# Oracle® Fleet Patching and Provisioning

## Oracle Fleet Patching and Provisioning Administrator's Guide

23ai
F46763-03
January 2025

Oracle Fleet Patching and Provisioning Oracle Fleet Patching and Provisioning Administrator's Guide, 23ai

F46763-03

Copyright © 2016, 2025, Oracle and/or its affiliates.

Primary Author: Subhash Chandra

Contributors: Eric Belden

Contributors: Jonathan Creighton, Kamalesh Ramasamy, Kannan S. Viswanathan, Ludovico Caldara, Philippe Fierens, Rahul Desale, Ricardo A. Gonzalez, Ricardo Tamez Durandeau, Sampath Ravindhran, Siddharth Shankaran, Soo Huey Wong

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, software documentation, data (as defined in the Federal Acquisition Regulation), 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," "commercial computer software documentation," or "limited rights data" 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®, Java, MySQL, and NetSuite 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.

# Contents

## 4 Managing Gold Images and Working Copies



## 5 Patching and Upgrading Oracle Grid Infrastructure



## 6 Patching and Upgrading Oracle Database

## 7 Updating Oracle Exadata Infrastructure



## 8 Fleet Patching and Provisioning Postinstallation Tasks



## 9 Oracle Fleet Patching and Provisioning Use Cases

## A RHPCTL Command Reference

# Preface

Information in *Oracle Fleet Patching and Provisioning Administrator's Guide* applies to Oracle Fleet Patching and Provisioning as it runs on all platforms unless otherwise noted. Where necessary, this manual refers to platform-specific documentation.



## Audience

The *Oracle Fleet Patching and Provisioning Administrator's Guide* is intended for database administrators and system administrators who provision and maintain Oracle homes.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

**Access to Oracle Support**

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

## 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.

## Set Up Java Access Bridge to Implement Java Accessibility

Install Java Access Bridge so that assistive technologies on Microsoft Windows systems can use the Java Accessibility API.

Java Access Bridge is a technology that enables Java applications and applets that implement the Java Accessibility API to be visible to assistive technologies on Microsoft Windows systems.

Refer to *Java Platform, Standard Edition Accessibility Guide* for information about the minimum supported versions of assistive technologies required to use Java Access Bridge. Also refer to this guide to obtain installation and testing instructions, and instructions for how to use Java Access Bridge.

**Related Topics**

- *Java Platform, Standard Edition Java Accessibility Guide*

# Command Syntax

Refer to these command syntax conventions to understand command examples in this guide.

| Convention | Description |
|---|---|
| `$` | Bourne or BASH shell prompt in a command example. Do not enter the prompt as part of the command. |
| `%` | C Shell prompt in a command example. Do not enter the prompt as part of the command. |
| `#` | Superuser (root) prompt in a command example. Do not enter the prompt as part of the command. |
| `monospace` | UNIX command syntax |
| backslash \ | A backslash is the UNIX and Linux command continuation character. It is used in command examples that are too long to fit on a single line. Enter the command as displayed (with a backslash) or enter it on a single line without a backslash:<br><br>`dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b \  count=10000` |
| braces { } | Braces indicate required items:<br><br>`.DEFINE {macro1}` |
| brackets [ ] | Brackets indicate optional items:<br><br>`cvtcrt termname [outfile]` |
| ellipses ... | Ellipses indicate an arbitrary number of similar items:<br><br>`CHKVAL fieldname value1 value2 ... valueN` |
| *italic* | Italic type indicates a variable. Substitute a value for the variable:<br><br>*`library_name`* |
| vertical line \| | A vertical line indicates a choice within braces or brackets:<br><br>`FILE filesize [K\|M]` |

# 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. |

1

# Oracle Fleet Patching and Provisioning

Oracle Fleet Patching and Provisioning is a software lifecycle management method for provisioning and maintaining Oracle homes.

Oracle Fleet Patching and Provisioning (Oracle FPP) enables mass deployment and maintenance of standard operating environments for databases, clusters, and user-defined software types. With Oracle Fleet Patching and Provisioning, you can also install clusters and provision, patch, scale, and upgrade Oracle Grid Infrastructure and Oracle Database 12c release 2 (12.2), and later. Additionally, you can provision applications and middleware.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 19c, the feature formerly known as Rapid Home Provisioning (RHP) is now Oracle Fleet Patching and Provisioning (Oracle FPP).

- About Oracle Fleet Patching and Provisioning
  Oracle FPP is a **one-stop** solution for all your patching and upgrade needs for Oracle Databases, including single-instance Oracle Database, Oracle RAC, and Data Guard configurations.
- Oracle Fleet Patching and Provisioning Features
  Oracle Fleet Patching and Provisioning (Oracle FPP) provides various features to ease configuration and management tasks.
- Oracle Fleet Patching and Provisioning Architecture
  Oracle Fleet Patching and Provisioning (Oracle FPP) architecture consists of an Oracle FPP Server and any number of Oracle FPP Clients.
- Oracle Fleet Patching and Provisioning Concepts
  Oracle Fleet Patching and Provisioning (Oracle FPP) enables you to manage clientless targets, images, and working copies.

**Related Topics**

- About Deploying Oracle Databases Using Oracle Fleet Patching and Provisioning in *Oracle Real Application Clusters Installation Guide for Linux and UNIX*

## About Oracle Fleet Patching and Provisioning

Oracle FPP is a **one-stop** solution for all your patching and upgrade needs for Oracle Databases, including single-instance Oracle Database, Oracle RAC, and Data Guard configurations.

Oracle Fleet Patching and Provisioning (Oracle FPP) performs Out-Of-Place Gold Image-based patching, keeping in-line with the **Best Practices Recommendations** from the **Maximum Availability Architecture** team of Oracle. Oracle FPP supports standardization, rollback after failure or success, and revert and resume after failures.

Oracle FPP supports two deployment models, in the simplest, quickest, and most intuitive way:

1. The simple-to-use and no-setup needed Oracle FPP Local Mode, which is ready to be out-of-the-box for patching all configurations and deployments of Oracle Databases and Oracle Grid Infrastructure.
2. The central Oracle FPP Server, which can serve a fleet of databases and Grid Infrastructure from a single central server, making it easy to patch thousands of databases simultaneously. You can deploy a single Oracle FPP server for a given data center and use it to patch your entire fleet in that data center.

Continue to read further for more details on how to use Oracle FPP Local Mode and the Oracle FPP Server.

- As a local server (Oracle FPP Local Mode), that is the default configuration when you install Oracle Grid Infrastructure. Oracle FPP Local Mode operation enables you to perform Oracle Grid Infrastructure and Oracle Database patching operations on the local cluster in a simplified environment without having to register or deploy gold images. Deploy either the Oracle Grid Infrastructure or the Oracle Database patched home and run the patch operation using either the `rhpctl move gihome` or `rhpctl move database` command, specifying the source and destination paths instead of working copy names.
- As a central server (Oracle FPP Server), that stores and manages standardized images, called gold images. You can deploy gold images to any number of nodes across a data center. You can use the deployed homes to create new clusters and databases, and patch, upgrade, and scale existing installations.

  The server manages software homes on the cluster hosting the Oracle FPP Server, itself, Oracle FPP Clients, and can also manage Oracle Grid Infrastructure and Oracle Database installations running 11*g* release 2 (11.2.0.4) and later releases. The server can also manage `rhpclient`-less targets, which do not have Oracle Clusterware installed. Refer to My Oracle Support note 551141.1 for more information about Oracle Grid Infrastructure and Oracle Database upgrade paths.

  An Oracle FPP Server can provision new installations and can manage existing installations without any changes to the existing installations (such as no agent, daemon, or configuration prerequisites). Oracle FPP Servers also include capabilities for automatically sharing gold images among peer Oracle FPP Servers to support enterprises with geographically distributed data centers.

> **Note:**
>
> - Oracle does not support Oracle FPP on HP-UX and Windows operating systems.
> - The Oracle FPP Server does not manage operating system images on generic servers.
> - Combined Oracle FPP patching for Oracle Grid Infrastructure and Oracle Database is not supported for standalone configurations.
> - Oracle FPP enables you to manage and patch the whole Oracle software stack on Oracle Exadata, including Oracle Grid Infrastructure, Oracle Database, RoCE/IB switches, Cell Storage Servers, and compute nodes.

**Oracle Fleet Patching and Provisioning Advantages**

Deploying Oracle software using Oracle FPP has the following advantages:

- Ensures standardization and enables high degrees of automation with gold images and managed lineage of deployed software.

- Minimizes downtime by deploying new homes as images (working copies of gold images) out-of-place rolling patching, without disrupting active databases or clusters.
- Simplifies local maintenance operations using Oracle FPP Local Mode with a consistent API across database versions and deployment models.
- Provides REST API interface for Oracle FPP operations to ensure flexibility when integrating with bespoke and third-party orchestration engines.
- Reduces maintenance risk with built-in validations and a *dry run* mode to test the operations.
- Enables you to resume or restart the commands in the event of an unforeseen issue, reducing the impact of maintenance operations.
- Schedules and submits operations at a scheduled time instead of running the command immediately. The job scheduler performs the job and stores the metadata for the job, along with the current job status.
- Minimizes and often eliminates the impact of patching and upgrades, with features that include:
  - Options to do rolling patching to ensure continuous availability of the services.
  - Adaptive management of database sessions and OJVM during rolling patching.
  - Options for management of consolidated deployments.
- The deployment and maintenance operations enable customizations to include environment-specific actions into the automated workflow.

**Related Topics**

- My Oracle Support Note 551141.1

# Oracle Fleet Patching and Provisioning Features

Oracle Fleet Patching and Provisioning (Oracle FPP) provides various features to ease configuration and management tasks.

- Patching/software updates to Oracle Databases (Oracle RAC, Oracle RAC One Node, and Single Instance), Oracle Grid Infrastructure, Oracle Restart, and Oracle Exadata Engineered Systems (DBNode, Storage Cells, and Network).
- Software upgrades to Oracle Databases and Oracle Grid Infrastructure.

> **Note:**
>
> **Patching** is defined as a software revision within a major version, such as from 19.3 to 19.4, whereas **Upgrade** is defined as a revision of major version, such as from 19c to 23ai.

- Operating system configuration monitoring. Enables you to keep track of the OS configurations and keep track of changes made to the OS to help in troubleshooting.

The deployment and maintenance operations are extensible, allowing customizations to include environment-specific actions into the automated workflow.

### Oracle FPP Deployment Models

Oracle FPP supports two deployment models, in the simplest, quickest, and most intuitive way:

1. The simple-to-use and no-setup needed Oracle FPP Local Mode, which is ready to be out-of-the-box for patching all configurations and deployments of Oracle Databases and Oracle Grid Infrastructure.
2. The central Oracle FPP Server, which can serve a fleet of databases and Grid Infrastructure from a single central server, making it easy to patch thousands of databases simultaneously. You can deploy a single Oracle FPP server for a given data center and use it to patch your entire fleet in that data center.

### FPP Fleet Scale Job Framework

Oracle FPP supports submission of a number of patching operations through the job framework. Each operation is identified by a job associated with a job number. You can monitor, abort, or resume jobs after failures. Parent and child jobs are supported for simplified grouping of similar jobs.

### Oracle FPP Tag Framework

Starting with Oracle Grid Infrastructure 23ai, the **tag** framework offers a way to initiate a large no of operations using a single command through the use of tags associated with Oracle Database or Oracle Grid Infrastructure targets. You can collectively monitor jobs through simplified user interface to minimize and simplify human interactions, and to speed up operations.

### Global Fleet Standardization and Management

- *Sharing gold images between peer Oracle FPP Servers*: Large enterprises typically host multiple data centers and, within each data center, there may be separate network segments. In the Oracle FPP architecture, one Oracle FPP Server operates on a set of Oracle FPP Clients and `rhpclient`-less targets within a given data center (or network segment of a data center). Therefore each data center requires at least one Oracle FPP Server.

  While each data center may have some unique requirements in terms of the gold images that destination servers will use, the goal of standardization is using the same gold image across all data centers whenever possible. To that end, Oracle FPP supports peer-to-peer sharing of gold images to easily propagate gold images among multiple Oracle FPP Servers.
- *Gold image drift detection and aggregation*: After you provision a software home from a gold image, you may have to apply a patch directly to the deployed home. At this point the deployed home has drifted from the gold image. Oracle FPP provides two capabilities for monitoring and reporting drift:
  - Oracle FPP compares a specific home to its parent gold image and lists any patches that are applied to the home but that are not in the gold image.
  - Oracle FPP compares a specific gold image to all of its descendant homes and lists the aggregation of all patches applied to those homes that are not in the gold image. This provides a build specification for a new gold image that could be applied to all of the descendants of the original gold image, such that no patches will be lost from any of those deployments.

> **See Also:**
>
> – rhpctl query image for information about the `-drift` option for this command
>
> – rhpctl query workingcopy for information about the `-drift` option for this command

- *Configuration collection and reporting*: The Oracle FPP Server can collect and retain operating system configuration and the root file system contents of specified Oracle FPP Clients. If an Oracle FPP Client node is rendered unusable (for example, a user accidentally deletes or changes operating system configuration or the root file system), then it can be difficult to determine the problem and correct it. This feature automates the collection of relevant information, enabling restoration in the event of node failure.

**Flexibility and Extensibility**

- *RESTful API*: Oracle FPP provides a RESTful API for many common operations, including provisioning, patching, upgrading, and query operations.

> **See Also:**
>
> *Oracle Database REST API Reference*

- *Customizable authentication*: Host-to-host authentication in certain environments, particularly in compliance-conscious industries, such as financials and e-commerce, often uses technologies and products that are not supported, natively, by Oracle FPP. This feature allows integrating Oracle FPP authentication with the mechanisms in use at your data center.
- *Command scheduler*: The ability to schedule and bundle automated tasks is essential for maintenance of a large database estate. Oracle FPP supports scheduling tasks such as provisioning software homes, switching to a new home, and scaling a cluster. Also, you can add a list of clients to a command, facilitating large-scale operations.
- *Configurable connectivity*: As security concerns and compliance requirements increase, so do the restrictions on connectivity across the intranets of many enterprises. You can configure the small set ports used for communication between the Oracle FPP Server and its Clients, allowing low-impact integration into firewalled or audit-conscious environments.

**Other Oracle FPP Features**

- *Oracle Exadata Patching*: Oracle FPP enables you to manage and patch the whole Oracle software stack on Oracle Exadata, including Oracle Grid Infrastructure, Oracle Database, RoCE/IB switches, Cell Storage Servers, and compute nodes.
- *Vertical Oracle Exadata Patching*: Oracle FPP enables you to patch Oracle Exadata compute node and Oracle Grid Infrastructure vertically.
- *External Metadata Repository*: You can create an external database for Oracle FPP Server metadata. You can specify an external metadata repository, which can be an Oracle Database, during initial configuration.
- *Provision and manage any software home*: Oracle FPP enables you to create a gold image from any software home. You can then provision that software to any Oracle FPP Client or `rhpclient`-less target as a working copy of a gold image. The software may be any binary that you will run on an Oracle FPP Client or `rhpclient`-less target.

- *Adaptive Oracle RAC Rolling Patching for OJVM Deployments*: In a clustered environment, the default approach for Oracle FPP for patching a database is Oracle RAC rolling patching. However non-rolling may be required if the patched database home contains OJVM patches. In this case, Oracle FPP determines whether rolling patching is possible and does so, if applicable.

  > **Note:**
  >
  > Starting with Oracle Database 19c, you can perform OJVM patching in the rolling mode. For earlier database versions, Oracle FPP determines whether rolling patching is possible and performs rolling patching, if possible.

- *Pre-checks evaluation*: Before running any command, Oracle FPP checks various preconditions to ensure the command will succeed. However, some conditions cannot be detected prior to a command running. And, while Oracle FPP allows a failed command to be reverted or resumed after an error condition is corrected, it is preferable to address as many potential issues as possible before the command is run. The command evaluation mode will test the preconditions for a given command, without making any changes, and report potential problems and correct them before the command is actually run.
- *Provision, scale, patch, and upgrade Oracle Grid Infrastructure*: The Oracle FPP Server can provision Oracle Grid Infrastructure 11*g* release 2 (11.2.0.4) homes, and later, add or delete nodes from an Oracle Grid Infrastructure configuration, and can also be used to patch and upgrade Oracle Grid Infrastructure homes. In addition, there is a rollback capability that facilitates undoing a failed patch procedure.
- *Provision, scale, patch, and upgrade Oracle Database*: You can use Oracle FPP to provision, scale, and patch Oracle Database 11*g* release 2 (11.2.0.4), and later releases. Refer My Oracle Support note 551141.1 for more information about Grid Infrastructure and Oracle Database upgrade paths.

  When you provision such software, Oracle FPP offers additional features for creating various types of databases (such as Oracle RAC, single instance, and Oracle Real Application Clusters One Node (Oracle RAC One Node) databases) on different types of storage, and other options, such as using templates and creating Container Databases (CDBs). The Oracle FPP Server can add nodes to an Oracle RAC configuration, and remove nodes from an Oracle RAC configuration. Oracle FPP also improves and makes more efficient patching of database software, allowing for rapid and remote patching of the software, in most cases, without any downtime for the database.
- *Support for single-instance databases*: You can use Oracle FPP to provision, patch, and upgrade single-instance databases running on clusters or Oracle Restart, or on single, standalone nodes.
- *Combined Oracle Grid Infrastructure and Oracle Database patching*: When you patch an Oracle Grid Infrastructure deployment, Oracle FPP enables you to simultaneously patch the Oracle Database homes on the cluster, so you can patch both types of software homes within the same maintenance window. Combined patching reduces the brownout time.
- *Advanced patching capabilities*: When patching an Oracle Grid Infrastructure or Oracle Database home, Oracle FPP offers a batch mode that speeds the patching process by patching some or all nodes of a cluster in parallel and/or a specific node order, rather than sequentially.

  For Oracle Database homes, you can define disjoint sets of nodes. Each set of nodes is updated sequentially. By defining sets with reference to the database instances running on them, you can minimize the impact of rolling updates by ensuring that services are never

taken completely offline. A “smartmove” option is available to help define the sets of batches to meet this goal.

Integration with Application Continuity is another enhancement to help eliminate the impact of maintenance. This provides the ability to gracefully drain and relocate services within a cluster, completely masking the maintenance from users.

- *Notifications*:The Oracle FPP Server is the central repository for the software homes available to the data center. Therefore, it is essential that administrators throughout the data center be aware of changes to the inventory which might impact their areas of responsibility.

  Oracle FPP enables you and other users to subscribe to image series events. Anyone subscribed will be notified by email of any changes to the images available in a particular image series. Also, users can be notified by email when a working copy of a gold image is added to or deleted from a client.
- *Custom workflow support*: You can create actions for various Oracle FPP operations, such as importing images, adding or deleting working copies of the gold images, and managing a software home. You can define different actions for each operation, and further differentiate by the type of image to which the operation applies. Actions that you define can be performed before or after the given operation, and are run on the deployment the operation applies to, whether it is the Oracle FPP Server, an Oracle FPP Client, or an `rhpclient`-less target.
- *Resume failed operations*: If an operation, such as adding an image, provisioning a working copy of a gold image, or performing a scale, patch or upgrade fails, then Oracle FPP reports the error and stops. After the problem is corrected (for example, a directory permissions or ownership misconfiguration on a destination node), you can rerun the RHPCTL command that failed, and it will resume from the point of failure. This avoids redoing any work that may have been completed prior to the failure.
- *Audit command*: The Oracle FPP Server records all the Oracle FPP operations and also records their outcome (whether success or failure). An audit mechanism enables you to query the audit log in a variety of dimensions, and also to manage its contents and size.

**Related Topics**

- My Oracle Support Note 551141.1

# Oracle Fleet Patching and Provisioning Architecture

Oracle Fleet Patching and Provisioning (Oracle FPP) architecture consists of an Oracle FPP Server and any number of Oracle FPP Clients.

Oracle recommends deploying the Oracle FPP Server in a multi-node cluster so that it is highly available. Oracle FPP Server supports single-node deployment, but it is not recommended.

> **Note:**
>
> The Oracle FPP Server needs an Oracle Clusterware installation, thus you can not configure Oracle FPP Server on an Oracle Restart server.

The Oracle FPP Server cluster is a repository for all data, of which there are primarily two types:

- Gold images

- Metadata related to users, roles, permissions, and identities

The Oracle FPP Server acts as a central server for provisioning Oracle Database homes, Oracle Grid Infrastructure homes, and other application software homes, making them available to the cluster hosting the Oracle FPP Server and to the Oracle FPP Client clusters, and `rhpclient`-less targets.

Users operate on the Oracle FPP Server or Oracle FPP Client to request deployment of Oracle homes or to query gold images. When a user makes a request for an Oracle home, specifying a gold image, the Oracle FPP Client communicates with the Oracle FPP Server to pass on the request. The Oracle FPP Server processes the request by taking appropriate action to instantiate a copy of the gold image, and to make it available to the Oracle FPP Client cluster using available technologies such as Oracle Advanced Cluster File System (Oracle ACFS) and local file systems.

- Oracle Fleet Patching and Provisioning Server
  The Oracle Fleet Patching and Provisioning (Oracle FPP) Server is a highly available software provisioning system that uses Oracle Automatic Storage Management (Oracle ASM), Oracle Advanced Cluster File System (Oracle ACFS), application VIPs, and other components.
- Oracle Fleet Patching and Provisioning Clients and Targets
  The Oracle Fleet Patching and Provisioning (Oracle FPP) Client is part of Oracle Clusterware. Users operate on an Oracle FPP Client to perform tasks such as requesting deployment of Oracle homes and listing available gold images.
- Oracle FPP Metadata Repository
  Starting with Oracle Grid Infrastructure 23ai, you can configure either a local or an external metadata repository for your Oracle Fleet Patching and Provisioning (Oracle FPP) Server.

## Oracle Fleet Patching and Provisioning Server

The Oracle Fleet Patching and Provisioning (Oracle FPP) Server is a highly available software provisioning system that uses Oracle Automatic Storage Management (Oracle ASM), Oracle Advanced Cluster File System (Oracle ACFS), application VIPs, and other components.

The Oracle FPP Server primarily acts as a central server for provisioning Oracle homes and making them available to Oracle FPP Clients and `rhpclient`-less targets.

Features of the Oracle FPP Server:

- Efficiently stores gold images and image series for the managed homes, including separate binaries, and metadata related to users, roles, and permissions.
- Stores working copies and Oracle FPP Client information.
- Provides a list of available homes to Oracle FPP Clients and `rhpclient`-less targets upon request.
- Patch a software home once and then deploy the home to any Oracle FPP Client or any `rhpclient`-less targets, instead of patching every site.
- Provides the ability to report on existing deployments.
- Deploys homes on physical servers and virtual machines.
- Notifies subscribers of changes to image series.
- Maintains an audit log of all RHPCTL commands run.

# Oracle Fleet Patching and Provisioning Clients and Targets

The Oracle Fleet Patching and Provisioning (Oracle FPP) Client is part of Oracle Clusterware. Users operate on an Oracle FPP Client to perform tasks such as requesting deployment of Oracle homes and listing available gold images.

> **Note:**
>
> The Oracle FPP Server release must be later than or equal to the Oracle FPP Client software release, including the Release Update (RU). For example, if Oracle FPP Server is 23.4, the Oracle FPP Client must also be 23.4.

When a user requests an Oracle home specifying a gold image, the Oracle FPP Client communicates with the Oracle FPP Server to pass on the request. The Oracle FPP Server processes the request by instantiating a working copy of the gold image and making it available to the Oracle FPP Client using Oracle ACFS or a different local file system.

Oracle FPP Client has Oracle Clusterware and the additional `rhpclient` component enabled. This additional `rhpclient` component enables the Oracle FPP Client to initiate the tasks.

The Oracle FPP Client:

- Provides a list of available homes from the Oracle FPP Server.
- Has full functionality in Oracle Clusterware 12*c* release 2 (12.2) and can communicate with Oracle FPP Servers from Oracle Clusterware 12*c* release 2 (12.2), or later.

### Oracle Fleet Patching and Provisioning Targets

Computers of which Oracle Fleet Patching and Provisioning (Oracle FPP) is aware are known as `rhpclient`-less targets, which do not have the `rhpclient` component enabled.

Oracle FPP Servers can create new `rhpclient`-less targets, and can also install and configure Oracle Grid Infrastructure on such targets with only an operating system installed. Subsequently, Oracle FPP Server can provision database and other software on those `rhpclient`-less targets, perform maintenance, scale the cluster, in addition to many other operations. All Oracle FPP commands are run on the Oracle FPP Server.

`rhpclient`-less targets running the Oracle FPP Client in Oracle Clusterware 12*c* release 2 (12.2), and later, may also run many of the Oracle FPP commands to request new software from the Oracle FPP Server and initiate maintenance themselves, among other tasks.

> **Note:**
>
> The Oracle FPP Server communicates with Oracle Grid Infrastructure Clusters at version 12.2.0.1 and later through an Oracle FPP Client that can be configured and started up on the destination cluster. The Oracle FPP Client is not supported for `rhpclient`-less targets on Oracle Grid Infrastructure version 12.1 and earlier, on all versions of Oracle Restart and database standalone `rhpclient`-less targets, such as database homes without an Oracle Grid Infrastructure home.

**Related Topics**

- Creating an Oracle Fleet Patching and Provisioning Client
  Users operate on an Oracle Fleet Patching and Provisioning (Oracle FPP) Client to perform tasks such as requesting deployment of Oracle homes and querying gold images.

## Oracle FPP Metadata Repository

Starting with Oracle Grid Infrastructure 23ai, you can configure either a local or an external metadata repository for your Oracle Fleet Patching and Provisioning (Oracle FPP) Server.

You can create an external metadata repository and provide its information during Oracle FPP setup or upgrade.

Oracle FPP Server has the following metadata repository configuration workflows:

- If the external metadata repository option is specified, then you must provide a database connection string. The Oracle FPP Server automatically creates a schema in the database to store metadata. Use the `-dbtype EXTERNAL -connstr` *`db_connect_string`* `-dbadmin` *`db_user`* parameters to configure an external metadata repository.
- If the external metadata repository option is not specified, then Oracle FPP Server automatically creates a database to store metadata.

If you do not use Oracle FPP self-upgrade feature, then you need to upgrade Oracle FPP metadata repository separately because database credentials can not be passed to the installer. In such a case, Oracle FPP Server remains disabled until the metadata repository upgrade is complete. If you try to start the Oracle FPP Server, then you will get an error asking to run the upgrade command.

> **Note:**
>
> Multiple Oracle FPP Servers can share the same CDB, however, they require separate metadata repositories. You must create a separate PDB for each Oracle FPP Server to avoid schema name conflicts.

**Related Topics**

- Creating an Oracle Fleet Patching and Provisioning Server
  The Oracle FPP Server uses a repository that you create in an Oracle ACFS file system in which you store all the software homes that you want to make available to clients and `rhpclient`-less targets.

# Oracle Fleet Patching and Provisioning Concepts

Oracle Fleet Patching and Provisioning (Oracle FPP) enables you to manage clientless targets, images, and working copies.

- Images of Oracle Fleet Patching and Provisioning
  You can easily copy an image of an Oracle home to a new host on a new file system to serve as an active usable Oracle home.
- Working Copies of Oracle Fleet Patching and Provisioning
  Working copy is a copy of the gold image that you use to provision the software on an Oracle Fleet Patching and Provisioning (Oracle FPP) Client or an `rhpclient`-less target.

