must remain locked at Oracle Linux 6 Update 7. Oracle releases new errata and patches for the latest release, such as Oracle Linux 6 Update 9. If upgrading to the latest update is not feasible for these systems, Oracle strongly recommends that you apply the latest security errata to avoid compromising the security of these systems.

One strategy is to maintain *latest* channels that contain the latest errata separately, usually without any client systems subscribed to these channels. You can then copy the

errata and dependent packages from these latest channels to other channels such as release and update-specific patch channels if you need to make the latest fixes available.

5

Overview of Development, Test, Acceptance, and Production Life Cycles

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

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

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

You can use either the Oracle Linux Manager web interface or the `spacecmd` command to clone channels in the following ways:

- Clone a channel in its current state, including all errata.
- Clone a channel in its original state, with only a selected set of errata.
- Clone a channel in its original state, with no errata.

Note the following additional points about cloning:

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

Note:

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

- When establishing a DTAP workflow, you might want to clone channels based on a particular date. You can use the `spacewalk-clone-by-date` command to ensure that the new channel contains packages and errata up to and including a specified date.
- You can blacklist or remove certain packages and choose types of errata to include or exclude when cloning.
- You can use the `spacewalk-clone-by-date` command to perform channel cloning based on a configuration file. You can then create a configuration file by using the `--sample_config` option and edit the file according to your requirements.
- In a DTAP environment, you might want to clone Oracle Linux channels on a periodic schedule, such as monthly or weekly. A typical DTAP workflow involves Oracle Linux Manager server pulling packages and the latest errata from the ULN repository for development channels. Assuming integration testing requires less than five days in this scenario, you could configure Oracle Linux Manager to create test channel clones of the updated development channels automatically each week. After the test channels have passed all quality assurance tests and have been accepted, you can then manually clone them as production channels.

While it is not possible to use the `spacewalk-clone-by-date` command to set up a DTAP workflow, you might find it more convenient to use the `spacewalk-manage-channel-lifecycle` command to manage the life cycle of a software channel from development through testing to production, in the following ways:

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

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

- After promoting a test channel to a production channel, you can put the accepted software configuration into production by subscribing your production systems to that channel on the Software Channel Subscriptions page of the web interface or by using the `spacecmd system_setbasechannel` command.
- To change the base channel for multiple production systems, you can map an existing base channel to a new base channel.

For more information, including instructions and examples, see Cloning Software Channels in [Oracle Linux Manager: Client Life Cycle Management Guide for Release 2.10](#).

6

Configuring System Groups to Standardize Configurations

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

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

You can also make changes to all of the servers in a system group by specifying the *group name*, for example, to change the base channel for all of the servers in a system group. Or, you can change the channel subscriptions for many systems at the same time to promote all of the systems that are currently subscribed to test channels to subscribe to production channels instead.

You can also perform Oracle Linux Manager management operations on a single system group or on the union or intersection of two or more system groups. See [Configuring System Groups to Manage Client Systems in Oracle Linux Manager: Client Life Cycle Management Guide for Release 2.10](#) .

7

Applying Errata to Systems and System Groups

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

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

- `system_listerrata`: Provides similar functionality. For example, list the errata that are available for a specified system.
- `erratadetails`: Displays additional information about a given erratum.
- `system_applyerrata`: Applies errata to multiple systems within a system group at the same time.

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