# Images of Oracle Fleet Patching and Provisioning

You can easily copy an image of an Oracle home to a new host on a new file system to serve as an active usable Oracle home.

By default, when you create a gold image using either `rhpctl import image` or `rhpctl add image`, the image is ready to provision new homes, called working copies. However, under certain conditions, you may want to restrict access to images and require someone to test or validate the image before making it available for general use.

You can also create a set of gold images on the Oracle Fleet Patching and Provisioning Server that can be collectively categorized as a gold image series which relate to each other, such as identical release versions, gold images published by a particular user, or images for a particular department within an organization.

**Related Topics**

- Image State
  Am image state is a way to restrict provisioning of an image for users with specified roles.
- Image Series
  An image series is a convenient way to group different gold images into a logical sequence.
- Image Type
  When you add or import a gold image, you must specify an image type.

# Working Copies of Oracle Fleet Patching and Provisioning

Working copy is a copy of the gold image that you use to provision the software on an Oracle Fleet Patching and Provisioning (Oracle FPP) Client or an `rhpclient`-less target.

By default, when you create a gold image using either `rhpctl import image` or `rhpctl add image`, the image is ready to provision working copies. You can use the `rhpctl add workingcopy` command to add a working copy to a client cluster.

After you create and import a gold image, you can provision software by adding a copy of the gold image (called a working copy) on the Fleet Patching and Provisioning Server, on a Fleet Patching and Provisioning Client, or an `rhpclient`-less target.

**Related Topics**

- Image State
  Am image state is a way to restrict provisioning of an image for users with specified roles.
- Image Series
  An image series is a convenient way to group different gold images into a logical sequence.
- Image Type
  When you add or import a gold image, you must specify an image type.

2

# Oracle Fleet Patching and Provisioning Local Mode

When you install Oracle Grid Infrastructure, the Oracle FPP is configured, by default, in the Oracle FPP local mode to support the local switch home capability.

- About Oracle Fleet Patching and Provisioning Local Mode
  Oracle Fleet Patching and Provisioning (Oracle FPP) is configured in the Oracle FPP Local Mode mode by default when you install Oracle Grid Infrastructure.
- Performing Patching Using Oracle FPP Local Mode
  Use this procedure to patch Oracle Grid Infrastructure and Oracle Database using Oracle Fleet Patching and Provisioning (Oracle FPP) Local Mode.
- Storing User-Action Scripts in OCR For Oracle FPP Local Mode
  Starting with Oracle Grid Infrastructure 23ai, you can store the Oracle FPP Local Mode user action scripts in the Oracle Cluster Registry (OCR).
- Switching from Oracle FPP Local Mode to Oracle FPP Central Server
  Oracle Grid Infrastructure has Fleet Patching and Provisioning (Oracle FPP) is configured automatically as Oracle FPP Local Mode.

## About Oracle Fleet Patching and Provisioning Local Mode

Oracle Fleet Patching and Provisioning (Oracle FPP) is configured in the Oracle FPP Local Mode mode by default when you install Oracle Grid Infrastructure.

> **Note:**
>
> - Oracle FPP Local Mode does not require a Java container to perform Oracle Grid Infrastructure and Oracle Database patching operations.

Oracle FPP Local Mode does not require a separate setup. You can use it to manage Oracle Database and Oracle Grid Infrastructure patching only on a local cluster. Oracle FPP Local Mode offers a reduced set of features as compared to Oracle FPP Central Server.

The Oracle FPP Local Mode operation enables you to perform Oracle Grid Infrastructure and Oracle Database patching operations on the local cluster. Deploy either the Oracle Grid Infrastructure or the Oracle Database patched home and run the patch operation using either the `rhpctl move gihome` or `rhpctl move database` command, specifying the source and destination paths.

The Oracle FPP Local Mode supports Oracle Database 11*g* (11.2.0.4), 12*c* (12.1.0.2), 12*c* (12.2.0.1), or later in a clustered environment. In a standalone (non-clustered) environment, you can patch only the database home and the database home must be Oracle Database 18c or later.

The Oracle FPP Local Mode runs locally on the deployment and does not require an Oracle Fleet Patching and Provisioning Server in the architecture. If you configure an Oracle Fleet

Patching and Provisioning Server, then you must remove the Oracle FPP Local Mode configuration.

# Performing Patching Using Oracle FPP Local Mode

Use this procedure to patch Oracle Grid Infrastructure and Oracle Database using Oracle Fleet Patching and Provisioning (Oracle FPP) Local Mode.

> **Note:**
>
> Oracle FPP Local Mode supports only one patching operation at a time. Wait for the current operation to finish before running another patch operation.

1. Deploy an Oracle Grid Infrastructure home using Oracle FPP Local Mode.

   ```
   $ rhpctl deploy home -path /u01/app/23.6.0/grid -zip /tmp/images/
   grid_home.zip
   ```

2. Switch from the old Oracle Grid Infrastructure home to the new Oracle Grid Infrastructure home using Oracle FPP Local Mode.

   ```
   $ rhpctl move gihome -desthome /u01/app/23.6.0/grid -sourcehome /u01/app/
   23.0.0/grid
   ```

   > **Note:**
   >
   > - During Oracle FPP Local Mode patching, you must run the patching command only on one node. The patching completes automatically on all other nodes.
   > - If any command fails in the first run, then you must re-run the command from the same node where it was started initially.

3. Deploy an Oracle Database home using Oracle FPP Local Mode.

   ```
   $ rhpctl deploy home -path /u01/app/oracle/product/23.6.0/dbhome_1 -
   zip /tmp/images/db_home.zip -sourcehome /u01/app/oracle/product/23.0.0/
   dbhome_1
   ```

4. Switch from the old Oracle Database home to the new Oracle Database home using Oracle FPP Local Mode.

   ```
   $ rhpctl move database -desthome /u01/app/oracle/product/23.6.0/dbhome_1 -
   sourcehome /u01/app/oracle/product/23.0.0/dbhome_1
   ```

5. If the patching fails or you want to switch back to the old Oracle Grid Infrastructure home, then run the following command, from the destination home as the `grid` user, to rollback the patch.

```
$ rhpctl move gihome -desthome /u01/app/23.0.0/grid -sourcehome /u01/app/
23.6.0/grid
```

**Related Topics**

- Creating an Oracle Fleet Patching and Provisioning Server
  The Oracle FPP Server uses a repository that you create in an Oracle ACFS file system in which you store all the software homes that you want to make available to clients and `rhpclient`-less targets.
- rhpctl deploy home
  Use the `rhpctl deploy home` command to deploy Oracle Grid Infrastructure and Oracle Database homes from gold images in Oracle FPP Local Mode.
- rhpctl move gihome
  Moves the Oracle Grid Infrastructure software stack from one home to another.
- rhpctl move database
  Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.

# Storing User-Action Scripts in OCR For Oracle FPP Local Mode

Starting with Oracle Grid Infrastructure 23ai, you can store the Oracle FPP Local Mode user action scripts in the Oracle Cluster Registry (OCR).

If you store user action scripts in OCR, then you do not have to specify the user action script location every time you patch Oracle Grid Infrastructure or Oracle Database.

1. As the `grid` user, use the `rhphelper` command to set the user action script location to OCR:

   *Grid_home*`/srvm/admin/rhphelper` *Grid_home version* `-setuseractionloc uascriptloc=`*script_location*

   This command creates a key with value set to user action script location under a subtree `OCR.GRIDHOME.FPPLOCAL.UALOC` , and the key is readable by both the `oracle` user and the `grid` user. Leave the location empty to reset the value.
2. Optionally, you can specify the `-ualoc` parameter to override the OCR location when you patch Oracle Database or Oracle Grid Infrastructure using Oracle FPP Local Mode.

# Switching from Oracle FPP Local Mode to Oracle FPP Central Server

Oracle Grid Infrastructure has Fleet Patching and Provisioning (Oracle FPP) is configured automatically as Oracle FPP Local Mode.

> **Note:**
>
> Fleet Patching and Provisioning Local Mode is started automatically when patching command is invoked and the dependent resources are set up during the automatic startup process.

To switch from Oracle FPP Local Mode to the Oracle FPP Central Server mode (to manage Oracle FPP Clients and `rhpclient`-less targets), you must delete the current Oracle FPP Local Mode as follows:

```
$ srvctl stop rhpserver
$ srvctl remove rhpserver
```

Proceed with the steps described in "Creating an Oracle Fleet Patching and Provisioning Server" to create the central-mode Oracle FPP Server.

**Related Topics**

- Creating an Oracle Fleet Patching and Provisioning Server
  The Oracle FPP Server uses a repository that you create in an Oracle ACFS file system in which you store all the software homes that you want to make available to clients and `rhpclient`-less targets.

3

# Oracle Fleet Patching and Provisioning Central Mode

Configuring Oracle Fleet Patching and Provisioning (Oracle FPP) in central mode involves creating an Oracle FPP Server, adding gold images to the server, and creating working copies of gold images to provision software.

After you install and configure Oracle Grid Infrastructure, you can configure and start using Oracle FPP. You must create an Oracle FPP Server where you create and store gold images of database and other software homes.

- Configuring Oracle Fleet Patching and Provisioning Server
  Oracle Fleet Patching and Provisioning (Oracle FPP) Server configuration includes configuring storage, network, external Oracle Database, and creating an Oracle FPP resource.
- Configuring Oracle Fleet Patching and Provisioning Clients
  Oracle Fleet Patching and Provisioning (Oracle FPP) client configuration includes configuring network, creating client data file, and creating an Oracle FPP client.
- Oracle Fleet Patching and Provisioning Server Self-upgrade
  Perform these steps when upgrading Oracle Grid Infrastructure to 23ai using the Oracle Fleet Patching and Provisioning (Oracle FPP) Server self-upgrade feature.
- Migrating Metadata from GIMR to Metadata Repository for Manual Upgrade
  Use this procedure to import the metadata in to a new Oracle FPP metadata repository after manually upgrading to Oracle Grid Infrastructure 23ai.
- Oracle Fleet Patching and Provisioning Backup and Restore
  Create a backup of Oracle FPP Server to restore in the event of a failure, or to relocate the Oracle FPP Server to a different hardware.

## Configuring Oracle Fleet Patching and Provisioning Server

Oracle Fleet Patching and Provisioning (Oracle FPP) Server configuration includes configuring storage, network, external Oracle Database, and creating an Oracle FPP resource.

- Server Configuration Checklist for Oracle Fleet Patching and Provisioning
  Use this checklist to check minimum server configuration requirements for Oracle Fleet Patching and Provisioning (Oracle FPP).
- Oracle Fleet Patching and Provisioning Communication Ports
  Configure communication ports for Oracle Fleet Patching and Provisioning (Oracle FPP) Server, clients, and `rhpclient`-less targets.
- Creating an Oracle Fleet Patching and Provisioning Server
  The Oracle FPP Server uses a repository that you create in an Oracle ACFS file system in which you store all the software homes that you want to make available to clients and `rhpclient`-less targets.
- Switching from GNS to Oracle FPP Application VIP
  Use this procedure to switch from GNS to Oracle FPP application VIP for existing Oracle FPP Server and its clients.

# Server Configuration Checklist for Oracle Fleet Patching and Provisioning

Use this checklist to check minimum server configuration requirements for Oracle Fleet Patching and Provisioning (Oracle FPP).

**Table 3-1 Server Configuration Checklist for Oracle Fleet Patching and Provisioning**

| Check | Task |
|---|---|
| Oracle Grid Infrastructure installation | Install Oracle Grid Infrastructure on a new cluster on which you want to configure Oracle FPP.<br><br>**Note:**<br>The Oracle FPP Server needs an Oracle Clusterware installation, thus you can not configure Oracle FPP Server on an Oracle Restart server. |
| Operating System Kernel version | Install or upgrade the operating system kernel to a version for which an Oracle ACFS kernel module is already built. |
| Metadata Repository configuration | The Oracle FPP Server provides two options for metadata repository configuration:<br>• You can create an external metadata repository. This external metadata repository can be an Oracle Database.<br>• If external metadata repository is not configured, then Oracle FPP Server automatically creates an Oracle Database to store metadata. |
| Oracle FPP server storage | Allocate a minimum of 100 GB additional disk space to the Oracle Automatic Storage Management (Oracle ASM) disk group that is used by the Oracle FPP Server. |
| Oracle FPP server network | Create one Application VIP Managed by Oracle Clusterware.<br><br>**Note:**<br>The Highly Available Grid Naming Service feature of Grid Naming Service (GNS) in Oracle Grid Infrastructure is deprecated in Oracle Database 23ai. |
| Firewall | Make sure that the ports used by Oracle FPP Server and Client are not filtered by firewalls. Please refer to Table 2-2 Fleet Patching and Provisioning Communication Ports |
| Custom certificates | Optionally, create custom security certificates to specify during Oracle FPP Server or Oracle FPP Client setup for server-server and client-server communication. You can also change the security certificate after initial configuration. |

### Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide for Linux*

# Oracle Fleet Patching and Provisioning Communication Ports

Configure communication ports for Oracle Fleet Patching and Provisioning (Oracle FPP) Server, clients, and `rhpclient`-less targets.

The Oracle Fleet Patching and Provisioning Server communicates with Oracle Fleet Patching and Provisioning Clients and `rhpclient`-less targets using the following ports, several of which you can configure, as described in the below tables. Additionally, differences in ports used when communicating with Oracle Fleet Patching and Provisioning Clients versus `rhpclient`-less targets are noted.

> **Note:**
>
> - The Oracle FPP ports that allow an ephemeral range or a custom port, you must ensure that the ports are open bidirectionally.
> - You need six ports for each working copy operation on the same client. For example, if you perform two add working copy operations on the same client, then you need 12 open ports.

**Table 3-2 Ports Open on Oracle FPP Server to Communicate with Oracle FPP Client**

| Protocol | Port | Modifiable | Purpose | Description |
|---|---|---|---|---|
| UDP | 53 | No | GNS | The Oracle FPP clients use GNS to locate the Oracle FPP Server.<br><br>**Note:**<br>The Highly Available Grid Naming Service feature of Grid Naming Service (GNS) in Oracle Grid Infrastructure is deprecated in Oracle Database 23ai. |
| TCP | 8896 | Yes. Use the `srvctl modify rhpserver -port` *`port_number`* or `srvctl modify rhpserver -port` *`port_number`* command to modify this port.<br>It requires a restart. | JMX Server | The Oracle FPP Server uses this port to communicate with the Oracle FPP clients. |

**Table 3-2 (Cont.) Ports Open on Oracle FPP Server to Communicate with Oracle FPP Client**

| Protocol | Port | Modifiable | Purpose | Description |
|---|---|---|---|---|
| TCP | Ephemeral range or a custom port | Yes. Use the `srvctl modify rhpserver -pl_port` *`port_number`* or `srvctl modify rhpserver -clport` *`port_number`* command to modify this port. | Command Progress Listener | The Oracle FPP Server opens a random port from an ephemeral range to monitor progress on the client or `rhpclient`-less target. The Oracle FPP Server can also use a fixed port you specify using `srvctl modify rhpserver -pl_port` *`port_number`*, `srvctl modify rhpserver -clport` *`port_number`*, and multiplex the fixed port across clients or `rhpclient`-less targets. |

**Table 3-3 Ports Open on Oracle FPP Client to Communicate with Oracle FPP Server**

| Protocol | Port | Modifiable | Purpose | Description |
|---|---|---|---|---|
| TCP | 22 | Yes | SSH | The Oracle FPP Client requires an SSH port open during initial deployment of Oracle Grid Infrastructure. After you add the cluster as an Oracle FPP Client, Oracle FPP uses the JMX port for communication between Oracle FPP Client and Oracle FPP Server. The default JMX port is 8896. |
| TCP | 8896 | Yes. Use the `srvctl modify rhpclient -port` *`port_number`* command to modify this port. It requires a restart. | JMX Server | The Oracle FPP Client uses this port to communicate with the Oracle FPP Server. |
| TCP | Ephemeral range or a custom range | Yes. Use the `srvctl modify rhpserver -port_range` *`port_number_range`* command to modify this port. | Image Transfer | The Oracle FPP Server uses six ports chosen from an ephemeral range, or six ports from the range you define to transfer gold images to the Oracle FPP clients. |

**Note:**

The Oracle FPP Server must be able to connect to the port specified in the port range.

**Table 3-4 Ports Open on Oracle FPP Server to Communicate with `rhpclient`-Less Targets**

| Protocol | Port | Modifiable | Purpose | Description |
|---|---|---|---|---|
| TCP | Ephemeral range or a custom range | Yes. Use the `srvctl modify rhpserver -pl_port` *`port_number`* or `srvctl modify rhpserver -clport` *`port_number`* command to modify this port. | Command Progress Listener | The Oracle FPP Server opens a random port from an ephemeral range to monitor progress on the client or `rhpclient`-less target. The Oracle FPP Server can also use a fixed port you specify using `srvctl modify rhpserver -pl_port` *`port_number`*, `srvctl modify rhpserver -clport` *`port_number`*, and multiplex the fixed port across clients or `rhpclient`-less targets. |

**Table 3-5 Ports Open on `rhpclient`-Less Targets to Communicate with Oracle FPP Server**

| Protocol | Port | Modifiable | Purpose | Description |
|---|---|---|---|---|
| TCP | 22 | Yes | SSH | The Oracle FPP Client requires an SSH port open during initial deployment of Oracle Grid Infrastructure. After you add the cluster as an Oracle FPP Client, Oracle FPP uses the JMX port for communication between Oracle FPP Client and Oracle FPP Server. The default JMX port is 8896. |
| TCP | Ephemeral range or a custom range | Yes. Use the `srvctl modify rhpserver -port_range` *`port_number_range`* command to modify this port. | Image Transfer | The Oracle FPP Server uses a range of six ports, from an ephemeral range or six ports from the range you define, to transfer gold images to the Oracle FPP clients. |

**Table 3-6 Ports Open on Main Oracle FPP Server to Communicate with Peer Servers**

| Protocol | Port | Modifiable | Purpose | Description |
|---|---|---|---|---|
| UDP | 53 | No | GNS | The Oracle FPP clients and peer servers use GNS to locate the main Oracle FPP Server. |
| TCP | 8896 | Yes. Use the `srvctl modify rhpserver -port` *`port_number`* command to modify this port.<br>It requires a restart. | JMX Server | The Oracle FPP Server uses this port to communicate with the Oracle FPP clients and peer servers. |
| TCP | Ephemeral range or a custom range | Yes. Use the `srvctl modify rhpserver -port_range` *`port_number_range`* command to modify this port. | Image Transfer | The Oracle FPP Server uses a range of six ports, from an ephemeral range or six ports from the range you define, to transfer gold images to the Oracle FPP clients. |

**Table 3-7 Ports Open on Peer Servers to Communicate With Main Oracle FPP Server**

| Protocol | Port | Modifiable | Purpose | Description |
|---|---|---|---|---|
| UDP | 53 | No | GNS | The Oracle FPP clients and peer servers use GNS to locate the main Oracle FPP Server. |
| TCP | 8896 | Yes. Use the `srvctl modify rhpserver -port` *`port_number`* command to modify this port.<br>It requires a restart. | JMX Server | The Oracle FPP Server uses this port to communicate with the Oracle FPP clients and peer servers. |
| TCP | Ephemeral range or a custom range | Yes. Use the `srvctl modify rhpserver -port_range` *`port_number_range`* command to modify this port. | Image Transfer | The Oracle FPP Server uses a range of six ports, from an ephemeral range or six ports from the range you define, to transfer gold images to the Oracle FPP clients. |

# Creating an Oracle Fleet Patching and Provisioning Server

The Oracle FPP Server uses a repository that you create in an Oracle ACFS file system in which you store all the software homes that you want to make available to clients and `rhpclient`-less targets.

> **Note:**
>
> When you install Oracle Grid Infrastructure, the Oracle Fleet Patching and Provisioning (Oracle FPP) Server is configured, by default, in the Oracle FPP Local Mode to support the *local switch home* capability. To configure the central Oracle FPP Server, then you must remove the current Oracle FPP Local Mode configuration.

1. Use the Oracle ASM Configuration Assistant (Oracle ASMCA) to create an Oracle ASM disk group on the Oracle FPP Server to store software.

   ```
   $ Grid_home/bin/asmca
   ```

   Because this disk group is used to store software, Oracle recommends a minimum of 100 GB for this disk group.

   > **Note:**
   >
   > You must set Oracle ASM Dynamic Volume Manager (Oracle ADVM) compatibility settings for this disk group to `19.0` or later.

2. Provide a mount path that exists on all nodes of the cluster. The Oracle FPP Server uses this path to mount gold images. You must ensure that the `grid` user owns the mount point.

   ```
   $ mkdir -p storage_path
   ```

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, you can create an external metadata repository. This external metadata repository can be an Oracle Database.

3. Optional: As the `root` user, add the Grid Naming Service Virtual IP Address (GNS VIP) without zone delegation.

```
# srvctl add gns -vip myhost-gnsvip3
# srvctl start gns
# srvctl status gns
GNS is running on node myhost01.
GNS is enabled on node myhost01.
```

> **Note:**
>
> The Highly Available Grid Naming Service feature of Grid Naming Service (GNS) in Oracle Grid Infrastructure is deprecated in Oracle Database 23ai.

4. Optional: If you do not want to use GNS, then you can use the Oracle FPP VIP address by configuring it while adding the Oracle FPP Server.

> **Note:**
>
> Oracle recommends that you use Oracle FPP VIP address to configure your Oracle FPP Server.

5. Remove any existing Oracle FPP Local Mode configuration from your cluster.

```
# srvctl stop rhpserver
# srvctl remove rhpserver
```

6. To configure an internal metadata repository, setup an Enterprise Edition of single-instance Oracle Database or Oracle RAC One Node (with own license) across all active nodes in the cluster and associate the repository with your Oracle FPP Server.

```
$GRID_HOME/crs/install/reposScript.sh -db_home=database_home -
mode="Install" -diskgroup=disk_group_name
```

7. Optional: Generate a P12 file from the security certificate that you want to use for the Oracle FPP Server configuration.

```
$ openssl pkcs12 -export -inkey ${SERVER_KEY} -passin pass:${PASS} -in $
{SERVER_CRT}
-name orakey -certfile ${CA_CRT} -caname cakey -out ${P12_FILE} -password
pass:${P12_PASS}
```

8. Create the Oracle FPP Server resource.

```
# Grid_home/bin/srvctl add rhpserver -storage storage_path
-diskgroup disk_group_name -enableTLS YES -p12certpath
P12_file_absolute_path
-certname custom_cert_name -dbtype FPPDB [additional options] -
rhpsvip_address vip_name
```

Use the `enableTLS` parameter to enable TLS and specify the custom security certificate name.

> **Note:**
>
> You must provide the VIP name, instead of the VIP address, for the `-rhpsvip_address` parameter.

9. Start the Oracle FPP Server.

```
$ Grid_home/bin/srvctl start rhpserver
```

After you start the Oracle FPP Server, use the Fleet Patching and Provisioning Control (RHPCTL) utility to further manage Oracle FPP.

**Related Topics**

- RHPCTL Command Reference
  Use the Oracle Fleet Patching and Provisioning Control (RHPCTL) utility to manage Oracle Fleet Patching and Provisioning in your cluster.
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Automatic Storage Management Administrator's Guide*

## Switching from GNS to Oracle FPP Application VIP

Use this procedure to switch from GNS to Oracle FPP application VIP for existing Oracle FPP Server and its clients.

1. Stop the Oracle FPP Server, modify the server to use the application VIP, and start the server.

```
$ srvctl stop rhpserver
$ srvctl modify rhpserver -rhpsvip_address vip_name
$ srvctl start rhpserver
```

> **Note:**
>
> You must provide the VIP name, instead of the VIP address, for the `-rhpsvip_address` parameter.

2. Confirm that the endpoint is updated to Oracle FPP application VIP for the server.

```
$ srvctl config rhpserver
….
….
Endpoint : rhps-vip:vip_address
….
….
```

> **Note:**
>
> The existing Oracle FPP Clients continue to work if the GNS is still configured. The Oracle FPP Clients will continue to use GNS until you configure the clients to use Oracle FPP application VIP by following the steps in this procedure.

3. On the Oracle FPP Server as the Grid home owner, create a new client data file.

```
$ rhpctl export client -client client_name -clientdata new_file_location
```

4. Copy the generated client data file to the Oracle FPP Client.
5. As the `root` user, configure the existing Oracle FPP Client to use the new client data file.

```
# srvctl modify rhpclient -clientdata client_data_file_path
```

6. Restart Oracle FPP Client.

```
$ srvctl stop rhpclient
$ srvctl start rhpclient
```

7. Repeat Steps 4 through 6 on each Oracle FPP Client.
8. Once all the clients are configured to use Oracle FPP application VIP, you can remove GNS using the `srvctl remove gns` command.

# Configuring Oracle Fleet Patching and Provisioning Clients

Oracle Fleet Patching and Provisioning (Oracle FPP) client configuration includes configuring network, creating client data file, and creating an Oracle FPP client.

- Creating an Oracle Fleet Patching and Provisioning Client
  Users operate on an Oracle Fleet Patching and Provisioning (Oracle FPP) Client to perform tasks such as requesting deployment of Oracle homes and querying gold images.
- Enabling and Disabling Fleet Patching and Provisioning Clients
  On the Fleet Patching and Provisioning Server, you can enable or disable a Fleet Patching and Provisioning Client.
- Deleting a Fleet Patching and Provisioning Client
  Use the following procedure to delete a Fleet Patching and Provisioning Client.

## Creating an Oracle Fleet Patching and Provisioning Client

Users operate on an Oracle Fleet Patching and Provisioning (Oracle FPP) Client to perform tasks such as requesting deployment of Oracle homes and querying gold images.

1. On the Oracle Fleet Patching and Provisioning Server as the Grid home owner, create the client data file, as follows:

   ```
   $ rhpctl add client -client client_cluster_name [-clusternamealias
   cluster_name_alias] -toclientdata path
   ```

   RHPCTL creates the client data file in the directory path you specify after the `-toclientdata` flag. The name of the client data file is *client_cluster_name*`.xml`.

   > **Note:**
   >
   > Oracle recommends that you specify a unique *client_cluster_name* and it must match the cluster name of the client cluster where you run step 4. If the client cluster name is not unique, then you can specify a cluster name alias.

2. Copy the client data file that you created in the previous step to a directory on the client cluster that has read/write permissions to the Grid home owner on the Oracle Fleet Patching and Provisioning Client.
3. Optional: Generate a P12 file from the security certificate that you want to use for the Oracle FPP Server configuration.

   ```
   $ openssl pkcs12 -export -inkey ${SERVER_KEY} -passin pass:${PASS} -in $
   {SERVER_CRT}
   -name orakey -certfile ${CA_CRT} -caname cakey -out ${P12_FILE} -password
   pass:${P12_PASS}
   ```

4. Create the Oracle FPP Client by running the following command as `root` on the client cluster:

   ```
   # srvctl add rhpclient -clientdata path_to_client_data
   -diskgroup disk_group_name -storage base_path
   -enableTLS YES -p12certpath P12_file_absolute_path
   -certname custom_cert_name [additional options]
   ```

   If you want to provision working copies to Oracle ACFS storage on this cluster, and you have already created a disk group for this purpose, then specify this disk group in the preceding command. In this case, also specify a storage path which will be used as a base path for all mount points when creating Oracle ACFS file systems for storing working copies.

   > **Note:**
   >
   > Once you configure a disk group on a Fleet Patching and Provisioning Client, you cannot remove it from or change it in the Fleet Patching and Provisioning Client configuration. The only way you can do either (change or remove) is to completely remove the Fleet Patching and Provisioning Client using the `srvctl remove client` command, and then add it back with a different disk group, if necessary. Before you remove a Fleet Patching and Provisioning Client, ensure that you remove all registered users from this cluster and all working copies provisioned on this cluster.

Use the `enableTLS` parameter to enable TLS and specify the custom security certificate name.

5. Start the Fleet Patching and Provisioning Client, as follows:

```
$ srvctl start rhpclient
```

6. Check the status of the Fleet Patching and Provisioning Client, as follows:

```
$ srvctl status rhpclient
```

**Related Topics**

- *Oracle Clusterware Administration and Deployment Guide*
- RHPCTL Command Reference
  Use the Oracle Fleet Patching and Provisioning Control (RHPCTL) utility to manage Oracle Fleet Patching and Provisioning in your cluster.

## Enabling and Disabling Fleet Patching and Provisioning Clients

On the Fleet Patching and Provisioning Server, you can enable or disable a Fleet Patching and Provisioning Client.

Fleet Patching and Provisioning Clients communicate with the Fleet Patching and Provisioning Server for all actions. You cannot run any RHPCTL commands without a connection to a Fleet Patching and Provisioning Server.

To enable or disable a Fleet Patching and Provisioning Client, run the following command from the Fleet Patching and Provisioning Server cluster:

```
$ rhpctl modify client -client client_name -enabled TRUE | FALSE
```

To enable a Fleet Patching and Provisioning Client, specify `-enabled TRUE`. Conversely, specify `-enabled FALSE` to disable the client. When you disable a Fleet Patching and Provisioning Client cluster, all RHPCTL commands from that client cluster will be rejected by the Fleet Patching and Provisioning Server, unless and until you re-enable the client.

> **Note:**
>
> Disabling a Fleet Patching and Provisioning Client cluster does not disable any existing working copies on the client cluster. The working copies will continue to function and any databases in those working copies will continue to run.

## Deleting a Fleet Patching and Provisioning Client

Use the following procedure to delete a Fleet Patching and Provisioning Client.

1. Before deleting the Fleet Patching and Provisioning Client, you must first delete the working copies and users on the Fleet Patching and Provisioning Server, as follows:
   a. Query the list of working copies that have been provisioned on the Fleet Patching and Provisioning Client cluster.

Run the following command:

```
$ rhpctl query workingcopy -client client_name
```

b. Delete each of the working copies listed in the output of the preceding command.

Run the following command for each working copy and specify the name of the working copy you want to delete:

```
$ rhpctl delete workingcopy -workingcopy working_copy_name
```

c. Query the list of users from the Fleet Patching and Provisioning Client cluster.

Run the following command:

```
$ rhpctl query user -client client_name
```

d. Delete the users listed in the output of the preceding command, as follows:

Run the following command and specify the name of the user you want to delete and the name of the client:

```
$ rhpctl delete user -user user_name –client client_name
```

2. On the Fleet Patching and Provisioning Client cluster, delete the client, as follows:

a. Stop the Fleet Patching and Provisioning Client daemon.

Run the following command:

```
$ srvctl stop rhpclient
```

b. Delete the Fleet Patching and Provisioning Client configuration.

Run the following command:

```
$ srvctl remove rhpclient
```

3. Delete the client site configuration on the Fleet Patching and Provisioning Server cluster.

Run the following command and specify the name of the client:

```
$ rhpctl delete client -client client_name
```

# Oracle Fleet Patching and Provisioning Server Self-upgrade

Perform these steps when upgrading Oracle Grid Infrastructure to 23ai using the Oracle Fleet Patching and Provisioning (Oracle FPP) Server self-upgrade feature.

> **Note:**
>
> You must ensure that the fix for bug #36226033 is present in your Oracle Grid Infrastructure 19c home before starting the self-upgrade.

1. Create a backup of the Oracle FPP Server metadata.

```
$ srvctl stop rhpserver
$ rhprepos export -expfile /u01/app/rhp19c_bck/19c_backup.dmp
$ srvctl start rhpserver
```

   Ensure that the directory does not contain another file with the same name, otherwise you will get an error.

2. Import an Oracle Database 23ai software image on the Oracle FPP Server.

```
$ rhpctl import image -image db23ai -zip db_zip_file_path
-imagetype ORACLEDBSOFTWARE
```

3. Import an Oracle Grid Infrastructure 23ai software image on the Oracle FPP Server.

```
$ rhpctl import image -image gi23ai -zip gi_image_zip_path
-imagetype ORACLEGISOFTWARE
```

4. Add Oracle Database 23ai working copy on the Oracle FPP Server.

```
$ rhpctl add workingcopy -workingcopy fppdbwc -image db23ai -path
path_to_db_23ai_home
-oraclebase Oracle_base_path -storagetype LOCAL -ignoreprereq
```

5. Add software only Oracle Grid Infrastructure 23ai working copy on the Oracle FPP Server.

```
$ rhpctl add workingcopy -workingcopy gi23ai_wc -path path_to_GI_23ai_home
-oraclebase Oracle_base_path -softwareonly  -image gi23ai
```

6. As the `root` user, change ownership of the `rhp.pref` with CRS user on all cluster nodes.

```
$ Grid_home/bin/fppsPreGIUpg.sh
```

7. Run the upgrade command in batches.

```
$ Grid_home/bin/rhpctl.sh upgrade gihome -sourcehome old_GI_home -destwc
GI_23ai_working_copy_name
-batches "(batch1),(batch2)" -fppdbwc fpp_db_working_copy -ignoreprereq
```

8. Run the `upgrade gihome` command with the `-continue` option.

```
$ Grid_home/bin/rhpctl.sh upgrade gihome -destwc GI_23ai_working_copy_name
-continue
```

**Related Topics**

- Oracle Grid Infrastructure Installation and Upgrade Guide for Linux
- Upgrading Oracle Grid Infrastructure
  If you are using Fleet Patching and Provisioning, then you can use a single command to upgrade an Oracle Grid Infrastructure home.

# Migrating Metadata from GIMR to Metadata Repository for Manual Upgrade

Use this procedure to import the metadata in to a new Oracle FPP metadata repository after manually upgrading to Oracle Grid Infrastructure 23ai.

Starting from Oracle Grid Infrastructure 23ai, Grid Infrastructure Management Repository (GIMR) is desupported. Oracle FPP Server requires a central metadata repository and thus you need to migrate your existing metadata from GIMR to Oracle FPP metadata repository after manually upgrading to Oracle Grid Infrastructure 23ai.

You must export the metadata from GIMR before removing the GIMR while upgrading to Oracle Grid Infrastructure 23ai.

1. Setup an Enterprise Edition of single-instance Oracle Database or Oracle RAC One Node (with own license) across all active nodes in the cluster.
2. Upgrade the Oracle FPP metadata repository by importing the metadata, which you exported during the Oracle Grid Infrastructure 23ai upgrade.

```
/u01/app/23.0.0/grid/reposScript.sh -db_home=FPP_repos_database_home -
mode="Upgrade"
-from_version="19.0.0.0.0" -
import_dir=directory_with_exported_FPP_metadata -diskgroup=META
```

# Oracle Fleet Patching and Provisioning Backup and Restore

Create a backup of Oracle FPP Server to restore in the event of a failure, or to relocate the Oracle FPP Server to a different hardware.

- About Oracle FPP Server Backup and Restore Options
  Backup the Oracle FPP Server and use the restore and relocate features to restore the Oracle FPP Server to the same or another system.
- Backing Up Oracle Fleet Patching and Provisioning Server
  Create a backup of the Oracle FPP Server to restore in the event of a failure or to relocate to a different hardware.
- Restoring Oracle Fleet Patching and Provisioning Server
  Restore Oracle FPP Server from a backup to recover data after a system failure.
- Relocating Oracle Fleet Patching and Provisioning Server
  Relocate Oracle FPP Server to a new server on a new hardware and configure Oracle FPP Clients to associate with the new Oracle FPP Server.

## About Oracle FPP Server Backup and Restore Options

Backup the Oracle FPP Server and use the restore and relocate features to restore the Oracle FPP Server to the same or another system.

Oracle FPP is used to manage large fleets of Oracle Grid Infrastructure and Oracle Database deployments on different types of hardware. To avoid data loss due to system failure, or to relocate your Oracle FPP Server to a different hardware, you can create a backup of the entire Oracle FPP Server. The following backup and restore options are available:

- A full backup of the FPP server. It represents a recovery point-in-time of a complete and working Oracle FPP Server. The full backup is an auto-contained group of files and directories from where you can restore the Oracle FPP Server in case of a system failure or to relocate to a different hardware.
- A full restore from an Oracle FPP Server backup to restore Oracle FPP Server after a disaster. You must have a full backup to perform a full restore.
- A full restore to a different cluster, which relocates the current Oracle FPP Server to a new cluster. The original Oracle FPP Server converts into an Oracle FPP Client of the new server. You must move all Oracle FPP Clients to the new Oracle FPP Server after relocation.

Oracle FPP Server supports the following storage options for backup:

- Local file system.
- Network File System (NFS) accessible by the current Oracle FPP Server.

A full Oracle FPP Server backup includes metadata, credentials, list of the registered OS users, user actions, and gold images. Backup is stored in a zip format at the location that you specify during backup.

> **Note:**
>
> You should clean up the legacy gold images or archive the old gold images using the `rhpctl archive image` command.

**Oracle FPP Backup and Restore Exclusions**

- Ongoing **move** operations are not supported for restore and metadata of these operations is cleared.
- Incomplete images and working copies at the time of the backup are excluded from the backup and are removed from the metadata.
- All pending jobs for the Oracle FPP Server with expired timer are set to `UNKNOWN` and all pending jobs with running timer are set to `ABORTED`.
- For restore, working copies from the Oracle FPP Server are considered lost, thus their metadata is cleared. You can recreate or register them after the restore operation is complete.
- Any operations done between the backup and the restore are lost, including any operations done after recreating the Oracle FPP Server when creating the new cluster, whether for disaster recovery or relocation.

**Related Topics**

- Archiving and Unarchiving Gold Images
  Starting with Oracle Grid Infrastructure 23ai, you can archive the gold images to save disk space without deleting them and unarchive the gold images if you need to use them again.

# Backing Up Oracle Fleet Patching and Provisioning Server

Create a backup of the Oracle FPP Server to restore in the event of a failure or to relocate to a different hardware.

Oracle FPP Server always creates full backup. Backup is stored in a zip format at the location that you specify during backup.

- Provide an absolute path to an empty directory with write access.
- Stop all ongoing operations on the Oracle FPP Server before attempting the backup, otherwise the backup will be rejected.
- Create a full backup of the Oracle FPP Server.

  ```
  rhpctl backup server -path backup_path
  ```

  Oracle FPP Server creates a timestamped zip file, `fppsBackup_timestamp.zip`, at the specified location. The backup includes metadata dump file, wallet file, text file with non-built-in Oracle FPP users, zip file for each gold image, and zip file of the user actions directory.

  > **Note:**
  >
  > Incomplete images, working copy snapshots, and checkpoint files are excluded from the backup.

- Do no store backup in the same cluster so that they are not lost in a system failure.
- To relocate to a different server, stop Oracle FPP Server after creating a backup. New operations performed on the server makes the backup obsolete.

# Restoring Oracle Fleet Patching and Provisioning Server

Restore Oracle FPP Server from a backup to recover data after a system failure.

Oracle FPP Server always performs a full restore.

- Provide an absolute path of a directory with write access where the backup zip file is either stored or extracted.
- Stop all ongoing operations on the Oracle FPP Server before attempting the restore, otherwise the restore will fail.
- Ensure that the Oracle Grid Infrastructure version, including the RU version, is the same as when the backup was done.
- Check the cluster name. If the cluster name is the same as when the backup was done, Oracle FPP Server restores the services, otherwise it relocates the services to the new cluster.
- Stop Oracle FPP Server before relocating Oracle FPP services to a new server.

1. Perform a full restore of the Oracle FPP Server.

   ```
   $ rhpctl restore server -backup full_backup_location
   ```

   Restore creates a timestamped directory, `fppsRestore_timestamp`, at the backup location. This directory includes extracted backup files and other related information.

> **Note:**
>
> - Retry a failed restore, instead of starting from the beginning. Provide the path to the directory of the extracted backup created by the previous run, *fppsRestore_timestamp*, instead of the path to the backup zip.
> - Resuming any operations marked as pending or ongoing in the metadata is not supported.

2. Open the `osUsers.txt` file to make sure that the listed operating system users exist in the cluster.

## Relocating Oracle Fleet Patching and Provisioning Server

Relocate Oracle FPP Server to a new server on a new hardware and configure Oracle FPP Clients to associate with the new Oracle FPP Server.

1. Copy the Oracle FPP Server backup of the old server to the new server.
2. Perform a full restore of the Oracle FPP Server.

   ```
   $ rhpctl restore server -backup full_backup_location
   ```

   After the relocation is complete, the old Oracle FPP Server is marked as an Oracle FPP Client.
3. Open the `osUsers.txt` file to make sure that the listed operating system users exist in the cluster.
4. Copy the client data file for each Oracle FPP Client, located at *backup_path*`/fppsRestore_`*timestamp*`/clientData/` from the new Oracle FPP Server to the corresponding Oracle FPP Client.
5. Update each Oracle FPP Client, except the old Oracle FPP Server, with the new client data file.

   ```
   $ srvctl stop rhpclient
   $ srvctl modify rhpclient -clientdata new_client_data_file
   $ srvctl start rhpclient
   ```

6. Perform a sanity check from the Oracle FPP Client to ensure that the client can connect to the server.

   ```
   $ rhpctl query server
   ```

7. After all Oracle FPP Client clusters are associated with the new Oracle FPP Server, then convert the original Oracle FPP Server to an Oracle FPP Client.

   ```
   $ srvctl stop rhpserver
   # srvctl remove rhpserver -force
   # srvctl add rhpclient -clientdata new_client_data_file
   $ srvctl start rhpclient
   ```

8. Perform a sanity check from the converted Oracle FPP Client to ensure that the client can connect to the server.

```
$ rhpctl query server
```

**Related Topics**

- Restoring Oracle Fleet Patching and Provisioning Server
  Restore Oracle FPP Server from a backup to recover data after a system failure.

4

# Managing Gold Images and Working Copies

You can add new gold images to your Oracle Fleet Patching and Provisioning Server, create working copies from the gold images, and provision Oracle homes.

- Adding Gold Images to the Fleet Patching and Provisioning Server
  Use RHPCTL to add gold images for later provisioning of software.
- Provisioning Copies of Gold Images
  Use RHPCTL to provision copies of gold images to Fleet Patching and Provisioning Servers, Clients, and `rhpclient`-less targets.
- Provisioning Oracle Grid Infrastructure Homes
  When you create a working copy of a gold image as part of a move or upgrade operation, Fleet Patching and Provisioning configures the operating system groups in the new working copy to match those of the source software home.
- Provisioning Oracle Database Homes
  Use the `rhpctl add workingcopy` command to provision a working copy of a database home on a Oracle Fleet Patching and Provisioning Server, Client, or `rhpclient`-less target.
- Creating an Oracle Database
  Create an Oracle Database on a working copy.
- Archiving and Unarchiving Gold Images
  Starting with Oracle Grid Infrastructure 23ai, you can archive the gold images to save disk space without deleting them and unarchive the gold images if you need to use them again.

## Adding Gold Images to the Fleet Patching and Provisioning Server

Use RHPCTL to add gold images for later provisioning of software.

The Fleet Patching and Provisioning Server stores and serves gold images of software homes. These images must be instantiated on the Fleet Patching and Provisioning Server. Images are read-only, and you cannot run programs from them. To create a usable software home from an image, you must create a working copy of a gold image. You cannot directly use images as software homes. You can, however, use images to create working copies (software homes).

> **Note:**
>
> Oracle FPP allows you to install the gold images without transferring them to the destination host. This feature is known as *zipcopy* and you can use it to provision Oracle Database homes. You can also use this feature to provision Oracle Grid Infrastructure homes that exists on the destination hosts, but not to provision new Oracle Grid Infrastructure homes.

You can import software to the Fleet Patching and Provisioning Server using any one of the following methods:

- You can import an image from an installed home on the Fleet Patching and Provisioning Server using the following command:

  ```
  rhpctl import image -image image_name -path path_to_installed_home
    [-imagetype ORACLEDBSOFTWARE | ORACLEGISOFTWARE | ORACLEGGSOFTWARE |
  SOFTWARE]
  ```

  `ORACLEDBSOFTWARE` is the default if `-imagetype` is not specified.

- You can import a new image from a zip file using the `-zip` parameter or use the zipcopy feature with the `-zip` and `-location` parameters:

  ```
  $ rhpctl import image -image image_name -zip zip_file_path -location
    location_on_destination_host_where_image_is_available
  ```

  The `-zip` parameter specifies the location from which you can import the image to the Oracle FPP server. The `-location` parameter specifies a location where the image is available on the destination host as a zip file. The `-location` parameter also instructs the Oracle FPP server to not copy the image-related files from the Oracle FPP server to the destination host.

  You can make the image zip files available on the destination hosts using either local or shared storage. For shared storage, you can use NFS file system shared with servers, clients, and `rhpclient`-less targets. For local storage, you can copy the zip file using any option that guarantees its consistency on the destination because Oracle FPP does not verify consistency of the zip file. You can use SFTP, SCP, or download the zip file using `curl` or `wget` methods from a shared location. The file must be available at the specified location.

- You can import an image from an installed home on a Fleet Patching and Provisioning Client, using the following command run from the Fleet Patching and Provisioning Client:

  ```
  rhpctl import image -image image_name -path path_to_installed_home
  ```

Use the first two commands in the preceding list to seed the image repository, and to add additional images over time. Use the third command on the Oracle Fleet Patching and Provisioning Server as part of the workflow for creating a gold image that includes patches applied to a pre-existing gold image.

The preceding three commands also create an Oracle ACFS file system in the Oracle Fleet Patching and Provisioning root directory, similar to the following:

```
/u01/rhp/images/images/RDBMS_121020617524
```

- Image State
  Am image state is a way to restrict provisioning of an image for users with specified roles.
- Image Series
  An image series is a convenient way to group different gold images into a logical sequence.
- Image Type
  When you add or import a gold image, you must specify an image type.

**Related Topics**

- Patching Oracle Database
  To patch an Oracle database, you move the database home to a new home, which includes the patches you want to implement.
- RHPCTL Command Reference
  This section describes RHPCTL command usage information, and lists and describes RHPCTL commands.

## Image State

Am image state is a way to restrict provisioning of an image for users with specified roles.

You can set the state of an image to `TESTABLE` or `RESTRICTED` so that only users with the GH_IMG_TESTABLE or GH_IMG_RESTRICT roles can provision working copies from this image. Once the image has been tested or validated, you can change the state and make the image available for general use by running the `rhpctl promote image -image` *`image_name`* `-state PUBLISHED` command. The default image state is `PUBLISHED` when you add a new gold image, but you can optionally specify a different state with the `rhpctl add image` and `rhpctl import image` commands.

## Image Series

An image series is a convenient way to group different gold images into a logical sequence.

Fleet Patching and Provisioning treats each image as an independent entity with respect to other images. No relationship is assumed between images, even if they follow some specific nomenclature. The image administrator may choose to name images in a logical manner that makes sense to the user community, but this does not create any management grouping within the Fleet Patching and Provisioning framework.

Use the `rhpctl add series` command to create an image series and associate one or more images to this series. The list of images in an image series is an ordered list. Use the `rhpctl insertimage series` and `rhpctl deleteimage series` to add and delete images in an image series. You can also change the order of images in a series using these commands.

The `insertimage` and `deleteimage` commands do not instantiate or delete actual gold images but only change the list. Also, an image can belong to more than one series (or no series at all).

## Image Type

When you add or import a gold image, you must specify an image type.

Oracle Clusterware provides the following built-in base image types:

ORACLEDBSOFTWARE
ORACLEGISOFTWARE
ORACLEGGSOFTWARE
EXAPATCHSOFTWARE
SOFTWARE

Every gold image must have an image type, and you can create your own image types. A new image type must be based on one of the built-in types. The image type directs Fleet Patching and Provisioning to apply its capabilities for managing Oracle Grid Infrastructure and Oracle

Database homes. Fleet Patching and Provisioning also uses image type to organize the custom workflow support framework.

### Creating a Custom Image Type

Use the `rhpctl add imagetype` command to create custom image types.

For example, to create an image type called DBTEST, which is based on the ORACLEDBSOFTWARE image type:

```
$ rhpctl add imagetype -imagetype DBTEST -basetype ORACLEDBSOFTWARE
```

> **Note:**
>
> When you create an image type that is based on an existing image type, the new image type does not inherit any user actions (for custom workflow support) from the base type.

# Provisioning Copies of Gold Images

Use RHPCTL to provision copies of gold images to Fleet Patching and Provisioning Servers, Clients, and `rhpclient`-less targets.

After you create and import a gold image, you can provision software by adding a copy of the gold image (called a working copy) on the Fleet Patching and Provisioning Server, on a Fleet Patching and Provisioning Client, or an `rhpclient`-less target. You can run the software provisioning command on either the Server or a Client.

> **Note:**
>
> You can add working copy as Zip files by using the `-location` parameter and make the zip files available either on a local or a shared storage at the specified location on all the Oracle FPP Clients and `rhpclient`-less targets. You must specify the `-localmount` parameter to avoid transferring the image and to decompress the zip file on the local storage.

- To create a working copy on the Fleet Patching and Provisioning Server:

  ```
  $ rhpctl add workingcopy -workingcopy working_copy_name -softwareonly -
  image image_name
  ```

- To create a working copy in a local file system on a Fleet Patching and Provisioning Client:

  ```
  $ rhpctl add workingcopy -workingcopy working_copy_name -image image_name
     -storagetype LOCAL -path path_to_software_home -softwareonly
  ```

- To create a working copy on a Fleet Patching and Provisioning Client from the Fleet Patching and Provisioning Server:

```
$ rhpctl add workingcopy -workingcopy working_copy_name -image image_name
   -client client_cluster_name -softwareonly
```

- To create a working copy on the Fleet Patching and Provisioning Server using the image file that you imported with the `-zip` option:

```
$ rhpctl add workingcopy -image image_name -workingcopy working_copy_name -
user oracle
-oraclebase Oracle_base -targetnode target_node_name -path
path_to_software_home
-sudouser opc -sudopath /bin/sudo -storagetype LOCAL -softwareonly
```

The `-localmount` option instructs the Oracle FPP server to skip the copy operation. The `-location` option specifies where the zip image is available on the destination host. By default, Oracle FPP uses the location used to import the image.

> **Note:**
>
> - The directory you specify in the `-path` parameter must be empty.
> - You can re-run the provisioning command in case of an interruption or failure due to system or user errors. After you fix the reported errors, re-run the command and it will resume from the point of failure.

- Storage Options for Provisioned Software
  Choose one of two storage options where Fleet Patching and Provisioning stores working copies of gold images.
- Provisioning for a Different User
  If you want a different user to provision software other than the user running the command, then use the `-user` parameter of the `rhpctl add workingcopy` command.
- User Group Management in Fleet Patching and Provisioning
  When you create a working copy of a gold image as part of a move or upgrade operation, Fleet Patching and Provisioning configures the operating system groups in the new working copy to match those of the source software home.

**Related Topics**

- Storage Options for Provisioned Software
  Choose one of two storage options where Fleet Patching and Provisioning stores working copies of gold images.

# Storage Options for Provisioned Software

Choose one of two storage options where Fleet Patching and Provisioning stores working copies of gold images.

When you provision software using the `rhpctl add workingcopy` command, you can choose from two storage options where Fleet Patching and Provisioning places that software:

- In an Oracle ACFS shared file system managed by Fleet Patching and Provisioning (for database homes only)
- In a local file system not managed by Fleet Patching and Provisioning

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, you can transfer working copies as zip files using SSH and store imported gold images as zip files. If the image is not in the zip format, then the Oracle FPP Server will automatically create a zip file and transfer.

Using the `rhpctl add workingcopy` command with the `-storagetype` and `-path` parameters, you can choose where you store provisioned working copies. The applicability of the parameters depends on whether the node (or nodes) to which you are provisioning the working copy is a Fleet Patching and Provisioning Server, Fleet Patching and Provisioning Client, or a non-Fleet Patching and Provisioning client. You can choose from the following values for the `-stroragetype` parameter:

- `RHP_MANAGED`: Choosing this value, which is available for Fleet Patching and Provisioning Servers and Fleet Patching and Provisioning Clients, stores working copies in an Oracle ACFS shared file system. The `-path` parameter is not used with this option because Fleet Patching and Provisioning manages the storage option.

  > **Notes:**
  >
  > - You cannot store Oracle Grid Infrastructure homes in `RHP_MANAGED` storage.
  > - Oracle recommends using the `LOCAL` storage type.
  > - If you choose to provision working copies on a Fleet Patching and Provisioning Client, and you do not specify the `-path` parameter, then the storage type defaults to `LOCAL` only if there is an Oracle ASM disk group on the client. Otherwise the command will fail. If you specify a location on the client for the `-path` parameter, then the storage type defaults to `LOCAL` with or without an Oracle ASM disk group.

- `LOCAL`: Choosing this value stores working copies in a local file system that is not managed by Fleet Patching and Provisioning.

  When adding a database working copy, specifying a path is optional. If a path is not specified, then a path under `ORACLE_BASE` is automatically chosen. This path is displayed on the terminal.

In cases where you specify the `-path` parameter, if the file system is shared among all of the nodes in the cluster, then the working copy gets created on this shared storage. If the file system is not shared, then the working copy gets created in the location of the given path on every node in the cluster.

> **Note:**
>
> The directory you specify in the `-path` parameter must be empty.

**Related Topics**

- rhpctl add workingcopy
  Creates a working copy on a client cluster.

## Provisioning for a Different User

If you want a different user to provision software other than the user running the command, then use the `-user` parameter of the `rhpctl add workingcopy` command.

> **Note:**
>
> The default user is the user as which the RHPCTL command is being run.

When the provisioning is completed, all files and directories of the provisioned software are owned by the user you specified. Permissions on files on the remotely provisioned software are the same as the permissions that existed on the gold image from where you provisioned the application software.

## User Group Management in Fleet Patching and Provisioning

When you create a working copy of a gold image as part of a move or upgrade operation, Fleet Patching and Provisioning configures the operating system groups in the new working copy to match those of the source software home.

These operating system groups are used for operating system authentication, such as OSDBA and OSOPER. Oracle FPP configures operating system groups for both unmanaged and managed software homes from which you move or upgrade.

The `rhpctl move database`, `rhpctl move gihome`, `rhpctl upgrade database`, and `rhpctl upgrade gihome` commands all require you to specify a source home (either an unmanaged home or a managed home (working copy) that you provisioned using Fleet Patching and Provisioning), and a destination home (which must be a working copy).

When you have provisioned the destination home using the `rhpctl add workingcopy` command, prior to performing a move or upgrade operation, you must ensure that the groups configured in the source home match those in the destination home. Fleet Patching and Provisioning configures the groups as part of the add operation.

When you create a gold image of either the ORACLEGISOFTWARE or the ORACLEDBSOFTWARE image type from a source software home (using the `rhpctl import image` command) or from a working copy (using the `rhpctl add image` command), the gold image inherits the Oracle user groups that were configured in the source. You cannot override this feature.

You can define user groups for ORACLEGISOFTWARE and ORACLEDBSOFTWARE working copies using the `rhpctl add workingcopy` command, depending on the image type and user group, as discussed in the subsequent sections.

This section describes how Fleet Patching and Provisioning manages user group configuration, and how the `-groups` command-line option of `rhpctl add workingcopy` functions.

### ORACLEGISOFTWARE (Oracle Grid Infrastructure)

When you provision an Oracle Grid Infrastructure working copy of a gold image, the groups are set in the working copy according to the type of provisioning (whether configuring a new server or performing a software only setup), and whether you specify the `-groups` parameter with `rhpctl add workingcopy`. You can define OSDBA and OSASM user groups in Oracle Grid Infrastructure software with either the `-softwareonly` command parameter or by using a response file with the `rhpctl add workingcopy` command.

If you are provisioning only the Oracle Grid Infrastructure software using the `-softwareonly` command parameter, then you cannot use the `-groups` parameter, and Fleet Patching and Provisioning obtains OSDBA and OSASM user group information from the active Grid home.

If none of the preceding applies, then Fleet Patching and Provisioning uses the installer default user group.

If you are provisioning and configuring a working copy using information from a response file, then Fleet Patching and Provisioning:

1. Uses the value of the user group from the command line, if provided, for OSDBA or OSASM, or both.
2. If you provide no value on the command line, then Fleet Patching and Provisioning retrieves the user group information defined in the response file.

If you are defining the OSOPER Oracle group, then, again, you can either use the `-softwareonly` command parameter or use a response file with the `rhpctl add workingcopy` command.

If you use the `-softwareonly` command parameter, then you can provide the value on the command line (using the `-groups` parameter) or leave the user group undefined.

If you are provisioning and configuring a working copy of a gold image using information from a response file, then you can provide the value on the command line, use the information contained in the response file, or leave the OSOPER Oracle group undefined.

### ORACLEDBSOFTWARE (Oracle Database)

If you are provisioning a working copy of Oracle Database software and you want to define Oracle groups, then use the `-groups` command parameter with the `rhpctl add workingcopy` command. Oracle groups available in the various Oracle Database releases are as follows:

- Oracle Database 11*g* release 2 (11.2)

  OSDBA
  OSOPER

- Oracle Database 12*c* release 1 (12.1)

  OSDBA
  OSOPER
  OSBACKUP
  OSDG
  OSKM

- Oracle Database 12*c* release 2 (12.2) and later

  OSDBA
  OSOPER
  OSBACKUP

OSDG
OSKM
OSRAC

Regardless of which of the preceding groups you are defining (except for OSOPER), Fleet Patching and Provisioning takes the values of the groups from the command line (using the `-groups` parameter) or uses the default values, which Fleet Patching and Provisioning obtains from the `osdbagrp` binary of the gold image.

If any group picked up from the `osdbagrp` binary is not in the list of groups to which the database user belongs (listed by the `id` operating system command), then Fleet Patching and Provisioning uses the installer default user group. Otherwise, the database user is the user running the `rhpctl add workingcopy` command.

# Provisioning Oracle Grid Infrastructure Homes

When you create a working copy of a gold image as part of a move or upgrade operation, Fleet Patching and Provisioning configures the operating system groups in the new working copy to match those of the source software home.

Oracle Grid Infrastructure homes are distributed in the form of working copies of gold images. After a working copy has been provisioned, Oracle Fleet Patching and Provisioning can optionally configure Oracle Grid Infrastructure. This gives Oracle Fleet Patching and Provisioning the ability to create an Oracle Grid Infrastructure installation on a group of one or more nodes that initially do not have Oracle Grid Infrastructure installed.

Oracle Fleet Patching and Provisioning also has commands for managing Oracle Grid Infrastructure homes, such as switching to a patched home or upgrading to a new Oracle Grid Infrastructure version. These are both single commands that orchestrate the numerous steps involved. You can also revert to the original home. Also, Oracle Fleet Patching and Provisioning can add or delete nodes from an Oracle Grid Infrastructure configuration.

- About Deploying Oracle Grid Infrastructure Using Oracle Fleet Patching and Provisioning
  You can use Oracle Fleet Patching and Provisioning to provision and maintain your Oracle Grid Infrastructure homes.
- Provisioning Oracle Grid Infrastructure Software
  Fleet Patching and Provisioning has several methods to provision and, optionally, configure Oracle Grid Infrastructure and Oracle Restart homes.

## About Deploying Oracle Grid Infrastructure Using Oracle Fleet Patching and Provisioning

You can use Oracle Fleet Patching and Provisioning to provision and maintain your Oracle Grid Infrastructure homes.

Oracle Fleet Patching and Provisioning enables mass deployment and maintenance of standard operating environments for databases, clusters, and user-defined software types.

Oracle FPP enables you to install clusters, and provision, patch, scale, and upgrade Oracle Grid Infrastructure, Oracle Restart, and Oracle Database homes. The supported releases are 19c, 21c, and 23ai. You can also provision applications and middleware using Oracle Fleet Patching and Provisioning.

Oracle Fleet Patching and Provisioning is a service in Oracle Grid Infrastructure that you can use in either of the following modes:

- Central Oracle Fleet Patching and Provisioning Server

  The Oracle Fleet Patching and Provisioning Server stores and manages standardized images, called gold images. Gold images can be deployed to any number of nodes across the data center. You can create new clusters and databases on the deployed homes and can use them to patch, upgrade, and scale existing installations.

  The Oracle Fleet Patching and Provisioning Server can manage the following types of installations:

  – Software homes on the cluster hosting the Oracle Fleet Patching and Provisioning Server itself.
  – Installations running Oracle Grid Infrastructure 19c and later releases.
  – Oracle Fleet Patching and Provisioning Clients running Oracle Grid Infrastructure 19c and later releases.
  – Installations running without Oracle Grid Infrastructure.

  The Oracle Fleet Patching and Provisioning Server can provision new installations, and manage existing installations, without requiring any changes to the existing installations. The Oracle Fleet Patching and Provisioning Server can automatically share gold images among peer servers to support enterprises with geographically distributed data centers.

- Oracle Fleet Patching and Provisioning Client

  The Oracle Fleet Patching and Provisioning Client can be managed from the Oracle Fleet Patching and Provisioning Server, or directly by running commands on the client itself. The Oracle Fleet Patching and Provisioning Client is a service built into the Oracle Grid Infrastructure and is available in Oracle Grid Infrastructure 19c and later releases. The Oracle Fleet Patching and Provisioning Client can retrieve gold images from the Oracle Fleet Patching and Provisioning Server, upload new images based on the policy, and apply maintenance operations to itself.

**Related Topics**

- Creating an Oracle Grid Infrastructure 23ai Deployment
  Provision Oracle Grid Infrastructure software on two nodes that do not currently have a Grid home, and then configure Oracle Grid Infrastructure to form a multi-node Oracle Grid Infrastructure installation.

# Provisioning Oracle Grid Infrastructure Software

Fleet Patching and Provisioning has several methods to provision and, optionally, configure Oracle Grid Infrastructure and Oracle Restart homes.

Fleet Patching and Provisioning can provision and configure Oracle Grid Infrastructure on one or more nodes that do not currently have a Grid home, and then configure Oracle Grid Infrastructure to form a single-node or multi-node Oracle Grid Infrastructure installation.

Use the `rhpctl add workingcopy` command to install and configure Oracle Grid Infrastructure, and to enable repeatable creation of standardized deployments.

The Fleet Patching and Provisioning Server can also provision an Oracle Grid Infrastructure home to a node or cluster that is currently running Oracle Grid Infrastructure. The currently running Grid home can be a home that Fleet Patching and Provisioning did not provision (an *unmanaged* home) or a home that Fleet Patching and Provisioning did provision (a *managed* home).

You can also provision an Oracle Restart to a node in the cluster.

In either case, use the `-softwareonly` parameter of the `rhpctl add workingcopy` command. This provisions but does not activate the new Grid home, so that when you are ready to switch to that new home, you can do so with a single command.

- To inform Fleet Patching and Provisioning the nodes on which to install Oracle Grid Infrastructure, and to configure Oracle Grid Infrastructure, you provide directions in a response file, as in the following example:

  ```
  $ rhpctl add workingcopy -workingcopy GI_HOME_11204_WCPY -image
  GI_HOME_11204 -responsefile /u01/app/rhpinfo/GI_11204_install.rsp
    {authentication_option}
  ```

  The preceding command provisions the `GI_HOME_11204_WCPY` working copy based on the `GI_HOME_11204` gold image to a destination server specified in the `GI_11204_install.rsp` response file. In addition to identifying the destination nodes, the response file specifies information about the Oracle Grid Infrastructure configuration, such as Oracle ASM and GNS parameters.

  > **Note:**
  >
  > The `oracle.install.crs.rootconfig.executeRootScript=xxx` response file parameter is overridden and always set to `false` for Fleet Patching and Provisioning, regardless of what you specify in the response file.

- To provision an Oracle Grid Infrastructure home to a node or cluster that is currently running Oracle Grid Infrastructure:

  ```
  $ rhpctl add workingcopy -workingcopy GI_HOME_12201_PATCHED_WCPY -image
  GI_HOME_12201_PSU1 -client CLUST_002 -softwareonly
  ```

  The preceding command provisions a new working copy based on the `GI_HOME_12201_PSU1` gold image to the Fleet Patching and Provisioning Client (that is running Oracle Grid Infrastructure 12c release 2 (12.2)) named `CLUST_002`. When you provision to an `rhpclient`-less target that is not running Oracle Grid Infrastructure 12c release 2 (12.2) (such as, a node running Oracle Grid Infrastructure 12c release 1 (12.1) or Oracle Grid Infrastructure 11g release 2 (11.2)), use the `-targetnode` parameter instead of `-client`.

- Specify an `rhpclient`-less target node on which you want to provision an Oracle Restart, as follows:

  ```
  $ rhpctl add workingcopy -workingcopy SIHA_GI -image goldimage -targetnode
  remote_node_name -responsefile Oracle_Restart_response_file
  {authentication_option}
  ```

**Related Topics**

- Authentication Options for Oracle Fleet Patching and Provisioning Operations
  Some RHPCTL commands show authentication choices as an optional parameter.

# Provisioning Oracle Database Homes

Use the `rhpctl add workingcopy` command to provision a working copy of a database home on a Oracle Fleet Patching and Provisioning Server, Client, or `rhpclient`-less target.

- Run the `rhpctl add workingcopy` command on a Fleet Patching and Provisioning Server, similar to the following example:

```
$ rhpctl add workingcopy -image db12c -path /u01/app/dbusr/product/12.2.0/
db12201
  -client client_007 -oraclebase /u01/app/dbusr/ -workingcopy wc_db122_1
```

  The preceding command example creates a working copy named `wc_db122_1` on all nodes of the Fleet Patching and Provisioning Client cluster named `client_007`. The gold image `db12c` is the source of the workingcopy. The directory path locations that you specify in the command must be empty.

**Related Topics**

- rhpctl add workingcopy

# Creating an Oracle Database

Create an Oracle Database on a working copy.

The Oracle Fleet Patching and Provisioning Server can add a database on a working copy that is on the Oracle Fleet Patching and Provisioning Server, itself, an Oracle Fleet Patching and Provisioning Client, or an `rhpclient`-less target. An Oracle Fleet Patching and Provisioning Client can create a database on a working copy that is running on the Oracle Fleet Patching and Provisioning Client, itself.

- After you create a working copy of a gold image and provision that working copy to an `rhpclient`-less target, you can create an Oracle Database on the working copy using the `rhpctl add database` command, similar to the following command example, which creates an Oracle Real Application Clusters (Oracle RAC) database called `db12201` on a working copy called `wc_db122_1`:

```
$ rhpctl add database -workingcopy wc_db122_1 -dbname db12201 -node
client_007_node1,client_007_node2 -dbtype RAC -datafileDestination
DATA007_DG
```

The preceding example creates an administrator-managed Oracle RAC database on two nodes in a client cluster. The data file destination is an Oracle ASM disk group that was created prior to running the command. Additionally, you can create Oracle RAC One Node and non-cluster databases.

> **Note:**
>
> When you create a database using Oracle Fleet Patching and Provisioning, the feature uses random passwords for both the SYS and SYSTEM schemas in the database and you cannot retrieve these passwords. A user with the DBA or operator role must connect to the database, locally, on the node where it is running and reset the passwords to these two accounts.

# Archiving and Unarchiving Gold Images

Starting with Oracle Grid Infrastructure 23ai, you can archive the gold images to save disk space without deleting them and unarchive the gold images if you need to use them again.

Patching Oracle Grid Infrastructure and Oracle Database requires creating new gold images, which take up a lot of disk space on the Oracle Fleet Patching and Provisioning (Oracle FPP) Server. You may not want to delete these gold images to free up disk space because you may need to refer them back. The archive feature enables you to save disk space, and enables you to unarchive if you need to use it again.

You can archive and store gold images in zip format only if there are no working copies of these gold images. Furthermore, you can not use archived image to create a working copy. When a gold image is archived, its state is updated to `ARCHIVED`.

- To archive a gold image on the Fleet Patching and Provisioning Server:

  ```
  $ rhpctl archive image -image image_name
  ```

- To unarchive a gold image on the Fleet Patching and Provisioning Server:

  ```
  $ rhpctl unarchive image -image image_name
  ```

5

# Patching and Upgrading Oracle Grid Infrastructure

The Oracle Fleet Patching and Provisioning Server provides an efficient and secure platform for patching and upgrading Oracle Grid Infrastructure.

- Patching Oracle Grid Infrastructure
  Fleet Patching and Provisioning provides two methods to patch Oracle Grid Infrastructure software homes: rolling (in batches) and non-rolling.
- Upgrading Oracle Grid Infrastructure
  If you are using Fleet Patching and Provisioning, then you can use a single command to upgrade an Oracle Grid Infrastructure home.
- Oracle Restart Patching and Upgrading
  You can use Oracle Fleet Patching and Provisioning to patch and upgrade Oracle Restart using gold images.

## Patching Oracle Grid Infrastructure

Fleet Patching and Provisioning provides two methods to patch Oracle Grid Infrastructure software homes: rolling (in batches) and non-rolling.

Patching Oracle Grid Infrastructure software involves moving the Grid home to a patched version of the current Grid home. When the patching operation is initiated by a Fleet Patching and Provisioning Server or Client, the patched version must be a working copy of a gold image. The working copy to which you are moving the Grid home can be at a lower patch level than the current home. This facilitates rollback if any problems occur after moving to the higher-level patched home.

When Oracle FPP performs out-of-place patching, it evaluates the list of non-rolling patches on the source and the destination homes. Oracle FPP compares the two lists, and if the target home has additional non-rolling patches that are not in the source home, the Oracle FPP reports that the patching cannot be performed in a rolling mode. You can then take one of the following actions:

1. Re-provision the target home so that it does not include those additional non-rolling patches and proceed with patching in rolling mode.
2. Continue with the rolling patching operation with the currently provisioned home using the `-forcerolling` option.

You can also patch Oracle Grid Infrastructure using Fleet Patching and Provisioning Local Mode. In this case, the source and destination homes are not working copies of gold images, but are two installed homes that you deployed with some method other than using Fleet Patching and Provisioning.

For information about patching using batches, refer to Patching Oracle Grid Infrastructure and Oracle Database Using Batches.

- Patching Oracle Grid Infrastructure Using the Rolling Method
  The rolling method for patching Oracle Grid Infrastructure is the default method.

- Patching Oracle Grid Infrastructure Using the Non-Rolling Method
  You can use the `-nonrolling` parameter with the `rhpctl move gihome` command, which restarts the Oracle Grid Infrastructure stack on all nodes in parallel.
- Combined Oracle Grid Infrastructure and Oracle Database Patching
  When you patch an Oracle Grid Infrastructure deployment, Oracle FPP enables you to simultaneously patch the Oracle Database homes on the cluster, so you can patch both types of software homes in a single maintenance operation.
- Zero-Downtime Oracle Grid Infrastructure Patching
  Use Fleet Patching and Provisioning to patch Oracle Grid Infrastructure without bringing down Oracle RAC database instances.

**Related Topics**

- rhpctl add workingcopy
- rhpctl move gihome

# Patching Oracle Grid Infrastructure Using the Rolling Method

The rolling method for patching Oracle Grid Infrastructure is the default method.

You use the `rhpctl move gihome` command (an atomic operation), which returns after the Oracle Grid Infrastructure stack on each node has been restarted on the new home. Nodes are restarted sequentially, so that only one node at a time will be offline, while all other nodes in the cluster remain online.

- Move the Oracle Grid Infrastructure home to a working copy of the same release level, as follows:

  ```
  $ rhpctl move gihome –sourcewc Grid_home_1 –destwc Grid_home_2
  ```

  The preceding command moves the running Oracle Grid Infrastructure home from the current managed home (the `sourcewc`) to the patched home (`destwc`) on the specific client cluster. The patched home must be provisioned on the client.
- If the move operation fails at some point before completing, then you can rerun the operation by running the command again and the operation will resume where it left off. This enables you to fix whatever problem caused the failure and resume processing from the point of failure. Or you can undo the partially completed operation and return the configuration to its initial state, as follows:

  ```
  $ rhpctl move gihome -destwc destination_workingcopy_name -revert | -
  forcecomplete [authentication_option]
  ```

  You cannot use the `-revert` parameter with an un-managed home.

**Notes:**

- You cannot move the Grid home to a home that Fleet Patching and Provisioning does not manage. Therefore, rollback (to the original home) applies only to moves between two working copies. This restriction does not apply when using the independent automaton since it operates on unmanaged homes only.
- You can delete the source working copy at any time after moving a Grid home. Once you delete the working copy, however, you cannot perform a rollback. Also, use the `rhpctl delete workingcopy` command (as opposed to `rm`, for example) to remove the source working copy to keep the Fleet Patching and Provisioning inventory correct.
- If you use the `-abort` parameter to terminate the patching operation, then Fleet Patching and Provisioning does not clean up or undo any of the patching steps. The cluster, databases, or both may be in an inconsistent state because all nodes are not patched.
- Use the `-forcecomplete` parameter to mark the move operation as complete after completing it manually.

## Patching Oracle Grid Infrastructure Using the Non-Rolling Method

You can use the `-nonrolling` parameter with the `rhpctl move gihome` command, which restarts the Oracle Grid Infrastructure stack on all nodes in parallel.

As with the rolling method, this is an atomic command which returns after all nodes are online.

**Note:**

The non-rolling patching method is available only if there is just one cluster node. If there are multiple nodes, then there is an error suggesting to use the rolling patching method.

- Use the following command to patch Oracle Grid Infrastructure in an non-rolling fashion:

```
$ rhpctl move gihome -sourcewc Grid_home_1 -destwc Grid_home_2 -nonrolling
```

## Combined Oracle Grid Infrastructure and Oracle Database Patching

When you patch an Oracle Grid Infrastructure deployment, Oracle FPP enables you to simultaneously patch the Oracle Database homes on the cluster, so you can patch both types of software homes in a single maintenance operation.

**Note:**

You cannot patch both Oracle Grid Infrastructure and Oracle Database in combination with Oracle Fleet Patching and Provisioning (Oracle FPP) Local Mode.

The following optional parameters of the `rhpctl move gihome` command are relevant to the combined Oracle Grid Infrastructure and Oracle Database patching use case:

- `-auto`: Automatically patch databases along with patching Oracle Grid Infrastructure
- `-dbhomes` *`mapping_of_Oracle_homes`*: Mapping of source and destination working copies in the following format:

  `sourcewc1=destwc1,...,`*`source_oracle_home_path`*`=destwc`*`N`*

- `-dblist` *`db_name_list`*: Patch only the specified databases
- `-excludedblist` *`db_name_list`*: Patch all databases except the specified databases
- `-nodatapatch`: Indicates that `datapatch` is not be run for databases being moved

As an example, assume that an Oracle FPP Server with Oracle Grid Infrastructure 19c (19.20) has provisioned the following working copies on an Oracle Grid Infrastructure 19c (19.19) `rhpclient`-less target cluster which includes the node `test_749`:

- `WC_1919_fppc03`: The active Grid home on the Oracle Grid Infrastructure 19c (19.19) cluster
- `WC_GI_1920_fppc03`: A software-only Grid home on the Oracle Grid Infrastructure 19c (19.20) cluster
- `WC_DB_1919_fppc03`: An Oracle RAC 19c (19.19) database home running database instances
- `WC_DB_1920_fppc03`: An Oracle RAC 19c (19.20) database home with no database instances (this is the patched home)
- `WC_DB_1920_fppc03`: An Oracle RAC 19c (19.20) database home running database instances
- `WC_DB_1920_220719_fppc03`: An Oracle RAC 19c (19.20) database home with no database instances (this is the patched home)

Further assume that you want to simultaneously move

- Oracle Grid Infrastructure from working copy `WC_1919_fppc03` to working copy `WC_GI_1920_fppc03`
- Oracle RAC Database `db1` from working copy `WC_DB_1919_fppc03` to working copy `WC_DB_1920_fppc03`
- Oracle RAC Database `db2` in working copy `WC_DB_1920_fppc03` to working copy `WC_DB_1920_220719_fppc03`

The following single command accomplishes the moves:

```
$ rhpctl move gihome -destwc WC_GI_1920_fppc03 -sourcewc WC_1919_fppc03
-auto -dbhomes
WC_DB_1919_fppc03=WC_DB_1920_fppc03,WC_DB_1920_fppc03=WC_DB_1920_220719_fppc03
```

**Notes:**

- If you have an existing Oracle home that is not currently a working copy, then specify the Oracle home path instead of the working copy name for the source home. In the preceding example, if the Oracle home path for an existing 19.19 home is `/u01/app/prod/19.19.0/dbhome1`, then replace `WC_DB_1919_fppc03=WC_DB_1920_fppc03` with `/u01/app/prod/19.20.0/dbhome1=WC_DB_1920_fppc03`.
- If the move operation fails at some point before completing, then you can either resolve the cause of the failure and resume the operation by rerunning the command, or you can undo the partially completed operation by issuing the following command, which restores the configuration to its initial state:

  ```
  $ rhpctl move gihome -destwc WC_GI_1920_fppc03 -revert
  {authentication_option}
  ```

In the preceding command example, the Oracle Grid Infrastructure 19c Grid home moves from working copy `WC_1919_fppc03` to working copy `WC_GI_1920_fppc03`, databases running on working copy `WC_DB_1919_fppc03` move to working copy `WC_DB_1920_fppc03`, and databases running on working copy `WC_DB_1920_fppc03` move to working copy `WC_DB_1920_220719_fppc03`.

For each node in the client cluster, RHPCTL:

1. Runs any configured pre-operation user actions for moving the Oracle Grid Infrastructure (`move gihome`).
2. Runs any configured pre-operation user actions for moving the database working copies (`move database`).
3. Stops services running on the node, and applies drain and disconnect options.
4. Performs the relevant patching operations for Oracle Clusterware and Oracle Database.
5. Runs any configured post-operation user actions for moving the database working copies (`move database`).
6. Runs any configured post-operation user actions for moving the Oracle Grid Infrastructure working copy (`move gihome`).

**Related Topics**

- rhpctl move gihome
  Moves the Oracle Grid Infrastructure software stack from one home to another.

# Zero-Downtime Oracle Grid Infrastructure Patching

Use Fleet Patching and Provisioning to patch Oracle Grid Infrastructure without bringing down Oracle RAC database instances.

> **Note:**
>
> The Zero Downtime Upgrade (ZDU) feature of Oracle Fleet Patching and Provisioning (FPP) is deprecated in Oracle Database 23ai.
> ZDU uses either Transient Logical Standby or Oracle GoldenGate classic. However, Transient Logical Standby does not support the latest data types, and the Oracle GoldenGate classic implementation is deprecated. Instead of ZDU, Oracle recommends that you use FPP in combination with the `DBMS_ROLLING` PL/SQL package: You can use FPP for the provisioning and deployment of new Oracle homes, and you can use `DBMS_ROLLING` as a streamlined method of performing rolling upgrades.
>
> The `DBMS_ROLLING` package is closely integrated with Application Continuity to help to minimize disruptions in the database tier, and mask rolling patching activities.

Current methods of patching the Oracle Grid Infrastructure require that you bring down all Oracle RAC database instances on the node where you are patching the Oracle Grid Infrastructure home. This issue is addressed in the Grid Infrastructure layer where by the database instances can continue to run during the Oracle Grid Infrastructure patching.

> **Note:**
>
> You can use zero-downtime patching only for out-of-place patching of Oracle Grid Infrastructure 19c Release Update (RU) 19.8 or later releases with Oracle RAC or Oracle RAC One Node databases of 19c or later releases. If your Oracle RAC or Oracle RAC One Node database release is older than 19c, then the database instances stop during zero-downtime patching.

To enable zero-downtime Oracle Grid Infrastructure patching, use the `rhpctl move gihome` command in a manner similar to the following:

```
rhpctl move gihome -tgip -sourcewc source_workingcopy_name -destwc
destination_workingcopy_name
```

### Patching System Software Binaries

When using Zero Downtime Patching, only the binaries in the Oracle Grid Infrastructure user space are patched. Additional Oracle Grid Infrastructure OS system software, kernel modules and system commands including `ACFS`, `AFD`, and `OKA`, are not updated. These commands continue to run the version previous to the patch version. After patching, the OPatch inventory displays the new patch number in the inventory; however, the running OS system software does not contain these changes. Only the OS system software that is available in the Grid Infrastructure home has been patched.

To determine the OS system software that is available in the Grid Infrastructure home, you can run the `crsctl query driver activeversion -all` command. To determine what OS system software is running on the system, use `crsctl query driver softwareversion -all`.

To update the Grid Infrastructure OS system software on a single node, you must completely stop the Grid Infrastructure software. To stop the Grid Infrastructure software, you must stop the Oracle RAC databases on the single node. After stopping the Oracle RAC databases, run `root.sh -updateosfiles` to update all the Grid Infrastructure OS system software on the single node.

Zero-downtime patching does not automatically update the operating system drivers. During the zero-downtime patching process, updated operating system drivers are copied in the Grid home, but these drivers are not installed into the operating system. The cluster continues to use the older version of the operating system drivers until the updated drivers are installed in the operating system.

If you are using Oracle ASM Filter Driver (Oracle ASMFD) or Oracle ACFS for database storage, then operating system drivers are updated in either of the following scenarios:

1. You update your operating system kernel and restart the cluster node.
2. You run the `rootcrs.sh -updateosfiles` command on each cluster node and restart the cluster nodes, if the operating system drivers fail to install.

You can check the active operating system driver version on a cluster node using the `crsctl query driver activeversion [-all]` command and available operating system driver version using the `crsctl query driver softwareversion [-all] [-f]` command.

# Upgrading Oracle Grid Infrastructure

If you are using Fleet Patching and Provisioning, then you can use a single command to upgrade an Oracle Grid Infrastructure home.

Fleet Patching and Provisioning supports upgrades to Oracle Grid Infrastructure 23ai from 21c and 19c. The destination for the upgrade can be a working copy of a gold image already provisioned or you can choose to create the working copy as part of this operation. Refer My Oracle Support note 551141.1 for more information about Grid Infrastructure and Oracle Database upgrade paths.

As an example, assume that a cluster is running Oracle Grid Infrastructure on an Oracle Grid Infrastructure home that was provisioned by Fleet Patching and Provisioning. This Oracle Grid Infrastructure home is 21c and the working copy is named accordingly, for example `GIOH21C`.

After provisioning a working copy version of Oracle Grid Infrastructure 23ai (named `GIOH23ai` in this example), you can upgrade to that working copy with this single command:

```
$ rhpctl upgrade gihome -sourcewc GIOH21C -destwc GIOH23ai
```

Fleet Patching and Provisioning is able to identify the cluster to upgrade based on the name of the source working copy. If the cluster was running on an unmanaged Oracle Grid Infrastructure home, then you would specify the path of the source home rather than providing a source working copy name, and you must also specify the cluster name.

> **Note:**
>
> You can delete the source working copy at any time after completing an upgrade. Once you delete the working copy, however, you cannot perform a rollback. Also, use the `rhpctl delete workingcopy` command (as opposed to `rm`, for example) to remove the source working copy to keep the Fleet Patching and Provisioning inventory correct.

**Related Topics**

- My Oracle Support Note 551141.1

# Oracle Restart Patching and Upgrading

You can use Oracle Fleet Patching and Provisioning to patch and upgrade Oracle Restart using gold images.

You can move the `rhpclient`-less target of single-node Oracle Restart to an Oracle home that you provision from a gold image that includes any patches. Oracle Fleet Patching and Provisioning ensures the copying of the configuration files, such as listener.ora, to the new Oracle home.

You can also use Oracle Fleet Patching and Provisioning to upgrade Oracle Restart using gold images. Upgrade the Oracle Restart environment by upgrading the Oracle home on the destination Oracle home that you provision from a higher-level gold image. Oracle Fleet Patching and Provisioning updates the configuration and inventory settings.

Use the RHPCTL utility similar to the following to patch Oracle Restart:

```
rhpctl move gihome -sourcewc Oracle_Restart_home_1 -destwc
Oracle_Restart_home_2
-targetnode Oracle_Restart_node {superuser credentials}
```

Use the RHPCTL utility similar to the following to upgrade Oracle Restart:

```
rhpctl upgrade gihome -sourcewc source_Oracle_Restart_home -destwc
higher_version_Oracle_Restart_home
-targetnode Oracle_Restart_node {superuser credentials}
```

**Related Topics**

- rhpctl move gihome
  Moves the Oracle Grid Infrastructure software stack from one home to another.
- rhpctl upgrade gihome
  Upgrades the Oracle Grid Infrastructure from a source working copy or source home path to a destination working copy.

6

# Patching and Upgrading Oracle Database

The Oracle Fleet Patching and Provisioning Server provides an efficient and secure platform for patching and upgrading Oracle Database.

- Patching Oracle Database
  To patch an Oracle database, you move the database home to a new home, which includes the patches you want to implement.
- Patching Oracle Data Guard Database
  Starting with Oracle Grid Infrastructure 23ai, Oracle FPP enables you to automatically patch Oracle Data Guard standby databases before patching their primary databases.
- Upgrading Oracle Database
  Fleet Patching and Provisioning provides two options for upgrading Oracle Database. Both options are performed with a single command.
- Zero-Downtime Upgrade
  Using Oracle Fleet Patching and Provisioning, which automates and orchestrates database upgrades, you can upgrade an Oracle RAC or Oracle RAC One Node database with no disruption in service.

## Patching Oracle Database

To patch an Oracle database, you move the database home to a new home, which includes the patches you want to implement.

Use the `rhpctl move database` command to move one or more database homes to a working copy of the same database release level. The databases may be running on a working copy, or on an Oracle Database home that is not managed by Fleet Patching and Provisioning.

When the move operation is initiated by a Fleet Patching and Provisioning Server or Client, the version moved to must be a working copy of a gold image. You can also perform this operation using the independent automaton in an environment where no Fleet Patching and Provisioning Server is present. In this case, the source and destination homes are not working copies of gold images, but are two installed homes that you deployed with some method other than using Fleet Patching and Provisioning.

The working copy to which you are moving the database can be at a lower patch level than the current database home. This facilitates rollback in the event that you encounter any problems after moving to the higher level patched home.

The working copy to which you are moving the database home can be at the same patch level as the original working copy. This is useful if you are moving a database home from one storage location to another, or if you wish to convert an unmanaged home to a managed home while staying at the same patch level.

Fleet Patching and Provisioning applies all patches out-of-place, minimizing the downtime necessary for maintenance. Fleet Patching and Provisioning also preserves the current configuration, enabling the rollback capability previously described. By default, Fleet Patching and Provisioning applies patches in a rolling manner, which reduces, and in many cases eliminates, service downtime. Use the `-nonrolling` option to perform patching in non-rolling

mode. The database is then completely stopped on the old `ORACLE_HOME`, and then restarted to make it run from the newly patched `ORACLE_HOME`.

When Oracle FPP performs out-of-place patching, it evaluates the list of non-rolling patches on the source and the destination homes. Oracle FPP compares the two lists, and if the target home has additional non-rolling patches that are not in the source home, the Oracle FPP reports that the patching cannot be performed in a rolling mode. You can then take one of the following actions:

1. Re-provision the target home so that it does not include those additional non-rolling patches and proceed with patching in rolling mode.
2. Continue with the rolling patching operation with the currently provisioned home using the `-forcerolling` option.

For information about patching using batches, refer to Patching Oracle Grid Infrastructure and Oracle Database Using Batches.

> **Note:**
>
> Part of the patching process includes applying Datapatch. When you move an Oracle Database , Oracle FPP completes this step for you. Fleet Patching and Provisioning is Oracle Data Guard-aware, and will not apply Datapatch to Oracle Data Guard standbys.

### Workflow for Database Patching

Assume that a database named `myorcldb` is running on a working copy that was created from an Oracle Database 12c release 2 (12.2) gold image named `DB122`. The typical workflow for patching an Oracle Database home is as follows:

1. Create a working copy of the Oracle Database that you want to patch, in this case `DB122`.
2. Apply the patch to the working copy you created.
3. Test and validate the patched working copy.
4. Use the `rhpctl add image` command to create a gold image (for example, `DB122_PATCH`) from the patched working copy.

   > **Note:**
   >
   > The working copy you specify in the preceding command must be hosted on the Fleet Patching and Provisioning Server in Fleet Patching and Provisioning-managed storage.

5. Delete the patched working copy with the patched Oracle Database using the `rhpctl delete workingcopy` command.

   > **Note:**
   >
   > Do not remove directly using the `rm` command or some other method, because this does not update the Fleet Patching and Provisioning inventory information.

6. Create a working copy from the patched gold image, (`DB122_PATCH`).

7. Move `myorcldb` to the working copy you recently created.
8. When you are confident that you will not need to roll back to the working copy on which the database was running at the beginning of the procedure, delete that working copy using the `rhpctl delete workingcopy` command.

**Related Topics**

- Patching Oracle Grid Infrastructure
- Provisioning Copies of Gold Images
- Adding Gold Images to the Fleet Patching and Provisioning Server
- RHPCTL Command Reference

# Patching Oracle Data Guard Database

Starting with Oracle Grid Infrastructure 23ai, Oracle FPP enables you to automatically patch Oracle Data Guard standby databases before patching their primary databases.

Oracle Fleet Patching and Provisioning (Oracle FPP) automatically discovers the standby databases, associated with primary databases, and patches the standby databases before patching the primary database using a single command.

> **Note:**
>
> If the Oracle Database Guard databases are registered on a different client, which can be in different data center, then those servers must be register peers of each other.

1. Check the Oracle Data Guard configuration of a database to identify the primary database and its standby databases.

   ```
   rhpctl query workingcopy -workingcopy workingcopy_name -details
   ```

2. Schedule a job for each standby database associated with the primary database using a single job ID.

   ```
   rhpctl move database -dbname primary_dbname -sourcewc source_working_copy
   -patchedwc destination_working_copy -patchdataguard YES -schedule NOW
   ```

   The above command provides a parent job ID, which you can use as a single point of reference to monitor the progress of standby and primary database patching.

3. Monitor the progress of the entire patching operation.

   ```
   rhpctl query job -jobid job_id
   ```

4. Optionally, you can monitor the progress of the individual standby database patching operations.

   ```
   rhpctl query job -jobid job_id
   ```

**Related Topics**

- rhpctl query workingcopy
  Displays the configuration information of an existing working copy.
- rhpctl move database
  Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.
- rhpctl query job
  Queries the current status of a scheduled job with a specific job ID.

# Upgrading Oracle Database

Fleet Patching and Provisioning provides two options for upgrading Oracle Database. Both options are performed with a single command.

The `rhpctl upgrade database` command performs a traditional upgrade incurring downtime. The `rhpctl zdtupgrade database` command performs an Oracle RAC or Oracle RAC One Node upgrade with minimal or no downtime.

You can use Fleet Patching and Provisioning to provision, scale, and patch Oracle Database 19c and later releases. You can also upgrade Oracle Databases from 19c and 21c to Oracle Database 23ai. Refer to *Oracle Database Upgrade Guide* for information about Oracle Database direct upgrade paths.

> **Note:**
>
> The version of Oracle Grid Infrastructure on which the pre-upgrade database is running can be up to two RUs lower than the version of the database to which you are upgrading. For example, if the Oracle Database version is 23.5, then the Oracle Grid Infrastructure version must be 23.3 and higher.

The destination for the upgrade can be a working copy already provisioned, or you can choose to create the working copy of gold image as part of this operation.

The pre-upgrade database can be running on a working copy (a managed home that was provisioned by Fleet Patching and Provisioning) or on an unmanaged home. In the first case, you can roll back the upgrade process with a single RHPCTL command.

> **Note:**
>
> You can delete the source working copy at any time after completing an upgrade. Once you delete the working copy, however, you cannot perform a rollback. Also, use the `rhpctl delete workingcopy` command (as opposed to `rm`, for example) to remove the source working copy to keep the Fleet Patching and Provisioning inventory correct.

### Oracle Database AutoUpgrade

The AutoUpgrade utility identifies issues before upgrades, performs pre- and post-upgrade actions, deploys upgrades, and starts the upgraded Oracle Database. AutoUpgrade is included with each release update (RU).

Oracle FPP runs the `autoupgrade.jar` file that exists in the Oracle home. You can use the `autoupg` option with the `rhpctl upgrade database` command to automate the upgrade. However, before you create a gold image, Oracle strongly recommends that you download the latest AutoUpgrade version and add it to the gold image in the `$ORACLE_HOME/rdbms/admin` directory. The most recent AutoUpgrade version is always available from My Oracle Support Document 2485457.1.

Database Upgrade Assistant (DBUA) is desupported for any upgrades and migrations to Oracle Database 23ai, regardless of whether they are done on-premises, in any cloud, or in a hybrid approach. Oracle AutoUpgrade is the only supported tool to upgrade Oracle Database, with or without a migration to the multitenant (CDB) architecture.

**Related Topics**

- rhpctl upgrade database
- rhpctl zdtupgrade database
- Using AutoUpgade for Oracle Database Upgrades

# Zero-Downtime Upgrade

Using Oracle Fleet Patching and Provisioning, which automates and orchestrates database upgrades, you can upgrade an Oracle RAC or Oracle RAC One Node database with no disruption in service.

> **Note:**
>
> The Zero Downtime Upgrade (ZDU) feature of Oracle Fleet Patching and Provisioning (FPP) is deprecated in Oracle Database 23ai.

The zero-downtime upgrade process is resumable, restartable, and recoverable should any errors interrupt the process. You can fix the issue then re-run the command, and Oracle Fleet Patching and Provisioning continues from the error point. Oracle also provides hooks at the beginning and end of the zero-downtime upgrade process, allowing call outs to user-defined scripts, so you can customize the process.

You can use the zero-downtime upgrade process to upgrade databases that meet the following criteria:

- **Oracle Database release**: Oracle RAC and Oracle RAC One Node, with the following upgrade paths:

  11.2.0.4 to: 12.1.0.2, 12.2, 18c
  12.1.0.2 to: 12.2, 18c, 19c, 21c
  12.2 to: 18c, 19c, 21c
  18c to: 18c, 19c, 21c
  19c to: 21c, 23ai
  21c to: 23ai

- **Oracle Fleet Patching and Provisioning management**: The source database home can either be unmanaged (not provisioned by Oracle Fleet Patching and Provisioning service) or managed (provisioned by Oracle Fleet Patching and Provisioning service)
- **Database state**: The source database must be in archive log mode.

### Upgrading Container Databases (CDB)

You can use Oracle Fleet Patching and Provisioning to upgrade CDBs but Oracle Fleet Patching and Provisioning does not support converting a non-CDB to a CDB during upgrade. To prepare for a zero-downtime upgrade, you complete configuration steps and validation checks. When you run a zero-downtime upgrade using Oracle Fleet Patching and Provisioning, you can stop the upgrade and resume it, if necessary. You can recover from any upgrade errors, and you can restart the upgrade. You also have the ability to insert calls to your own scripts during the upgrade, so you can customize your upgrade procedure.

### Zero-Downtime Upgrade Environment Prerequisites

- **Server environment**: Oracle Grid Infrastructure 19c - 23ai with Oracle Fleet Patching and Provisioning
- **Database hosts**: Databases hosted on one of the following platforms:
  - Oracle Grid Infrastructure 19c to 23ai Oracle Fleet Patching and Provisioning, Client or Server
  - Oracle Grid Infrastructure 12c (12.2.0.1) Fleet Patching and Provisioning Client
  - Oracle Grid Infrastructure 12c (12.1.0.2) `rhpclient`-less target
- **Database-specific prerequisites for the environment**: During an upgrade, Oracle Fleet Patching and Provisioning manages replication to a local data file to preserve transactions applied to the new database when it is ready. There are two possibilities for the local data file:

  Snap clone, which is available if the database data files and redo and archive redo logs are on Oracle ACFS file systems

  Full copy, for all other cases
- Oracle Fleet Patching and Provisioning requires either Oracle GoldenGate or Oracle Data Guard during a zero-downtime database upgrade. As part of the upgrade procedure, Oracle Fleet Patching and Provisioning configures and manages the Oracle GoldenGate deployment.
- Running a Zero-Downtime Upgrade Using Oracle GoldenGate for Replication
  Run a zero-downtime upgrade using Oracle GoldenGate for replication.
- Running a Zero-Downtime Upgrade Using Oracle Data Guard for Replication
  Run a zero-downtime upgrade using Oracle Data Guard for replication.
- Customizing Zero-Downtime Upgrades
  You can customize zero-downtime upgrades using the user-action framework of Fleet Patching and Provisioning.

# Running a Zero-Downtime Upgrade Using Oracle GoldenGate for Replication

Run a zero-downtime upgrade using Oracle GoldenGate for replication.

> **Note:**
>
> The Zero Downtime Upgrade (ZDU) feature of Oracle Fleet Patching and Provisioning (FPP) is deprecated in Oracle Database 23ai.

1. Prepare the Fleet Patching and Provisioning Server.

   Create gold images of the Oracle GoldenGate software in the image library of the Fleet Patching and Provisioning Server.

   > **Note:**
   >
   > You can download the Oracle GoldenGate software for your platform from Oracle eDelivery. The Oracle GoldenGate 12.3 installable kit contains the required software for both Oracle Database 11*g* and Oracle Database 12*c* databases.

   If you download the Oracle GoldenGate software, then extract the software home and perform a *software only* installation on the Fleet Patching and Provisioning Server.

   Create gold images of the Oracle GoldenGate software for both databases, as follows:

   ```
   $ rhpctl import image -image 112ggimage -path path -imagetype
   ORACLEGGSOFTWARE
   $ rhpctl import image -image 12ggimage -path path -imagetype
   ORACLEGGSOFTWARE
   ```

   In both of the preceding commands, `path` refers to the location of the Oracle GoldenGate software home on the Fleet Patching and Provisioning Server for each release of the database.

2. Prepare the Oracle FPP Client or `rhpclient`-less target database.

   Provision working copies of the Oracle GoldenGate software to the cluster hosting the database, as follows:

   ```
   $ rhpctl add workingcopy -workingcopy GG_Wcopy_11g -image 112ggimage -user
     user_name -node 12102_cluster_node -path path {-root | -sudouser
   user_name
     -sudopath sudo_bin_path}
   $ rhpctl add workingcopy -workingcopy GG_Wcopy_12c -image 12ggimage -user
     user_name -node 12102_cluster_node -path path {-root | -sudouser
   user_name
     -sudopath sudo_bin_path}
   ```

   If the database is hosted on the Oracle Fleet Patching and Provisioning Server, itself, then neither the `-targetnode` nor `-client` parameters are required.

   > **Note:**
   >
   > Working copy names must be unique, therefore you must use a different working copy name on subsequent Oracle FPP Clients and `rhpclient`-less targets. You can create unique working copy names by including the name of the Oracle FPP Client or `rhpclient`-less target name in the working copy name.

3. Provision a working copy of the Oracle Database 12*c* software home to the Oracle FPP Client or `rhpclient`-less target.

> **Note:**
>
> You can do this preparation ahead of the maintenance window without disrupting any operations on the server.

- You can run the upgrade command on the Oracle Fleet Patching and Provisioning Server to upgrade a database hosted on the server, an Oracle Database 12*c* release 1 (12.1.0.2) `rhpclient`-less target, or a database hosted on an Oracle Fleet Patching and Provisioning Client 12*c* release 2 (12.2.0.1) or later releases. You can also run the command on an Oracle Fleet Patching and Provisioning Client 18c or later to upgrade a database hosted on the Oracle FPP Client, itself.

  Use the upgrade command similar to the following:

  ```
  $ rhpctl zdtupgrade database -dbname sierra -destwc DB_Wcopy_121 -ggsrcwc
    GG_Wcopy_11g -ggdstwc GG_Wcopy_12c -targetnode 12102_cluster_node -root
  ```

  In the preceding command, *`12102_cluster_node`* refers to the Oracle Grid Infrastructure 12*c* release 1 (12.1.0.2) cluster hosting the database you want to upgrade.

**Related Topics**

- rhpctl import image
  Creates an image on the Fleet Patching and Provisioning Server.
- rhpctl add workingcopy
  Creates a working copy on a client cluster.
- rhpctl zdtupgrade database
  The Zero Downtime Upgrade command `rhpctl zdtupgrade database` enables zero downtime database upgrades for Oracle RAC and Oracle RAC One Node Oracle Database instances.

# Running a Zero-Downtime Upgrade Using Oracle Data Guard for Replication

Run a zero-downtime upgrade using Oracle Data Guard for replication.

> **Note:**
>
> The Zero Downtime Upgrade (ZDU) feature of Oracle Fleet Patching and Provisioning (FPP) is deprecated in Oracle Database 23ai.

You can run the zero-downtime upgrade command using Oracle Data Guard's transient logical standby (TLS) feature. All of the steps involved are orchestrated by the zero-downtime upgrade command.

After you provision the destination database Home, the following prerequisites must be met:

- Data Guard Broker is not enabled
- Flash recovery area (FRA) is configured

- The following example of a zero-downtime upgrade using Oracle Data Guard upgrades an Oracle Database 19c, `sierra`, running on the cluster, which includes a node, `targetclust003`, to an Oracle Database 23ai (the destination working copy, which was provisioned from a Gold Image stored on the Fleet Patching and Provisioning Server named `rhps.example.com`):

```
$ rhpctl zdtupgrade database -dbname sierra -destwc WC235DB4344 -
clonedatadg DBDATA -targetnode node90743 -root

Enter user "root" password:
node90753.example.com: starting zero downtime upgrade operation ...
node90753.example.com: verifying patches applied to Oracle homes ...
node90753.example.com: verifying if database "sierra" can be upgraded with
zero downtime ...
node90743: 15:09:10.459: Verifying whether database "sierra" can be
cloned ...
node90743: 15:09:10.462: Verifying that database "sierra" is a primary
database ...
node90743: 15:09:14.672: Verifying that connections can be created to
database "sierra" ...
< ... >
node90743: 15:14:58.015: Starting redo apply ...
node90743: 15:15:07.133: Configuring primary database "sierra" ...
##################################################################
node90753.example.com: retrieving information about database "xmvotkvd" ...
node90753.example.com: creating services for snapshot database
"xmvotkvd" ...
##################################################################
node90743: 15:15:33.640: Macro step 1: Getting information and validating
setup ...
< ... >
node90743: 15:16:02.844: Macro step 2: Backing up user environment in case
upgrade is aborted  ...
node90743: 15:16:02.848: Stopping media recovery for database
"xmvotkvd" ...
node90743: 15:16:05.858: Creating initial restore point
"NZDRU_0000_0001" ...
< ... >
node90743: 15:16:17.611: Macro step 3: Creating transient logical standby
from existing physical standby ...
node90743: 15:16:18.719: Stopping instance "xmvotkvd2" of database
"xmvotkvd" ...
node90743: 15:16:43.187: Verifying that database "sierra" is a primary
database ...
< ... >
node90743: 15:19:27.158: Macro step 4: Upgrading transient logical standby
database ...
node90743: 15:20:27.272: Disabling service "sierrasvc" of database
"xmvotkvd" ...
node90743: 16:36:54.684: Macro step 5: Validating upgraded transient
logical standby database ...
node90743: 16:37:09.576: Creating checkpoint "NZDRU_0301" for database
"xmvotkvd" during stage "3" and task "1" ...
node90743: 16:37:09.579: Stopping media recovery for database
"xmvotkvd" ...
node90743: 16:37:10.792: Creating restore point "NZDRU_0301" for database
```

```
"xmvotkvd" ...
node90743: 16:37:11.998: Macro step 6: Switching role of transient logical
standby database ...
node90743: 16:37:12.002: Verifying that database "sierra" is a primary
database ...
< ... >
node90743: 16:39:07.425: Macro step 7: Flashback former primary database
to pre-upgrade restore point and convert to physical standby ...
node90743: 16:39:08.833: Stopping instance "sierra2" of database
"sierra" ...
< ... >
node90743: 16:41:17.138: Macro step 8: Recovering former primary
database ...
node90743: 16:41:19.045: Verifying that database "sierra" is mounted ...
< ... >
node90743: 17:20:21.378: Macro step 9: Switching back ...
< ... >
########################################################################
node90753.example.com: deleting snapshot database "xmvotkvd" ...
```

# Customizing Zero-Downtime Upgrades

You can customize zero-downtime upgrades using the user-action framework of Fleet Patching and Provisioning.

> **Note:**
>
> The Zero Downtime Upgrade (ZDU) feature of Oracle Fleet Patching and Provisioning (FPP) is deprecated in Oracle Database 23ai.

To use the user-action framework, you can provide a separate script for any or all of the points listed in the overall process.

**Table 6-1 Zero-Downtime Upgrade Plugins**

| Plugin Type | Pre or Post | Plugin runs... |
|---|---|---|
| ZDTUPGRADE_DATABASE | Pre | Before Fleet Patching and Provisioning starts zero-downtime upgrade. |
| | Post | After Fleet Patching and Provisioning completes zero-downtime upgrade. |
| ZDTUPGRADE_DATABASE_SNAPDB | Pre | Before creating the snapshot or full-clone database. |
| | Post | After starting the snapshot or full-clone database (but before switching over). |
| ZDTUPGRADE_DATABASE_DBUA | Pre | Before running DBUA (after switching over). |
| | Post | After DBUA completes. |
| ZDTUPGRADE_DATABASE_SWITCHBACK | Pre | Before switching back users to the upgraded source database. |
| | Post | After switching back users to the upgraded source database (before deleting snapshot or full-clone database). |

- To register a plugin to be run during a zero-downtime upgrade, run the following command:

```
$ rhpctl add useraction -useraction user_action_name -actionscript
script_name
  {-pre | -post} -optype {ZDTUPGRADE_DATABASE | ZDTUPGRADE_DATABASE_SNAPDB
|
   ZDTUPGRADE_DATABASE_DBUA | ZDTUPGRADE_DATABASE_SWITCHBACK}
```

  You can specify run-time input to the plugins using the `-useractiondata` option of the `rhpctl zdtupgrade database` command.

7

# Updating Oracle Exadata Infrastructure

The Oracle Fleet Patching and Provisioning Server provides an efficient and secure platform for updating Oracle Exadata Infrastructure.

- About Oracle Exadata Infrastructure Patching
  Oracle Fleet Patching and Provisioning (Oracle FPP) supports rolling patching of Oracle Exadata infrastructure, including storage cells, network switches, and database nodes.
- Updating Oracle Exadata Storage Server
  Update Oracle Exadata storage servers to a higher Oracle Exadata software version.
- Rolling Back Oracle Exadata Storage Server Patch
  Rollback successfully updated Oracle Exadata storage servers to their previous Oracle Exadata software version.
- Updating Oracle Exadata Database Node
  Update Oracle Exadata database servers to a higher Oracle Exadata software version.
- Rolling Back Oracle Exadata Database Node Patch
  Rollback Oracle Exadata database servers to their previous Oracle Exadata software version.
- Updating Oracle Exadata RoCE Switches
  Update Oracle Exadata RDMA over Converged Ethernet (RoCE) switches to a higher RoCE switch firmware version.
- Downgrading Oracle Exadata RoCE Switches
  Downgrade successfully updated Oracle Exadata RDMA over Converged Ethernet (RoCE) switches to the older RoCE switch firmware version as determined by the current Oracle Exadata release.
- Updating Oracle Exadata InfiniBand Switches
  Update Oracle Exadata InfiniBand switches to a higher InfiniBand switch firmware version.
- Downgrading Oracle Exadata InfiniBand Switches
  Downgrade successfully updated Oracle Exadata InfiniBand switches to the older InfiniBand switch firmware version as determined by the current Oracle Exadata release.
- Combined Oracle Exadata Database Server and Grid Infrastructure Update
  With combined Oracle Exadata database server and Oracle Grid Infrastructure update you can utilize the functionality of multiple independent capabilities.

## About Oracle Exadata Infrastructure Patching

Oracle Fleet Patching and Provisioning (Oracle FPP) supports rolling patching of Oracle Exadata infrastructure, including storage cells, network switches, and database nodes.

Patching Oracle Exadata infrastructure involves choosing a driving node from which you start the patching operation. A driving node can be either the Oracle FPP Server, or any of the Oracle FPP Client nodes.

Use the `-server` option to initiate the patching operation from the Oracle FPP Server. For this patch driving mode, the workflow is as follows:

1. Establish passwordless SSH `root` user equivalence from Oracle FPP Server to the target node.
2. Evaluate the patch operation impact (`-eval` precheck).
3. Apply the patch using the `rhpctl update exadata` command.

Use the `-client` *`client_node_name`* option to initiate the patching operation from the Oracle FPP Client. For this patch driving mode, the workflow is as follows:

1. Establish passwordless SSH `root` user equivalence from Oracle FPP Client to the target node.
2. Deploy the image on the target node using the `rhpctl import image` command.
3. Evaluate the patch operation impact (`-eval` precheck).
4. Apply the patch using the `rhpctl update exadata` command.

If the patch operation fails for any reason, then Oracle FPP enables you to rollback the cell server and database node patches, and downgrade the network switch patches.

# Updating Oracle Exadata Storage Server

Update Oracle Exadata storage servers to a higher Oracle Exadata software version.

1. Create an Oracle Exadata storage server image.

   The following command imports an Oracle Exadata storage server image to the Oracle FPP Server as an image. In the example, `image` specifies the name of the storage server image that you want to add, `imagetype` specifies `EXAPATCHSOFTWARE` for Oracle Exadata software, `version` specifies the version of the Oracle Exadata software, and `path` specifies the absolute path location of the storage server image that you want to import.

   ```
   rhpctl import image -image image_name -imagetype EXAPATCHSOFTWARE
   -version software_version -path absolute_path
   ```

   When you import a storage server image with this command, the `version` parameter must be the exact version of the storage server software required by the `patchmgr` on the node, for example `23.4.0.0.0.221111.1`. The `path` parameter should contain storage server update zip files.
2. Deploy the Oracle Exadata storage server image to the client cluster.

   > **Note:**
   >
   > You need to deploy the image only when running the patch operation from a client. This step is not required when running the patch operation from the Oracle FPP Server.

   The following command deploys an Oracle Exadata storage server image to a client cluster. In the example, `image` specifies the name of the Oracle Exadata storage server image that you want to deploy, `client` specifies the name of the cluster to which you want

to deploy the image, and `path` specifies the absolute path location for deploying the Oracle Exadata storage server image on the `rhpclient`-less target or client side.

```
rhpctl deploy image -image image_name
-client client_cluster_name -path image_file_path
```

3. Evaluate the current configuration and perform pre-upgrade checks.

   The following command evaluates the current configuration and performs pre-upgrade checks. In the example, `cells` specifies the list of storage servers, `image` specifies the name of the Oracle Exadata storage server image that you want to use for updating, and `client` specifies the name of the cluster in which you want to update the storage server.

   ```
   rhpctl update exadata -cells comma_separated_list_of_cells -image
   image_name -client client_cluster_name
   -patchmgrargs "patch_manager_arguments" -patchmgrdrivingsystem
   patchmgr_driving_node -eval
   ```

4. Update Oracle Exadata storage server with the new image.

   ```
   rhpctl update exadata -cells comma_separated_list_of_cells -image
   image_name -client client_cluster_name
   -patchmgrargs "patch_manager_arguments" -patchmgrdrivingsystem
   patchmgr_driving_node
   ```

5. Query the Oracle Exadata storage server image to check the image type:

   ```
   rhpctl query image -image cellserver1
   Image name: cellserver1
   Owner: user@server10
   Site: server10
   ...
   ...
   Image Type: Exadata Storage server
   Complete: TRUE
   ```

**Related Topics**

- rhpctl update exadata
  Patches an Oracle Exadata system.

# Rolling Back Oracle Exadata Storage Server Patch

Rollback successfully updated Oracle Exadata storage servers to their previous Oracle Exadata software version.

1. Check the current version of the Oracle Exadata storage server patch.
2. Roll back the Oracle Exadata storage server patch.

   ```
   rhpctl update exadata -cells comma_separated_list_of_cells -image
   image_name -client client_cluster_name
   -patchmgrargs "patch_manager_arguments"
   ```

3. Check the current version of the Oracle Exadata storage server patch to make sure that the rollback is successful.

# Updating Oracle Exadata Database Node

Update Oracle Exadata database servers to a higher Oracle Exadata software version.

> **Note:**
>
> You must download the latest version of the Patch Manager before starting the patch operation. Refer to my Oracle Support note 888828.1 to download the latest version of the Patch Manager software: https://support.oracle.com/rs?type=doc&id=888828.1.

1. Create an Oracle Exadata Database node image.

   The following command imports an Oracle Exadata image to the Oracle FPP Server as an image. In the example, `image` specifies the name of the Oracle Exadata image that you want to add, `imagetype` specifies `EXAPATCHSOFTWARE` for Oracle Exadata software, `version` specifies the version of the Oracle Exadata software, and `path` specifies the absolute path location of the Oracle Exadata software home that you want to import.

   ```
   rhpctl import image -image image_name -imagetype EXAPATCHSOFTWARE
   -version software_version -path absolute_path
   ```

   When you import an Oracle Exadata software home with this command, the `version` parameter must be the exact version of the Oracle Exadata software required by the `patchmgr` on the database node, for example `23.4.0.0.0.221111.1`. The `path` parameter should contain Oracle Exadata update zip files.

2. Deploy the Oracle Exadata Database node image to the client cluster.

   > **Note:**
   >
   > You need to deploy the image only when running the patch operation from an Oracle FPP Client. This step is not required when running the patch operation from the Oracle FPP Server.

   The following command deploys an Oracle Exadata image to a client cluster. In the example, `image` specifies the name of the Oracle Exadata image that you want to deploy, `client` specifies the name of the cluster to which you want to deploy the image, and `path` specifies the absolute path location for deploying the Oracle Exadata software home on the `rhpclient`-less target or client side.

   ```
   rhpctl deploy image -image image_name
   -client client_cluster_name -path image_file_path
   ```

   The `targetnode` parameter is required if the node hosting the home is not a Oracle Fleet Patching and Provisioning Client. If the `rhpclient`-less target or client option is not specified, then the Oracle Exadata image is deployed to the Oracle Fleet Patching and Provisioning Server.

3. Evaluate the current configuration and perform pre-upgrade checks.

The following command evaluates the current configuration and performs pre-upgrade checks. In the example, `image` specifies the name of the Oracle Exadata image that you want to use for update, `iso_repo` specifies the image in the ISO repository, `pathmgrloc` specifies the patch manager location, and `client` specifies the name of the cluster in which you want to update database nodes.

```
rhpctl update exadata -dbnodes comma_separates_list_of_nodes -image
image_name
-iso_repo iso_image_name -client client_cluster_name -patchmgrloc
patch_mgr_loc
-patchmgrargs "patch_manager_arguments" -liveupdate full -eval
```

- If you do not specify the list of nodes for `-dbnodes`, then Oracle FPP automatically discovers all active database nodes in the cluster.
- If the `client` option is not specified when issuing the command, then database node update is performed on the Oracle Fleet Patching and Provisioning Server.

4. Create a backup of the current configuration.

```
rhpctl update exadata -dbnodes {comma_separates_list_of_nodes | [-batches
"comma_separated_list_of_batches]}
-image image_name -iso_repo iso_image_name -client client_cluster_name
-patchmgrloc patch_mgr_loc -patchmgrargs "patch_manager_arguments" -backup
```

5. Update Oracle Exadata Database node with the new image.

```
rhpctl update exadata -dbnodes {comma_separates_list_of_nodes | [-batches
"comma_separated_list_of_batches"]}
-image image_name -iso_repo iso_image_name -client client_cluster_name
-patchmgrloc patch_mgr_loc -patchmgrargs "patch_manager_arguments" -
liveupdate full
```

6. Query the Oracle Exadata client to check the client configuration and Exadata live update information:

```
rhpctl query client -client client1 -exadata

Audit ID: 90
Site: client1
Rapid Home Provisioning Client Version: 23.0.0.0.0
Rapid Home Provisioning Client Full Version: 23.6.0.0.0
Enabled: true
Host from which RHPC last registered: client1vm01.example.com
Port number last registered by RHPC: 8896
...
...
Image status: success
Exadata software version: 23.1.7.0.0.231012
Exadata Live Update Type: full
Exadata Live Update Version: 23.1.90.0.0.240226 (Live Update applied.
Reboot at any time to finalize outstanding items.)
Node type: GUEST
...
...
```

**Related Topics**

- rhpctl update exadata
  Patches an Oracle Exadata system.

# Rolling Back Oracle Exadata Database Node Patch

Rollback Oracle Exadata database servers to their previous Oracle Exadata software version.

1. Check the current version of the Oracle Exadata database node patch.
2. Roll back the Oracle Exadata database node patch.

   ```
   rhpctl update exadata -dbnodes comma_separates_list_of_nodes -image
   image_name
   -client client_cluster_name -patchmgrloc patch_mgr_loc -rollback
   ```

3. Check the current version of the Oracle Exadata database node patch to make sure that the rollback is successful.

# Updating Oracle Exadata RoCE Switches

Update Oracle Exadata RDMA over Converged Ethernet (RoCE) switches to a higher RoCE switch firmware version.

1. Create an Oracle Exadata RoCE switch image.

   The following command imports an Oracle Exadata RoCE switch image to the Oracle FPP Server as an image. In the example, `image` specifies the name of the RoCE switch image that you want to add, `imagetype` specifies `EXAPATCHSOFTWARE` for Oracle Exadata software, `version` specifies the version of the Oracle Exadata software, and `path` specifies the absolute path location of the RoCE switch image that you want to import.

   ```
   rhpctl import image -image image_name -imagetype EXAPATCHSOFTWARE
   -version software_version -path absolute_path
   ```

   When you import a RoCE switch image with this command, the `version` parameter must be the exact version of the RoCE switch software required by the `patchmgr` on the node, for example `23.4.0.0.0.221111.1`. The `path` parameter should contain RoCE switch update zip files.
2. Deploy the Oracle Exadata RoCE switch image to the client cluster.

   > **Note:**
   >
   > You need to deploy the image only when running the patch operation from a client. This step is not required when running the patch operation from the Oracle FPP Server.

   The following command deploys an Oracle Exadata RoCE switch image to a client cluster. In the example, `image` specifies the name of the Oracle Exadata RoCE switch image that you want to deploy, `client` specifies the name of the cluster to which you want to deploy

the image, and `path` specifies the absolute path location for deploying the Oracle Exadata RoCE switch image on the `rhpclient`-less target or client side.

```
rhpctl deploy image -image image_name
-client client_cluster_name -path image_file_path
```

3. Evaluate the current configuration and perform pre-upgrade checks for RoCE switches.

   The following command evaluates the current configuration and performs pre-upgrade checks. In the example, `image` specifies the name of the Oracle Exadata RoCE switch image that you want to use for update and `client` specifies the name of the cluster in which you want to update RoCE switches.

   ```
   rhpctl update exadata -roceswitches comma_separated_list_of_RoCE_nodes -
   image image_name
   -client client_cluster_name -eval
   ```

4. Update Oracle Exadata RoCE switches with the new image.

   ```
   rhpctl update exadata -roceswitches comma_separated_list_of_IB_nodes -
   image image_name
   -client client_cluster_name
   ```

5. Query the Oracle Exadata RoCE switch image to check the image type:

   ```
   rhpctl query image -image RoCE1
   Image name: RoCE1
   Owner: user@server10
   Site: server10
   ...
   ...
   Image Type: Exadata network switch
   Complete: TRUE
   ```

**Related Topics**

- rhpctl update exadata
  Patches an Oracle Exadata system.

# Downgrading Oracle Exadata RoCE Switches

Downgrade successfully updated Oracle Exadata RDMA over Converged Ethernet (RoCE) switches to the older RoCE switch firmware version as determined by the current Oracle Exadata release.

1. Check the current version of the Oracle Exadata RoCE switch patch.
2. Evaluate the current configuration and perform pre-downgrade checks for RoCE switches.

   ```
   rhpctl update exadata -roceswitches comma_separated_list_of_IB_nodes -
   image image_name
   -client client_cluster_name -patchmgrargs "-patch_manager_arguments"
   -patchmgrdrivingsystem patchmgr_driving_node -downgrade -eval
   ```

3. Downgrade the Oracle Exadata RoCE switch patch.

```
rhpctl update exadata -roceswitches comma_separated_list_of_IB_nodes -
image image_name
-client client_cluster_name -patchmgrargs "-patch_manager_arguments"
-patchmgrdrivingsystem patchmgr_driving_node -downgrade
```

4. Check the current version of the Oracle Exadata RoCE switch patch to make sure that the downgrade is successful.

# Updating Oracle Exadata InfiniBand Switches

Update Oracle Exadata InfiniBand switches to a higher InfiniBand switch firmware version.

1. Create an Oracle Exadata InfiniBand switch image.

   The following command imports an Oracle Exadata InfiniBand switch image to the Oracle FPP Server as an image. In the example, `image` specifies the name of the InfiniBand switch image that you want to add, `imagetype` specifies `EXAPATCHSOFTWARE` for Oracle Exadata software, `version` specifies the version of the Oracle Exadata software, and `path` specifies the absolute path location of the InfiniBand switch image that you want to import.

```
rhpctl import image -image image_name -imagetype EXAPATCHSOFTWARE
-version software_version -path absolute_path
```

   When you import an InfiniBand switch image with this command, the `version` parameter must be the exact version of the InfiniBand switch software required by the `patchmgr` on the node, for example `23.4.0.0.0.221111.1`. The `path` parameter should contain InfiniBand switch updated zip files.

2. Deploy the Oracle Exadata InfiniBand switch image to the client cluster.

> **Note:**
>
> You need to deploy the image only when running the patch operation from a client. This step is not required when running the patch operation from the Oracle FPP Server.

   The following command deploys an Oracle Exadata InfiniBand switch image to a client cluster. In the example, `image` specifies the name of the Oracle Exadata InfiniBand switch image that you want to deploy, `client` specifies the name of the cluster to which you want to deploy the image, and `path` specifies the absolute path location for deploying the Oracle Exadata InfiniBand switch image on the `rhpclient`-less target or client side.

```
rhpctl deploy image -image image_name
-client client_cluster_name -path image_file_path
```

3. Evaluate the current configuration and perform pre-upgrade checks for InfiniBand switches.

   The following command evaluates the current configuration and performs pre-upgrade checks. In the example, `image` specifies the name of the Oracle Exadata InfiniBand switch image that you want to use for update, `client` specifies the name of the cluster in which

you want to update database nodes, and `patchmgrdrivingsystem` specifies the node from which you start the update operation.

```
rhpctl update exadata -ibswitches comma_separated_list_of_IB_nodes -image
image_name
-client client_cluster_name -patchmgrdrivingsystem patchmgr_driving_node -
eval
```

> **Note:**
>
> You must specify the `patchmgrdrivingsystem` if the Oracle FPP Server does not have direct access to the Oracle Exadata InfiniBand switches and the patch operation needs to be performed through an intermediary `dom0` node

4. Update Oracle Exadata InfiniBand switches with the new image.

```
rhpctl update exadata -ibswitches comma_separated_list_of_IB_nodes -image
image_name
-client client_cluster_name -patchmgrdrivingsystem patchmgr_driving_node
```

5. Query the Oracle Exadata InfiniBand switch image to check the image type:

```
rhpctl query image -image infini1
Image name: infini1
Owner: user@server10
Site: server10
...
...
Image Type: Exadata network switch
Complete: TRUE
```

**Related Topics**

- rhpctl update exadata
  Patches an Oracle Exadata system.

# Downgrading Oracle Exadata InfiniBand Switches

Downgrade successfully updated Oracle Exadata InfiniBand switches to the older InfiniBand switch firmware version as determined by the current Oracle Exadata release.

1. Check the current version of the Oracle Exadata InfiniBand switch patch.
2. Evaluate the current configuration and perform pre-downgrade checks for InfiniBand switches.

```
rhpctl update exadata -ibswitches comma_separated_list_of_IB_nodes -image
image_name
-client client_cluster_name -patchmgrargs "patch_manager_arguments"
-patchmgrdrivingsystem patchmgr_driving_node -downgrade -eval
```

3. Downgrade the Oracle Exadata InfiniBand switch patch.

```
rhpctl update exadata -ibswitches comma_separated_list_of_IB_nodes -image
image_name
-client client_cluster_name -patchmgrargs "patch_manager_arguments"
-patchmgrdrivingsystem patchmgr_driving_node -downgrade
```

4. Check the current version of the Oracle Exadata InfiniBand switch patch to make sure that the downgrade is successful.

# Combined Oracle Exadata Database Server and Grid Infrastructure Update

With combined Oracle Exadata database server and Oracle Grid Infrastructure update you can utilize the functionality of multiple independent capabilities.

Patching Grid Infrastructure and updating the Exadata database nodes both require a shutdown and startup of every database instance on that node. This can take considerable time, depending on the number of applications running and the time it takes to shutdown instances and start them up. Performing both of these patching actions independently doubles the downtime on production databases. Using the combined patching feature of Oracle FPP automates both of these patching actions into a single integrated patching process that requires only one sequence of shutdown and startup of database instances on each node. The combined patching on multiple nodes in batches further brings down the overall patching window.

Oracle FPP internally uses the `patchmgr` tool to patch Exadata database nodes. The combined patching method uses an integrated flow of the inherent Oracle FPP implementation for Oracle Grid Infrastructure patching and then invokes the `patchmgr` tool to patch each Exadata database node.

To complete combined Oracle Exadata database node and Oracle Grid Infrastructure patching, you must perform the operations discussed in the following:

- Creating the Oracle Exadata Image
- Deploying the Oracle Exadata Update Image
- Combined Oracle Grid Infrastructure Move and Database Node Update

> **Note:**
>
> Creating and deploying an Oracle Exadata image does not require any downtime and you can perform both these operations before patching Oracle Grid Infrastructure and Oracle Exadata database. You need to create an Oracle Exadata image on the Oracle FPP server only once in a patching cycle, however, you need to deploy Oracle Exadata image, and patch Oracle Grid Infrastructure and Oracle Exadata database node on each server.

### Creating the Oracle Exadata Image

Use the `rhpctl import image` command to create the Oracle Exadata update image by copying the entire software contents from the specified path to the Oracle Fleet Patching and Provisioning Server (FPPS).

**Example 7-1 Creating an Oracle Exadata Update Image**

The following command imports an Oracle Exadata image to the Oracle FPP Server as an image. In the example, `image` specifies the name of the Oracle Exadata image that you want to add, `path` specifies the absolute path location of the Oracle Exadata software home that you want to import, `imagetype` specifies `EXAPATCHSOFTWARE` for Oracle Exadata software, and `version` specifies the version of the Oracle Exadata software.

```
$ rhpctl import image -image EXADATAIMAGEV1
   -path /tmp/ExadataPatchBundle -imagetype EXAPATCHSOFTWARE -version
19.2.2.0.0.190513.2
```

When you import an Oracle Exadata software home with this command, the `version` parameter must be the version of the Oracle Exadata software required by the `patchmgr` on the database node. The `path` parameter should contain Oracle Exadata update zip files.

> **See Also:**
>
> rhpctl import image for the complete syntax of the `rhpctl import image` command

**Deploying the Oracle Exadata Update Image**

Use the `rhpctl deploy image` command to propagate the Oracle Exadata update image to server.

**Example 7-2 Deploying an Oracle Exadata image**

The following command deploys an Oracle Exadata image to a client cluster. In the example, `image` specifies the name of the Oracle Exadata image that you want to deploy, `client` specifies the name of the cluster to which you want to deploy the image, and `path` specifies the absolute path location for deploying the Oracle Exadata software home on the `rhpclient`-less target or client side.

```
$ rhpctl deploy image -image EXADATAIMAGEV1 -client CLUSTER1 -path /
exadatasoftware
```

The `targetnode` parameter is required if the node hosting the home is not a Oracle Fleet Patching and Provisioning Client. If the `rhpclient`-less target or client option is not specified, then the Oracle Exadata image is deployed to the Oracle Fleet Patching and Provisioning Server.

> **See Also:**
>
> rhpctl deploy image for the complete syntax of the `rhpctl deploy image` command.

**Combined Oracle Grid Infrastructure Move and Database Node Update**

Use the `rhpctl move gihome` command to move the Oracle Grid Infrastructure software stack from one home to another while updating the Oracle Exadata database node.

**Example 7-3 Moving an Oracle Grid Infrastructure home and updating a database node**

The following example performs a combined Oracle Grid Infrastructure move and database node update on client cluster. In the example, `sourcewc` specifies the name of the source working copy, `destwc` specifies the name of the destination working copy to which you want to move the Oracle Grid Infrastructure home, `image` specifies the name of the Oracle Exadata image, `batches` specifies a comma-delimited list of batches of nodes where each batch is a comma-delimited list of node names enclosed in parentheses and node names are enclosed in double quotation marks ("") in the format: `"(nA,nB,...),(...,nY,nZ)"`, `iso_repo` specifies the image in the ISO repository, and `pathmgrloc` specifies the patch manager location.

```
$ rhpctl move gihome -sourcewc prodHomeV1 -destwc prodHomeV2 -image
EXADATAIMAGEV1
    -batches "(rac07box1,rac07box2,rac07box3),(rac07box4)"
   -patchmgrargs "-ignore_alerts" -iso_repo p28802055_192000_Linux-x86-64.zip
-client prodcluster
    -patchmgrloc /patchMgr/dbserver_patch_19.190306
```

With each invocation of the `rhpctl move gihome` command, FPP patches the database node first and then patches Oracle Grid Infrastructure. This is the processing order for each node in the specified batch.

If the first batch includes more than one database node, then FPP invokes `patchmgr` in parallel for all the nodes. As soon as a node completes the `patchmgr` operation, including the post `patchmgr` operations, FPP starts the Oracle Grid Infrastructure patching on that node. When the Oracle Grid Infrastructure patching completes on this node, FPP then begins patching with Oracle Grid Infrastructure patching on the other nodes when the database node patching completes on those nodes.

If rebooting a node is delayed because of a `patchmgr` failure or a `patchmgr` operation timeout, the `rhpctl move gihome` command can be resumed after the node is back up.

> **See Also:**
>
> rhpctl move gihome for the complete syntax of the `rhpctl move gihome` command

8

# Fleet Patching and Provisioning Postinstallation Tasks

Complete these postinstallation tasks after you configure Oracle Fleet Patching and Provisioning Server.

- Oracle Fleet Patching and Provisioning Security Postinstallation Tasks
  Perform these postinstallation tasks to make your Oracle Fleet Patching and Provisioning Server secure.
- Advanced Oracle Fleet Patching and Provisioning Configurations
  Use these advanced configurations of Oracle Fleet Patching and Provisioning Server to perform specialized tasks.
- Error Prevention and Automated Recovery Options
  Fleet Patching and Provisioning has error prevention and automated recovery options to assist you during maintenance operations.
- Fleet Patching and Provisioning Logs and Trace Files
  Use Oracle Fleet Patching and Provisioning logs and traces to obtain more information for identifying and debugging Oracle FPP Server and client errors.

## Oracle Fleet Patching and Provisioning Security Postinstallation Tasks

Perform these postinstallation tasks to make your Oracle Fleet Patching and Provisioning Server secure.

- Authentication Options for Oracle Fleet Patching and Provisioning Operations
  Some RHPCTL commands show authentication choices as an optional parameter.
- Oracle Fleet Patching and Provisioning Roles
  An administrator assigns roles to Oracle Fleet Patching and Provisioning users with access-level permissions defined for each role.
- Managing the Fleet Patching and Provisioning Client Password
  The Oracle Fleet Patching and Provisioning (Oracle FPP) Client uses a password stored internally to authenticate itself with the RHP server.
- Switching Oracle FPP Server to a New Custom Certificate
  You can switch the Oracle FPP Server to use a new custom security certificate to replace self-signed certificate or update expired certificate.
- Switching Oracle FPP Client to a New Custom Certificate
  You can switch the Oracle FPP Client to use a new custom security certificate to replace self-signed certificate or update expired certificate.
- Oracle Fleet Patching and Provisioning Server Auditing
  The Oracle Fleet Patching and Provisioning Server records the processing of all Oracle Fleet Patching and Provisioning operations, and also records whether those operations succeeded or failed.

# Authentication Options for Oracle Fleet Patching and Provisioning Operations

Some RHPCTL commands show authentication choices as an optional parameter.

Specifying an authentication option is not required when running an RHPCTL command on an Oracle Fleet Patching and Provisioning Client, nor when running an RHPCTL command on the Oracle Fleet Patching and Provisioning Server and operating on an Oracle Fleet Patching and Provisioning Client, because the server and client establish a trusted relationship when the client is created, and authentication is handled internally each time a transaction takes place. (The only condition for server/client communication under which an authentication option must be specified is when the server is provisioning a new Oracle Grid Infrastructure deployment—in this case, the client does not yet exist.)

To operate on an `rhpclient`-less target, you must provide the Oracle Fleet Patching and Provisioning Server with information allowing it to authenticate with the `rhpclient`-less target. The options are as follows:

- Provide the `root` password (on `stdin`) for the `rhpclient`-less target
- Provide the `sudo` user name, `sudo` binary path, and the password (`stdin`) for `rhpclient`-less target
- Provide a password (either `root` or `sudouser`) non-interactively from local encrypted store (using the `-cred` authentication parameter)
- Create credentials using the `rhpctl add credentials` command and provide credentials using the `-cred` option.
- Provide a path to the identity file stored on the Oracle Fleet Patching and Provisioning Server for SSL-encrypted passwordless authentication (using the `-auth sshkey` option)

### Passwordless Authentication Details

The Oracle Fleet Patching and Provisioning Server can authenticate to `rhpclient`-less targets over SSH using a key pair. To enable this option, you must establish user equivalence between the `crsusr` on the Oracle Fleet Patching and Provisioning Server and `root` or a `sudouser` on the `rhpclient`-less target.

> **Note:**
>
> The steps to create that equivalence are platform-dependent and so not shown in detail here. For Linux, see commands `ssh-keygen` to be run on the `rhpclient`-less target and `ssh-copy-id` to be run on the Oracle Fleet Patching and Provisioning Server.

For example, assuming that you have established user equivalency between `crsusr` on the Oracle Fleet Patching and Provisioning Server and `root` on the `rhpclient`-less target, `nonRHPClient4004.example.com`, and saved the key information on the Oracle Fleet Patching and Provisioning Server at `/home/oracle/rhp/ssh-key/key -path`, then the following

command will provision a copy of the specified gold image to the `rhpclient`-less target with passwordless authentication:

```
$ rhpctl add workingcopy -workingcopy db12102_160607wc1 -image db12102_160607
  -targetnode nonRHPClient4004.example.com -path /u01/app/oracle/12.1/rhp/
dbhome_1
  -oraclebase /u01/app/oracle -auth sshkey -arg1 user:root -arg2
   identity_file:/home/oracle/rhp/ssh-key/key
```

For equivalency between `crsusr` on the Oracle Fleet Patching and Provisioning Server and a privileged user (other than `root`) on the `rhpclient`-less target, the `-auth` portion of the command would be similar to the following:

```
-auth sshkey -arg1 user:ssh_user -arg2
identity_file:path_to_identity_file_on_RHPS
 -arg3 sudo_location:path_to_sudo_binary_on_target
```

**Related Topics**

- rhpctl add credentials
- rhpctl delete credentials
- rhpctl add workingcopy
  Creates a working copy on a client cluster.

# Oracle Fleet Patching and Provisioning Roles

An administrator assigns roles to Oracle Fleet Patching and Provisioning users with access-level permissions defined for each role.

Users on Oracle Fleet Patching and Provisioning Clients are also assigned specific roles. Oracle Fleet Patching and Provisioning includes basic built-in and composite built-in roles.

**Basic Built-In Roles**

The basic built-in roles and their functions are:

- GH_ROLE_ADMIN: An administrative role for everything related to roles. Users assigned this role are able to run `rhpctl verb role` commands.
- GH_SITE_ADMIN: An administrative role for everything related to Oracle Fleet Patching and Provisioning Clients. Users assigned this role are able to run `rhpctl verb client` commands.
- GH_SERIES_ADMIN: An administrative role for everything related to image series. Users assigned this role are able to run `rhpctl verb series` commands.
- GH_SERIES_CONTRIB: Users assigned this role can add images to a series using the `rhpctl insertimage series` command, or delete images from a series using the `rhpctl deleteimage series` command.
- GH_WC_ADMIN: An administrative role for everything related to working copies of gold images. Users assigned this role are able to run `rhpctl verb workingcopy` commands.
- GH_WC_OPER: A role that enables users to create a working copy of a gold image for themselves or others using the `rhpctl add workingcopy` command with the `-user` option (when creating for others). Users assigned this role do not have administrative privileges and can only administer the working copies of gold images that they create.

- GH_WC_USER: A role that enables users to create a working copy of a gold image using the `rhpctl add workingcopy` command. Users assigned this role do not have administrative privileges and can only delete working copies that they create.
- GH_IMG_ADMIN: An administrative role for everything related to images. Users assigned this role are able to run `rhpctl verb image` commands.
- GH_IMG_USER: A role that enables users to create an image using the `rhpctl add | import image` commands. Users assigned this role do not have administrative privileges and can only delete images that they create.
- GH_IMG_TESTABLE: A role that enables users to add a working copy of an image that is in the `TESTABLE` state. Users assigned this role must also be assigned either the GH_WC_ADMIN role or the GH_WC_USER role to add a working copy.
- GH_IMG_RESTRICT: A role that enables users to add a working copy from an image that is in the `RESTRICTED` state. Users assigned this role must also be assigned either the GH_WC_ADMIN role or the GH_WC_USER role to add a working copy.
- GH_IMG_PUBLISH: Users assigned this role can promote an image to another state or retract an image from the `PUBLISHED` state to either the `TESTABLE` or `RESTRICTED` state.
- GH_IMG_VISIBILITY: Users assigned this role can modify access to promoted or published images using the `rhpctl allow | disallow image` commands.
- GH_AUTHENTICATED_USER: Users assigned to this role can perform any operation in an Oracle Fleet Patching and Provisioning Client.
- GH_CLIENT_ACCESS: Any user created automatically inherits this role. The `GH_CLIENT_ACCESS` role includes the `GH_AUTHENTICATED_USER` built-in role.
- GH_ROOT_UA_CREATE: A role that enables users to create a root user action. Users assigned this role can run the `rhpctl add useraction` command with the `-runasroot` option.
- GH_ROOT_UA_ASSOCIATE: A role that enables users to associate a root user action with the `-imagetype` option. Users assigned this role can associate an existing root user action to an image type.
- GH_ROOT_UA_USE: A role that enables users to perform a root user action within the operation selected at user action creation.

**Composite Built-In Roles**

The composite built-in roles and their functions are:

- GH_SA: The Oracle Grid Infrastructure user on an Oracle Fleet Patching and Provisioning Server automatically inherits this role.

  The GH_SA role includes the following basic built-in roles: GH_ROLE_ADMIN, GH_SITE_ADMIN, GH_SERIES_ADMIN, GH_SERIES_CONTRIB, GH_WC_ADMIN, GH_IMG_ADMIN, GH_IMG_TESTABLE, GH_IMG_RESTRICT, GH_IMG_PUBLISH, and GH_IMG_VISIBILITY.
- GH_CA: The Oracle Grid Infrastructure user on an Oracle Fleet Patching and Provisioning Client automatically inherits this role.

  The GH_CA role includes the following basic built-in roles: GH_SERIES_ADMIN, GH_SERIES_CONTRIB, GH_WC_ADMIN, GH_IMG_ADMIN, GH_IMG_TESTABLE, GH_IMG_RESTRICT, GH_IMG_PUBLISH, and GH_IMG_VISIBILITY.
- GH_OPER: This role includes the following built-in roles: GH_WC_OPER, GH_SERIES_ADMIN, GH_IMG_TESTABLE, GH_IMG_RESTRICT, and GH_IMG_USER. Users assigned this role can delete only images that they have created.

Consider a gold image called `G1` that is available on the Oracle Fleet Patching and Provisioning Server.

Further consider that a user, `U1`, on an Oracle Fleet Patching and Provisioning Client, `Cl1`, has the GH_WC_USER role. If `U1` requests to provision an Oracle home based on the gold image `G1`, then `U1` can do so, because of the permissions granted by the GH_WC_USER role. If `U1` requests to delete `G1`, however, then that request would be denied because the GH_WC_USER role does not have the necessary permissions.

The Oracle Fleet Patching and Provisioning Server can associate user-role mappings to the Oracle Fleet Patching and Provisioning Client. After the Oracle Fleet Patching and Provisioning Server delegates user-role mappings, the Oracle Fleet Patching and Provisioning Client can then modify user-role mappings on the Oracle Fleet Patching and Provisioning Server for all users that belong to the Oracle Fleet Patching and Provisioning Client. This is implied by the fact that only the Oracle Fleet Patching and Provisioning Server qualifies user IDs from an Oracle Fleet Patching and Provisioning Client site with the client cluster name of that site. Thus, the Oracle Fleet Patching and Provisioning Client `CL1` will not be able to update user mappings of a user on `CL2`, where `CL2` is the cluster name of a different Oracle Fleet Patching and Provisioning Client.

- Creating Users and Assigning Roles for Fleet Patching and Provisioning Client Cluster Users
  Oracle Fleet Patching and Provisioning (Oracle FPP) enables you to create users and assign roles to them when you create an Oracle FPP client.

## Creating Users and Assigning Roles for Fleet Patching and Provisioning Client Cluster Users

Oracle Fleet Patching and Provisioning (Oracle FPP) enables you to create users and assign roles to them when you create an Oracle FPP client.

When you create a Fleet Patching and Provisioning Client with the `rhpctl add client` command, you can use the `-maproles` parameter to create users and assign roles to them. You can associate multiple users with roles, or you can assign a single user multiple roles with this command.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 21c, the `-maproles` parameter is deprecated. This parameter can be desupported in a future release.

After the client has been created, you can add and remove roles for users using the `rhpctl grant role` command and the `rhpctl revoke role`, respectively.

# Managing the Fleet Patching and Provisioning Client Password

The Oracle Fleet Patching and Provisioning (Oracle FPP) Client uses a password stored internally to authenticate itself with the RHP server.

You cannot query the Oracle FPP Client password, however, if for some reason, you are required to reset this password, then you can do so, as follows, on the RHP server cluster:

1. Run the following command on the Fleet Patching and Provisioning Server cluster to generate a new password and store it in the client credential:

```
$ rhpctl modify client -client client_name -password
```

2. Run the following command on the Fleet Patching and Provisioning Server cluster to generate a credential file:

   ```
   $ rhpctl export client -client client_name -clientdata file_path
   ```

   For example, to generate a credential file for a Fleet Patching and Provisioning Client named `mjk9394`:

   ```
   $ rhpctl export client -client mjk9394 -clientdata /tmp/mjk9394.xml
   ```

3. Continuing with the preceding example, transport the generated credential file securely to the Fleet Patching and Provisioning Client cluster and then run the following command on any node in the Fleet Patching and Provisioning Client cluster:

   ```
   $ srvctl modify rhpclient -clientdata path_to_mjk9394.xml
   ```

4. Restart the Fleet Patching and Provisioning Client daemon by running the following commands on the Fleet Patching and Provisioning Client cluster:

   ```
   $ srvctl stop rhpclient
   $ srvctl start rhpclient
   ```

# Switching Oracle FPP Server to a New Custom Certificate

You can switch the Oracle FPP Server to use a new custom security certificate to replace self-signed certificate or update expired certificate.

When you switch Oracle FPP Server to use a new custom certificate, communication with the Oracle FPP clients is broken. You must complete the following steps to restore the communication.

1. Perform a sanity check from the Oracle FPP Client to ensure that the client can connect to the server.

   ```
   $ rhpctl query server
   ```

2. Generate a P12 file from the security certificate that you want to use for the Oracle FPP Server configuration.

   ```
   $ openssl pkcs12 -export -inkey ${SERVER_KEY} -passin pass:${PASS} -in ${SERVER_CRT}
   -name orakey -certfile ${CA_CRT} -caname cakey -out ${P12_FILE} -password pass:${P12_PASS}
   ```

3. Configure Oracle FPP Server to use the new custom certificate from the generated P12 file and provide a name for this certificate.

   ```
   $ srvctl stop rhpserver
   # srvctl modify rhpserver -p12certpath P12_file_path -certname cert_name
   $ srvctl start rhpserver
   ```

4. Export credentials from the Oracle FPP Server to a client data file.

   ```
   $ rhpctl export client -client cluster_name -clientdata file_path
   ```

5. Copy the generated credential file securely to the Oracle FPP Client cluster.

6. Perform a sanity check from the Oracle FPP Client.

```
$ rhpctl query server
```

This command is expected to fail with the `PRGO-1068` error.

7. Configure the Oracle FPP Client to use a new custom certificate.
   a. Generate a P12 file from the security certificate that you want to use for the Oracle FPP Client configuration.

```
$ openssl pkcs12 -export -inkey ${SERVER_KEY} -passin pass:${PASS} -in ${SERVER_CRT}
-name orakey -certfile ${CA_CRT} -caname cakey -out ${P12_FILE} -
password pass:${P12_PASS}
```

   b. Configure Oracle FPP Client to use the new custom certificate from the generated P12 file and provide a name for this certificate.

```
$ srvctl stop rhpclient
# srvctl modify rhpclient -p12certpath P12_file_path -certname cert_name
```

> **Note:**
>
> It is not mandatory to use custom security certificate for all Oracle FPP Clients, even if you are using it for the Oracle FPP Server. Similarly, you can use custom security certificate for one or more Oracle FPP Clients, but not for the Oracle FPP Server.

8. Update the public security key of the Oracle FPP Server.

```
# srvctl modify rhpclient -clientdata file_path
$ srvctl start rhpclient
```

This command updates Oracle FPP Client's public key on the Oracle FPP Server.

9. Perform a sanity check from the Oracle FPP Client to ensure that the client can connect to the server.

```
$ rhpctl query server
```

10. Repeat steps 4 through 9 on all other Oracle FPP Client cluster nodes.

## Switching Oracle FPP Client to a New Custom Certificate

You can switch the Oracle FPP Client to use a new custom security certificate to replace self-signed certificate or update expired certificate.

1. Perform a sanity check from the Oracle FPP Client to ensure that the client can connect to the server.

```
$ rhpctl query server
```

2. Generate a P12 file from the security certificate that you want to use for the Oracle FPP Client configuration.

```
$ openssl pkcs12 -export -inkey ${SERVER_KEY} -passin pass:${PASS} -in $
{SERVER_CRT}
-name orakey -certfile ${CA_CRT} -caname cakey -out ${P12_FILE} -password
pass:${P12_PASS}
```

3. Configure Oracle FPP Client to use the new custom certificate from the generated P12 file and provide a name for this certificate.

```
$ srvctl stop rhpclient
# srvctl modify rhpserver -p12certpath P12_file_path -certname cert_name
```

4. Configure the Oracle FPP Client to store Oracle FPP Server's public key along with the new custom certificate.
   a. On the Oracle FPP Server, export credentials from the Oracle FPP Server to a client data file and copy the generated credential file securely to the Oracle FPP Client cluster..

```
$ rhpctl export client -client cluster_name -clientdata file_path
```

   b. On the Oracle FPP Client, update the public security key of the Oracle FPP Server..

```
# srvctl modify rhpclient -clientdata file_path
$ srvctl start rhpclient
```

   This command updates Oracle FPP Client's public key on the Oracle FPP Server.

5. Perform a sanity check from the Oracle FPP Client to ensure that the client can connect to the server.

```
$ rhpctl query server
```

6. If you want to configure custom security certificate on other Oracle FPP Clients, then repeat steps 1 through 5 on those Oracle FPP Client cluster nodes.

## Oracle Fleet Patching and Provisioning Server Auditing

The Oracle Fleet Patching and Provisioning Server records the processing of all Oracle Fleet Patching and Provisioning operations, and also records whether those operations succeeded or failed.

An audit mechanism enables administrators to query the audit log in a variety of dimensions, and also to manage its contents and size.

# Advanced Oracle Fleet Patching and Provisioning Configurations

Use these advanced configurations of Oracle Fleet Patching and Provisioning Server to perform specialized tasks.

- User-Defined Actions
  You can create actions for various Oracle Fleet Patching and Provisioning operations, such as import image, add and delete working copy, and add, delete, move, and upgrade a software home.
- Oracle Fleet Patching and Provisioning Notifications
  The Oracle Fleet Patching and Provisioning Server is the central repository for the software homes available to the data center. Therefore, it is essential for administrators throughout the data center to be aware of changes to the inventory that may impact their areas of responsibility.
- Job Scheduler for Operations
  The Oracle Fleet Patching and Provisioning job scheduler provides you with a mechanism to submit operations at a scheduled time instead of running the command immediately, querying the metadata of the job, and then deleting the job from the repository.
- Patching Oracle Grid Infrastructure and Oracle Database Using Batches
  You can sequentially process batches of nodes, with a number of nodes in each batch being restarted in parallel.
- Combined Oracle Exadata Database Server and Grid Infrastructure Update
  With combined Oracle Exadata database server and Oracle Grid Infrastructure update you can utilize the functionality of multiple independent capabilities.
- Gold Image Distribution Among Oracle Fleet Patching and Provisioning Servers
  Oracle Fleet Patching and Provisioning can automatically share and synchronize gold images between Oracle Fleet Patching and Provisioning Servers.

# User-Defined Actions

You can create actions for various Oracle Fleet Patching and Provisioning operations, such as import image, add and delete working copy, and add, delete, move, and upgrade a software home.

You can define different actions for each operation, which can be further differentiated by the type of image to which the operation applies. User-defined actions can be run before or after a given operation, and are run on the deployment on which the operation is run, whether it be an Oracle Fleet Patching and Provisioning Server, an Oracle Fleet Patching and Provisioning Client (12*c* release 2 (12.2), or later), or an `rhpclient`-less target.

User-defined actions are shell scripts which are stored on the Oracle Fleet Patching and Provisioning Server. When a script runs, it is given relevant information about the operation on the command line. Also, you can associate a file with the script. The Oracle Fleet Patching and Provisioning Server will copy that file to the same location on the Client or `rhpclient`-less target where the script is run.

For example, perhaps you want to create user-defined actions that are run after a database upgrade, and you want to define different actions for Oracle Database 11*g* and 12*c*. This requires you to define new image types, as in the following example procedure.

1. Create a new image type, (`DB11IMAGE`, for example), based on the ORACLEDBSOFTWARE image type, as follows:

   ```
   $ rhpctl add imagetype -imagetype DB11IMAGE -basetype ORACLEDBSOFTWARE
   ```

   When you add or import an Oracle Database 11*g* gold image, you specify the image type as `DB11IMAGE`.

2. Define a user action and associate it with the `DB11IMAGE` image type and the upgrade operation. You can have different actions that are run before or after upgrade.
3. To define an action for Oracle Database 12*c*, create a new image type (`DB12IMAGE`, for example) that is based on the ORACLEDBSOFTWARE image type, as in the preceding step, but with the `DB12IMAGE` image type.

> **Note:**
>
> If you define user actions for the base type of a user-defined image type (in this case the base type is ORACLEDBSOFTWARE), then Oracle Fleet Patching and Provisioning performs those actions before the actions for the user-defined image type.

You can modify the image type of an image using the `rhpctl modify image` command. Additionally, you can modify, add, and delete other actions. The following two tables, Table 8-1 and Table 8-2, list the operations you can customize and the parameters you can use to define those operations, respectively.

**Table 8-1 Oracle Fleet Patching and Provisioning User-Defined Operations**

| Operation | Parameter List |
|---|---|
| IMPORT_IMAGE | RHP_OPTYPE, RHP_PHASE, RHP_PATH, RHP_PATHOWNER, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ADD_WORKINGCOPY | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_PATH, RHP_STORAGETYPE, RHP_USER, RHP_NODES, RHP_ORACLEBASE, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ADD_DATABASE | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| DELETE_WORKINGCOPY | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOS, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| DELETE_DATABASE | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| MOVE_GIHOME | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOS, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CL, RHP_USERACTIONDATA |

**Table 8-1 (Cont.) Oracle Fleet Patching and Provisioning User-Defined Operations**

| Operation | Parameter List |
|---|---|
| MOVE_DATABASE<br>This user action is run for each database involved in a patching operation.<br>If the run scope is set to ALLNODES, then the script is run for each database on every cluster node.<br>If the run scope is set to ONENODE, then the script is run for each database on the node on which the patch was applied to the database. | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_DATAPATCH, RHP_USERACTIONDATA |
| UPGRADE_GIHOME | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| UPGRADE_DATABASE | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ADDNODE_DATABASE | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| DELETENODE_DATABASE | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_DBNAME, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ADDNODE_GIHOME | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| DELETENODE_GIHOME | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ADDNODE_WORKINGCOPY | RHP_OPTYPE, RHP_PHASE, RHP_WORKINGCOPY, RHP_CLIENT, RHP_PATH, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_IMAGE, RHP_IMAGETYPE, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ZDTUPGRADE_DATABASE | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |

**Table 8-1 (Cont.) Oracle Fleet Patching and Provisioning User-Defined Operations**

| Operation | Parameter List |
|---|---|
| ZDTUPGRADE_DATABASE_SNAPDB | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ZDTUPGRADE_DATABASE_DBUA | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |
| ZDTUPGRADE_DATABASE_SWITCHBACK | RHP_OPTYPE, RHP_PHASE, RHP_SOURCEWC, RHP_SOURCEPATH, RHP_DESTINATIONWC, RHP_DESTINATIONPATH, RHP_SRCGGWC, RHP_SRCGGPATH, RHP_DSTGGWC, RHP_DSTGGPATH, RHP_DBNAME, RHP_IMAGE, RHP_IMAGETYPE, RHP_PROGRESSLISTENERHOST, RHP_PROGRESSLISTENERPORT, RHP_VERSION, RHP_CLI, RHP_USERACTIONDATA |

**Table 8-2 User-Defined Operations Parameters**

| Parameter | Description |
|---|---|
| RHP_OPTYPE | The operation type for which the user action is being processed, as listed in the previous table. |
| RHP_PHASE | This parameter indicates whether the user action is processed before or after the operation (is either PRE or POST). |
| RHP_SOURCEWC | The source working copy name for a patch of upgrade operation. |
| RHP_SOURCEPATH | The path of the source working copy home. |
| RHP_DESTINATIONWC | The destination working copy name for a patch or upgrade operation. |
| RHP_DESTINATIONPATH | The path of the destination working copy home. |
| RHP_SRCGGWC | The name of the version of the Oracle GoldenGate working copy from which you want to upgrade. |
| RHP_SRCGGPATH | The absolute path of the version of the Oracle GoldenGate software home from which you want to upgrade. |
| RHP_DESTGGWC | The name of the version of the Oracle GoldenGate working copy to which you want to upgrade. |
| RHP_DESTGGPATH | The absolute path of the version of the Oracle GoldenGate software home to which you want to upgrade. |
| RHP_PATH | This is the path to the location of the software home. This parameter represents the path on the local node from where the RHPCTL command is being run for an IMPORT_IMAGE operation. For all other operations, this path is present on the site where the operation is taking place. |
| RHP_PATHOWNER | The owner of the path for the gold image that is being imported. |

**Table 8-2 (Cont.) User-Defined Operations Parameters**

| Parameter | Description |
|---|---|
| RHP_PROGRESSLISTENERHOST | The host on which the progress listener is listening. You can use this parameter, together with a progress listener port, to create a TCP connection to print output to the console on which the RHPCTL command is being run. |
| RHP_PROGRESSLISTENERPORT | The port on which the progress listener host is listening. You can use this parameter, together with a progress listener host name, to create a TCP connection to print output to the console on which the RHPCTL command is being run. |
| RHP_IMAGE | The image associated with the operation. In the case of a move operation, it will reflect the name of the destination image. |
| RHP_IMAGETYPE | The image type of the image associated with the operation. In the case of a move operation, it will reflect the name of the destination image. |
| RHP_VERSION | The version of the Oracle Grid Infrastructure software running on the Oracle Fleet Patching and Provisioning Server. |
| RHP_CLI | The exact command that was run to invoke the operation. |
| RHP_STORAGETYPE | The type of storage for the home (either LOCAL or RHP_MANAGED) |
| RHP_USER | The user for whom the operation is being performed. |
| RHP_NODES | The nodes on which a database will be created. |
| RHP_ORACLEBASE | The Oracle base location for the provisioned home. |
| RHP_DBNAME | The name of the database to be created. |
| RHP_CLIENT | The name of the client cluster. |
| RHP_DATAPATCH | This parameter is set to TRUE at the conclusion of the user action on the node where the SQL patch will be run after the move database operation is complete. |
| RHP_USERACTIONDATA | This parameter is present in all of the operations and is used to pass user-defined items to the user action as an argument during runtime. |

### Example of User-Defined Action

Suppose there is an image type, APACHESW, to use for provisioning and managing Apache deployments. Suppose, too, that there is a Gold Image of Apache named apacheinstall. The following example shows how to create a user action that will run prior to provisioning any copy of our Apache Gold Image.

The following is a sample user action script named addapache_useraction.sh:

```
$ cat /scratch/apacheadmin/addapache_useraction.sh
#!/bin/sh

#refer to arguments using argument names
touch /tmp/SAMPLEOUT.txt;
for i in "$@"
do
    export $i
done
```

```
echo "OPTYPE = $RHP_OPTYPE" >> /tmp/SAMPLEOUT.txt;
echo "PHASE = $RHP_PHASE" >> /tmp/SAMPLEOUT.txt;
echo "WORKINGCOPY = $RHP_WORKINGCOPY" >> /tmp/SAMPLEOUT.txt;
echo "PATH = $RHP_PATH" >> /tmp/SAMPLEOUT.txt;
echo "STORAGETYPE = $RHP_STORAGETYPE" >> /tmp/SAMPLEOUT.txt;
echo "USER = $RHP_USER" >> /tmp/SAMPLEOUT.txt;
echo "NODES = $RHP_NODES" >> /tmp/SAMPLEOUT.txt;
echo "ORACLEBASE = $RHP_ORACLEBASE" >> /tmp/SAMPLEOUT.txt;
echo "DBNAME = $RHP_DBNAME" >> /tmp/SAMPLEOUT.txt;
echo "PROGRESSLISTENERHOST = $RHP_PROGRESSLISTENERHOST" >> /tmp/SAMPLEOUT.txt;
echo "PROGRESSLISTENERPORT = $RHP_PROGRESSLISTENERPORT" >> /tmp/SAMPLEOUT.txt;
echo "IMAGE = $RHP_IMAGE" >> /tmp/SAMPLEOUT.txt;
echo "IMAGETYPE = $RHP_IMAGETYPE" >> /tmp/SAMPLEOUT.txt;
echo "RHPVERSION = $RHP_VERSION" >> /tmp/SAMPLEOUT.txt;
echo "CLI = $RHP_CLI" >> /tmp/SAMPLEOUT.txt;
echo "USERACTIONDATA = $RHP_USERACTIONDATA" >> /tmp/SAMPLEOUT.txt;
$
```

The script is registered to run at the start of `rhpctl add workingcopy` commands. The add working copy operation stops if the script fails.

The following command creates a user action called addapachepre:

```
$ rhpctl add useraction -optype ADD_WORKINGCOPY -pre -onerror ABORT -
useraction
  addapachepre -actionscript /scratch/apacheadmin/addapache_useraction.sh
  -runscope ONENODE
```

The following command registers the user action for the `APACHESW` image type:

```
$ rhpctl modify imagetype -imagetype APACHESW -useractions addapachepre
```

The registered user action is invoked automatically at the start of commands that deploy a working copy of any image of the APACHESW type, such as the following:

```
$ rhpctl add workingcopy -workingcopy apachecopy001 -image apacheinstall
  -path /scratch/apacheadmin/apacheinstallloc -sudouser apacheadmin -sudopath
  /usr/local/bin/sudo -node targetnode003 -user apacheadmin -useractiondata
"sample"
```

The sample script creates the `/tmp/SAMPLEOUT.txt` output file. Based on the example command, the output file contains:

```
$ cat /tmp/SAMPLEOUT.txt
OPTYPE = ADD_WORKINGCOPY
PHASE = PRE
WORKINGCOPY = apachecopy001
PATH = /scratch/apacheadmin/apacheinstallloc
STORAGETYPE =
USER = apacheadmin
NODES = targetnode003
ORACLEBASE =
DBNAME =
PROGRESSLISTENERHOST = mds11042003.my.example.com
```

```
PROGRESSLISTENERPORT = 58068
IMAGE = apacheinstall
IMAGETYPE = APACHESW
RHPVERSION = 12.2.0.1.0
CLI = rhpctl__add__workingcopy__-image__apacheinstall__-path__/scratch/
apacheadmin
 /apacheinstallloc__-node__targetnode003__-useractiondata__sample__
 -sudopath__/usr/local/bin/sudo__-workingcopy__apachecopy__-
user__apacheadmin__
 -sudouser__apacheadmin__USERACTIONDATA = sample
$
```

**Notes:**

- In the preceding output example empty values terminate with an equals sign (=).
- The spaces in the command-line value of the `RHP_CLI` parameter are replaced by two underscore characters (__) to differentiate this from other parameters.

## Oracle Fleet Patching and Provisioning Notifications

The Oracle Fleet Patching and Provisioning Server is the central repository for the software homes available to the data center. Therefore, it is essential for administrators throughout the data center to be aware of changes to the inventory that may impact their areas of responsibility.

You can create subscriptions to image series events. Oracle Fleet Patching and Provisioning notifies a subscribed role or number of users by email of any changes to the images available in the series, including addition or removal of an image. Each series may have a unique group of subscribers.

Also, when a working copy of a gold image is added to or deleted from an `rhpclient`-less target, the owner of the working copy and any additional users can be notified by email. If you want to enable notifications for additional Oracle Fleet Patching and Provisioning events, you can create a user-defined action as described in the next section.

## Job Scheduler for Operations

The Oracle Fleet Patching and Provisioning job scheduler provides you with a mechanism to submit operations at a scheduled time instead of running the command immediately, querying the metadata of the job, and then deleting the job from the repository.

**Note:**

Starting with Oracle Grid Infrastructure 23ai, you can pause and resume scheduled jobs. This feature also enables you to perform controlled batch processing.

The Oracle Fleet Patching and Provisioning job scheduler includes the following features:

- Enables you to schedule a command to run at a specific point in time by providing the time value

- Performs the job and stores the metadata for the job, along with the current status of the job
- Pauses jobs to postpone processing of the job until you validate pre-requisites for the job and resume the job.
- Stores the logs for each of the jobs that have run or are running
- Enables you to query job details ( for all jobs or for specific jobs, based on the user roles)
- Deletes jobs
- Authorizes the running, querying, and deleting of jobs, based on role-based access for users

Use the `-schedule timer_value` or `-schedule PAUSE` command parameter with any of the following RHPCTL commands to schedule certain Oracle Fleet Patching and Provisioning operations:

- `rhpctl add workingcopy`
- `rhpctl import image`
- `rhpctl delete image`
- `rhpctl add database`
- `rhpctl move gihome`
- `rhpctl upgrade gihome`
- `rhpctl move database`
- `rhpctl upgrade database`
- `rhpctl addnode database`
- `rhpctl deletenode database`
- `rhpctl delete workingcopy`

For example:

```
$ rhpctl add workingcopy -workingcopy 23_3 -image 23_3_Base -
oraclebase /u01/app/oracle -schedule 2016-12-21T19:13:17+05
```

All commands are run in reference with the time zone of the server, according to the ISO-8601 value, and RHPCTL displays the command result by specifying the same time zone.

**Command Results**

RHPCTL stores any command that is run from the command queue on the Oracle Fleet Patching and Provisioning Server. When you query a command result by specifying the command identifier, then RHPCTL returns the path to the job output file, along with the results.

**Job Operation**

When you run an RHPCTL command with the `-schedule` parameter, the operation creates a job with a unique job ID that you can query to obtain the status of the job or to resume the job.

**Job Status**

At any point in time, a job can be in any of the following states:

```
EXECUTED | TIMER_RUNNING | EXECUTING | UNKNOWN | TERMINATED | PAUSED
```

- `EXECUTED`: The job is complete.
- `TIMER_RUNNING`: The timer for the job is still running.
- `EXECUTING`: The timer for the job has expired and the job is running.
- `UNKNOWN`: There is an unexpected failure due to issues such as a server going down, nodes going down, or any resource failures.
- `TERMINATED`: There is an abrupt failure or the operation has stopped.
- `PAUSED`: The job remains paused until you resume it using the `rhpctl resume job` command.

**Related Topics**

- rhpctl delete job
  Deletes a specific scheduled job from the repository.
- rhpctl query job
  Queries the current status of a scheduled job with a specific job ID.

# Patching Oracle Grid Infrastructure and Oracle Database Using Batches

You can sequentially process batches of nodes, with a number of nodes in each batch being restarted in parallel.

Using batches maximizes service availability during the patching process. When you patch Oracle Grid Infrastructure 12*c* release 2 (12.2.x) or later software homes, you can define the batches on the command line or choose to have Fleet Patching and Provisioning generate the list of batches based on its analysis of the database services running in the cluster.

There are several methods for defining batches:

- User-Defined Batches
- Fleet Patching and Provisioning-Defined Batches
- Patching Oracle Database Using Batches

### User-Defined Batches

When you use this method of patching, the first time you run the `rhpctl move gihome` command, you must specify the source home, the destination home, the batches, and other options, as needed. The command terminates after the first node restarts.

To patch Oracle Grid Infrastructure using batches that you define:

1. Define a list of batches on the command line and begin the patching process, as in the following example:

   ```
   $ rhpctl move gihome -sourcewc wc1 -destwc wc2 -batches "(n1),(n2,n3),(n4)"
   ```

   The preceding command example initiates the move operation, and terminates and reports successful when the Oracle Grid Infrastructure stack restarts in the first batch. Oracle Grid Infrastructure restarts the batches in the order you specified in the `-batches` parameter.

   If your batches do not include all the nodes in the cluster, then Oracle FPP automatically adds the excluded nodes as a new batch group at the end of the list of batches. For example, if your cluster has four nodes n1, n2, n3, n4, and you create two batches as "(n1),(n2)", then Oracle FPP automatically adds a third batch group at the end as "(n1),(n2),(n3,n4)".

In the command example, node `n1` forms the first batch, nodes `n2` and `n3` form the second batch, and node `n4` forms the last batch. The command defines the source working copy as `wc1` and the patched (destination) working copy as `wc2`.

> **Notes:**
>
> You can specify batches such that singleton services are relocated between batches and non-singleton services remain partially available during the patching process.

2. You must process the next batch by running the `rhpctl move gihome` command, again, as follows:

```
$ rhpctl move gihome -destwc wc2 -continue
```

The preceding command example restarts the Oracle Grid Infrastructure stack on the second batch (nodes `n2` and `n3`). The command terminates by reporting that the second batch was successfully patched.

3. Repeat the previous step until you have processed the last batch of nodes. If you attempt to run the command with the `-continue` parameter after the last batch has been processed, then the command returns an error.

   If the `rhpctl move gihome` command fails at any time during the above sequence, then, after determining and fixing the cause of the failure, rerun the command with the `-continue` option to attempt to patch the failed batch. If you want to skip the failed batch and continue with the next batch, use the `-continue` and `-skip` parameters. If you attempt to skip over the last batch, then the move operation is terminated.

   Alternatively, you can reissue the command using the `-revert` parameter to undo the changes that have been made and return the configuration to its initial state.

   You can use the `-abort` parameter instead of the `-continue` parameter at any point in the preceding procedure to terminate the patching process and leave the cluster in its current state.

**Notes:**

- Administrator-managed services with one preferred instance and no available instances cannot be relocated and will go OFFLINE while instances are being restarted.
- If a move operation is in progress, then you cannot initiate another move operation from the same source home or to the same destination home.
- After the move operation has ended, services may be running on nodes different from the ones they were running on before the move and you will have to manually relocate them back to the original instances, if necessary.
- If you use the `-abort` parameter to terminate the patching operation, then Fleet Patching and Provisioning does not clean up or undo any of the patching steps. The cluster, databases, or both may be in an inconsistent state because all nodes are not patched.
- Depending on the start dependencies, services that were offline before the move began could come online during the move.

**Fleet Patching and Provisioning-Defined Batches**

Using Fleet Patching and Provisioning to define and patch batches of nodes means that you need only run one command, as shown in the following command example, where the source working is `wc1` and the destination working copy is `wc2`:

```
$ rhpctl move gihome -sourcewc wc1 -destwc wc2 -smartmove -saf 50 [-eval]
```

If the move operation fails at some point before completing, then you can either rerun the operation by running the command again, or you can undo the partially completed operation, as follows:

```
$ rhpctl move gihome -destwc destination_workingcopy_name -revert
[authentication_option]
```

You can use the `-revert` parameter with an un-managed home.

The parameters used in the preceding example are as follows:

- `-smartmove`: This parameter restarts the Oracle Grid Infrastructure stack on disjoint sets of nodes so that singleton resources are relocated before Oracle Grid Infrastructure starts.

  The `-smartmove` parameter:

  - Creates a map of services and nodes on which they are running.
  - Creates batches of nodes. A node can be merged into a batch if:
    - The availability of any non-singleton service, running on this node, does not go below the specified service availability factor (or the default of 50%).
    - There is a singleton service running on this node and the batch does not contain any of the relocation destination nodes for the service.
  - Restarts the Oracle Grid Infrastructure stack batch by batch.

- Service availability factor (`-saf Z+`): You can specify a positive number, as a percentage, that will indicate the minimum number of database instances on which a database service must be running. For example:
  - If you specify `-saf 50` for a service running on two instances, then only one instance can go offline at a time.
  - If you specify `-saf 50` for a service running on three instances, then only one instance can go offline at a time.
  - If you specify `-saf 75` for a service running on two instances, then an error occurs because the destination can never be met.
  - The service availability factor is applicable for services running on at least two instances. As such, the service availability factor can be 0% to indicate a non-rolling move, but not 100%. The default is 50%.
  - If you specify a service availability factor for singleton services, then the parameter will be ignored because the availability of such services is 100% and the services will be relocated.
- `-eval`: You can optionally use this parameter to view the auto-generated batches. This parameter also shows the sequence of the move operation without actually patching the software.

**Patching Oracle Database Using Batches**

During database patching, Fleet Patching and Provisioning can sequentially process batches of nodes, with a number of nodes in each batch being restarted in parallel. This method maximizes service availability during the patching process. You can define the batches on the command line or choose to have Fleet Patching and Provisioning generate the list of batches based on its analysis of the database services running in the cluster.

**Related Topics**

- rhpctl move gihome
  Moves the Oracle Grid Infrastructure software stack from one home to another.

# Combined Oracle Exadata Database Server and Grid Infrastructure Update

With combined Oracle Exadata database server and Oracle Grid Infrastructure update you can utilize the functionality of multiple independent capabilities.

Patching Grid Infrastructure and updating the Exadata database nodes both require a shutdown and startup of every database instance on that node. This can take considerable time, depending on the number of applications running and the time it takes to shutdown instances and start them up. Performing both of these patching actions independently doubles the downtime on production databases. Using the combined patching feature of Oracle FPP automates both of these patching actions into a single integrated patching process that requires only one sequence of shutdown and startup of database instances on each node. The combined patching on multiple nodes in batches further brings down the overall patching window.

Oracle FPP internally uses the `patchmgr` tool to patch Exadata database nodes. The combined patching method uses an integrated flow of the inherent Oracle FPP implementation for Oracle Grid Infrastructure patching and then invokes the `patchmgr` tool to patch each Exadata database node.

To complete combined Oracle Exadata database node and Oracle Grid Infrastructure patching, you must perform the operations discussed in the following:

- Creating the Oracle Exadata Image
- Deploying the Oracle Exadata Update Image
- Combined Oracle Grid Infrastructure Move and Database Node Update

> **Note:**
>
> Creating and deploying an Oracle Exadata image does not require any downtime and you can perform both these operations before patching Oracle Grid Infrastructure and Oracle Exadata database. You need to create an Oracle Exadata image on the Oracle FPP server only once in a patching cycle, however, you need to deploy Oracle Exadata image, and patch Oracle Grid Infrastructure and Oracle Exadata database node on each server.

### Creating the Oracle Exadata Image

Use the `rhpctl import image` command to create the Oracle Exadata update image by copying the entire software contents from the specified path to the Oracle Fleet Patching and Provisioning Server (FPPS).

**Example 8-1 Creating an Oracle Exadata Update Image**

The following command imports an Oracle Exadata image to the Oracle FPP Server as an image. In the example, `image` specifies the name of the Oracle Exadata image that you want to add, `path` specifies the absolute path location of the Oracle Exadata software home that you want to import, `imagetype` specifies `EXAPATCHSOFTWARE` for Oracle Exadata software, and `version` specifies the version of the Oracle Exadata software.

```
$ rhpctl import image -image EXADATAIMAGEV1
   -path /tmp/ExadataPatchBundle -imagetype EXAPATCHSOFTWARE -version
19.2.2.0.0.190513.2
```

When you import an Oracle Exadata software home with this command, the `version` parameter must be the version of the Oracle Exadata software required by the `patchmgr` on the database node. The `path` parameter should contain Oracle Exadata update zip files.

> **See Also:**
>
> rhpctl import image for the complete syntax of the `rhpctl import image` command

### Deploying the Oracle Exadata Update Image

Use the `rhpctl deploy image` command to propagate the Oracle Exadata update image to server.

**Example 8-2 Deploying an Oracle Exadata image**

The following command deploys an Oracle Exadata image to a client cluster. In the example, `image` specifies the name of the Oracle Exadata image that you want to deploy, `client` specifies the name of the cluster to which you want to deploy the image, and `path` specifies the

absolute path location for deploying the Oracle Exadata software home on the `rhpclient`-less target or client side.

```
$ rhpctl deploy image -image EXADATAIMAGEV1 -client CLUSTER1 -path /
exadatasoftware
```

The `targetnode` parameter is required if the node hosting the home is not a Oracle Fleet Patching and Provisioning Client. If the `rhpclient`-less target or client option is not specified, then the Oracle Exadata image is deployed to the Oracle Fleet Patching and Provisioning Server.

> **See Also:**
>
> rhpctl deploy image for the complete syntax of the `rhpctl deploy image` command.

#### Combined Oracle Grid Infrastructure Move and Database Node Update

Use the `rhpctl move gihome` command to move the Oracle Grid Infrastructure software stack from one home to another while updating the Oracle Exadata database node.

**Example 8-3 Moving an Oracle Grid Infrastructure home and updating a database node**

The following example performs a combined Oracle Grid Infrastructure move and database node update on client cluster. In the example, `sourcewc` specifies the name of the source working copy, `destwc` specifies the name of the destination working copy to which you want to move the Oracle Grid Infrastructure home, `image` specifies the name of the Oracle Exadata image, `batches` specifies a comma-delimited list of batches of nodes where each batch is a comma-delimited list of node names enclosed in parentheses and node names are enclosed in double quotation marks ("") in the format: `"(nA,nB,...),(...,nY,nZ)"`, `iso_repo` specifies the image in the ISO repository, and `pathmgrloc` specifies the patch manager location.

```
$ rhpctl move gihome -sourcewc prodHomeV1 -destwc prodHomeV2 -image
EXADATAIMAGEV1
    -batches "(rac07box1,rac07box2,rac07box3),(rac07box4)"
   -patchmgrargs "-ignore_alerts" -iso_repo p28802055_192000_Linux-x86-64.zip
-client prodcluster
    -patchmgrloc /patchMgr/dbserver_patch_19.190306
```

With each invocation of the `rhpctl move gihome` command, FPP patches the database node first and then patches Oracle Grid Infrastructure. This is the processing order for each node in the specified batch.

If the first batch includes more than one database node, then FPP invokes `patchmgr` in parallel for all the nodes. As soon as a node completes the `patchmgr` operation, including the post `patchmgr` operations, FPP starts the Oracle Grid Infrastructure patching on that node. When the Oracle Grid Infrastructure patching completes on this node, FPP then begins patching with Oracle Grid Infrastructure patching on the other nodes when the database node patching completes on those nodes.

If rebooting a node is delayed because of a `patchmgr` failure or a `patchmgr` operation timeout, the `rhpctl move gihome` command can be resumed after the node is back up.

> **See Also:**
>
> rhpctl move gihome for the complete syntax of the `rhpctl move gihome` command

## Gold Image Distribution Among Oracle Fleet Patching and Provisioning Servers

Oracle Fleet Patching and Provisioning can automatically share and synchronize gold images between Oracle Fleet Patching and Provisioning Servers.

In the Oracle Fleet Patching and Provisioning architecture, one Oracle Fleet Patching and Provisioning Server manages a set of Oracle Fleet Patching and Provisioning Clients and `rhpclient`-less targets within a given data center or network segment of a data center. If you have more than one data center or a segmented data center, you must have more than one Oracle Fleet Patching and Provisioning Server.

In the Oracle Fleet Patching and Provisioning architecture, one Oracle Fleet Patching and Provisioning Server manages a set of Oracle Fleet Patching and Provisioning Clients and `rhpclient`-less targets within a given data center or network segment of a data center. If you have more than one data center or a segmented data center, then you must have more than one Oracle Fleet Patching and Provisioning Server to facilitate large-scale standardization across multiple estates.

Oracle Fleet Patching and Provisioning Servers retain the ability to create and manage gold images private to their scope, so local customizations are seamlessly supported.

You must first establish a peer relationship between two Oracle Fleet Patching and Provisioning Servers. Registration uses the names of the Oracle Fleet Patching and Provisioning Server clusters. The names of the two clusters can be the same but there is one naming restriction: an Oracle Fleet Patching and Provisioning Server, such as `FPPS_1`, cannot register a peer Oracle Fleet Patching and Provisioning Server if that peer has the same name as an Oracle Fleet Patching and Provisioning Client or `rhpclient`-less target within the management domain of `FPPS_1`.

The following steps show how you can establish a peer relationship between two Oracle Fleet Patching and Provisioning Servers. Note that super user or root credentials are not required in this process.

1. On the first Oracle Fleet Patching and Provisioning Server (`FPPS_1`), create a file containing the server configuration information.

   ```
   $ rhpctl export server -serverdata file_path
   ```

2. Copy the server configuration file created on `FPPS_1` to a second Oracle Fleet Patching and Provisioning Server (`FPPS_2`).
3. On the second Oracle Fleet Patching and Provisioning Server (`FPPS_2`), complete the registration of `FPPS_2`.

   ```
   $ rhpctl register server -server FPPS_1_cluster_name
       -serverdata server_cfg_file_copied_from_FPPS_1
   ```

4. On FPPS_2, create a file containing the server configuration information.

```
$ rhpctl export server -serverdata file_path
```

5. Copy the server configuration file created on FPPS_2 to FPPS_1.
6. On the first Oracle Fleet Patching and Provisioning Server (FPPS_1), complete the registration ofFPPS_1.

```
$ rhpctl register server -server FPPS_2_cluster_name
    -serverdata server_cfg_file_copied_from_FPPS_2
```

After you register an Oracle Fleet Patching and Provisioning Server as a peer, the following command displays the peer (or peers) of the server:

```
$ rhpctl query peerserver
```

You can inspect the images on a peer Oracle Fleet Patching and Provisioning Server, as follows:

```
$ rhpctl query image -server server_cluster_name
```

The preceding command displays all images on a specific peer Oracle Fleet Patching and Provisioning Server. Additionally, you can specify a peer server along with the -image *image_name* parameter to display details of a specific image on a specific peer server.

An Oracle Fleet Patching and Provisioning Server can have multiple peers. Oracle does not support chained relationships between peers, however, such as, if FPPS_1 is a peer of FPPS_2, and FPPS_2 is also a peer of FPPS_3, then no relationship is established or implied between FPPS_1 and FPPS_3, although you can make them peers if you want.

Retrieve a copy or copies of gold images from a peer Oracle Fleet Patching and Provisioning Server, as follows:

```
$ rhpctl instantiate image -server server_cluster_name
```

Running the rhpctl instantiate image command activates an auto-update mechanism. From that point on, when you create gold images on a peer Oracle Fleet Patching and Provisioning Server, such as FPPS_2, they are candidates for being automatically copied to the Oracle Fleet Patching and Provisioning Server that performed the instantiate operation, such as FPPS_1. Whether a new gold image is automatically copied depends on that relevance of the image to any instantiate parameters that you may include in the command:

- -all: Creates an automatic push for all gold images created on FPPS_2 to FPPS_1
- -image *image_name*: Creates an automatic push for all new descendant gold images of the named image created on FPPS_2 to FPPS_1. A descendant of the named image is an image that is created on FPPS_2 using the rhpctl add image command.
- -series *series_name*: Creates an automatic push for all gold images added to the named series on FPPS_2 to FPPS_1
- -imagetype *image_type*: Creates an automatic push for all gold images created of the named image type on FPPS_2 to FPPS_1

To stop receiving updates that were established by the `rhpctl instantiate image` command, run `rhpctl uninstantiate image` and specify the peer Oracle Fleet Patching and Provisioning Server and one of the following: all, image name, image series name, or image type.

End the peer relationship, as follows, on any one of the Oracle Fleet Patching and Provisioning Servers:

```
$ rhpctl unregister server -server server_cluster_name
```

**Related Topics**

- rhpctl export server
- rhpctl register server
- rhpctl query peerserver
- rhpctl query image
  Displays the configuration of an existing image.
- rhpctl instantiate image
- rhpctl uninstantiate image
- rhpctl unregister server

# Error Prevention and Automated Recovery Options

Fleet Patching and Provisioning has error prevention and automated recovery options to assist you during maintenance operations.

During maintenance operations, errors must be avoided whenever possible and, when they occur, you must have automated recovery paths to avoid service disruption.

### Error Prevention

Many RHPCTL commands include the `-eval` parameter, which you can use to run the command and evaluate the current configuration without making any changes to determine if the command can be successfully run and how running the command will impact the configuration. Commands that you run using the `-eval` parameter run as many prerequisite checks as possible without changing the configuration. If errors are encountered, then RHPCTL reports them in the command output. After you correct any errors, you can run the command again using `-eval` to validate the corrections. Running the command successfully using `-eval` provides a high degree of confidence that running the actual command will succeed.

You can test commands with the `-eval` parameter outside of any maintenance window, so the full window is available for the maintenance procedure, itself.

### Automated Recovery Options

During maintenance operations, errors can occur either in-flight (for example, partway through either an `rhpctl move database` or `rhpctl move gihome` command) or after a successful operation (for example, after an `rhpctl move database` command, you encounter performance or behavior issues).

### In-Flight Errors

Should in-flight errors occur during move operations:

- Correct any errors that RHPCTL reports and rerun the command, which will resume running at the point of failure.

  If rerunning the command succeeds and the move operation has a post-operation user action associated with it, then the user action is run. If there is a pre-operation user action, however, then RHPCTL does not rerun the command.

- Run a new move command, specifying only the destination from the failed move (working copy or unmanaged home), an authentication option, if required, and use the `-revert` parameter. This will restore the configuration to its initial state.

  No user actions associated with the operation are run.

- Run a new move command, specifying only the destination from the failed move (working copy or unmanaged home), an authentication option if required, and the `-abort` parameter. This leaves the configuration in its current state. Manual intervention is required at this point to place the configuration in a final state.

  No user actions associated with the operation are run.

**Post-Update Issues**

Even after a successful move operation to a new database or Oracle Grid Infrastructure home, you still may need to undo the change and roll back to the prior home. You can do this by rerunning the command with the source and destination homes reversed. This is, effectively, a fresh move operation performed without reference to the previous move operation.

> **Note:**
>
> For the independent automatons, the source and destination homes are always unmanaged homes (those homes not provisioned by Fleet Patching and Provisioning). When the move operation is run on a Fleet Patching and Provisioning Server or Fleet Patching and Provisioning Client, the destination home must be a managed home that was provisioned by Fleet Patching and Provisioning.

# Fleet Patching and Provisioning Logs and Trace Files

Use Oracle Fleet Patching and Provisioning logs and traces to obtain more information for identifying and debugging Oracle FPP Server and client errors.

The following log and trace files are generated during Oracle FPP Server and client operations. These are the key log and trace files of interest for diagnostic purposes:

- `$ORACLE_BASE/crsdata/$HOSTNAME/rhp/rhpserver.log.{n}`
  Contains a detailed log of the actions that occur for the Oracle FPP Server operations. The log file numbers are updated automatically, and `.0` is always the most recent log file.
- `$ORACLE_BASE/crsdata/$HOSTNAME/rhp/srvmhelper_clsn_{unixtimestamp}.log.0`
  Contains a detailed log of the actions that occur during the Oracle FPP helper operations.
- `$ORACLE_BASE/crsdata/$HOSTNAME/rhp/logs/catalina.out`
  Contains a detailed log of the actions that occur during the Java Application Server operations.
- `/u01/app/grid/diag/crs/fpps/crs/trace`
  Contains a detailed log of the actions that occur during the Oracle Grid Infrastructure operations. Oracle Fleet Patching and Provisioning is part of the Oracle Grid Infrastructure

stack and thus Cluster Ready Services daemon (CRSD) log and trace files can be useful to debug stop and start errors of the Oracle FPP Server or client.

- `$ORACLE_BASE/cfgtoollogs/dbca`
  Contains a detailed log of the actions that occur during the Oracle Database deployment.
- `$ORACLE_BASE/cfgtoollogs/dbua`
  Contains a detailed log of the actions that occur during the Oracle Database upgrade.

You can also use the Trace File Analyzer (TFA) to collect Oracle FPP logs and traces by using the `-rhp` flag. For example:

```
$ tfactl diagcollect -rhp
```

9

# Oracle Fleet Patching and Provisioning Use Cases

Review these topics for step-by-step procedures to provision, patch, and upgrade your software using Oracle Fleet Patching and Provisioning.

Oracle Fleet Patching and Provisioning is a software lifecycle management solution and helps standardize patching, provisioning, and upgrade of your standard operating environment.

- Creating an Oracle Grid Infrastructure 23ai Deployment
  Provision Oracle Grid Infrastructure software on two nodes that do not currently have a Grid home, and then configure Oracle Grid Infrastructure to form a multi-node Oracle Grid Infrastructure installation.
- Provisioning an Oracle Database Home and Creating a Database
  This procedure provisions Oracle Database 23ai software and creates Oracle Database instances.
- Provisioning a Pluggable Database
  You can provision a Pluggable Database (PDB) on an existing Container Database (CDB) running in a provisioned database working copy.
- Upgrading to Oracle Grid Infrastructure 23ai
  This procedure uses Fleet Patching and Provisioning to upgrade your Oracle Grid Infrastructure cluster from 21c to 23ai.
- Patching Oracle Grid Infrastructure and Oracle Databases Simultaneously
  This procedure patches Oracle Grid Infrastructure and Oracle Databases on the cluster to the latest patch level without cluster downtime.
- Patching Oracle Data Guard Database Automatically
  Starting with Oracle Grid Infrastructure 23ai, Oracle FPP enables you to automatically patch Oracle Data Guard standby databases before patching their primary databases.
- Patching Oracle Database 23ai Without Downtime
  This procedure explains how to patch Oracle Database 23ai with the latest patching without bringing down the database.
- Upgrading to Oracle Database 23ai
  This procedure describes how to upgrade an Oracle database from Oracle Database 21c to 23ai with a single command, using Fleet Patching and Provisioning, both for managed and unmanaged Oracle homes.
- Performing Oracle Exadata Database Node Live Update
  Oracle Fleet Patching and Provisioning (Oracle FPP) supports Exadata Live Update, which was introduced in Oracle Exadata System Software release 24.1.0.
- Provisioning an Oracle Database Using Zip Copy
  Starting with Oracle Grid Infrastructure 19c Release Update (19.11), Oracle FPP allows you to install the gold images without transferring them to the destination host.
- Adding a Node to a Cluster and Scaling an Oracle RAC Database to the Node
  You can add a node to your two-node cluster by using Fleet Patching and Provisioning to add the node, and then extend an Oracle RAC database to the new node.

- Adding Gold Images for Fleet Patching and Provisioning
  Create gold images of software home and store them on the Fleet Patching and Provisioning Server, to use later to provision Oracle homes.
- User Actions for Common Fleet Patching and Provisioning Tasks
  You can use Fleet Patching and Provisioning user actions to perform many tasks, such as installing and configuring any type of software and running scripts.

# Creating an Oracle Grid Infrastructure 23ai Deployment

Provision Oracle Grid Infrastructure software on two nodes that do not currently have a Grid home, and then configure Oracle Grid Infrastructure to form a multi-node Oracle Grid Infrastructure installation.

**Before You Begin**

Provide configuration details for storage, network, users and groups, and node information for installing Oracle Grid Infrastructure in a response file. You can store the response file in any location on the Fleet Patching and Provisioning Server.

You can provision an Oracle Standalone Cluster. Ensure that the response file has the required cluster configuration details.

Ensure that you have storage, network, and operating system requirements configured as stated in the *Oracle Grid Infrastructure Installation Guide*.

**Procedure**

- From the Fleet Patching and Provisioning Server, run the command:

  ```
  $ rhpctl add workingcopy -workingcopy GI23ai -image GI_HOME_23ai -
  responsefile /u01/app/rhpinfo/GI_23ai_install.rsp {authentication_option}
  ```

  `GI23ai` is the working copy based on the image `GI_HOME_23ai`

  `/u01/app/rhpinfo/GI_23ai_install.rsp` is the response file location.

  Configuration Verification Utility (CVU) checks for preinstallation configuration as per requirements. Fleet Patching and Provisioning configures Oracle Grid Infrastructure.

Oracle Grid Infrastructure 23ai is provisioned as per the settings in the same response file.

During provisioning, if an error occurs, the procedure stops and allows you to fix the error. After fixing the error, you can resume the provisioning operation from where it last stopped.

Watch a video Video

# Provisioning an Oracle Database Home and Creating a Database

This procedure provisions Oracle Database 23ai software and creates Oracle Database instances.

**Procedure**

1. From the Fleet Patching and Provisioning Server, provision the Oracle Database home software:

   ```
   $ rhpctl add workingcopy -image db23ai -path /u01/app/dbusr/product/23.0.0/
   db23ai
     -client client_001 -oraclebase /u01/app/dbusr/ -workingcopy db23wc
   ```

   The command provisions the working copy `db23wc` to the specified path on the cluster `client_001`, from the image `db23ai`.

2. Create the database instance:

   ```
   $ rhpctl add database -workingcopy db23wc -dbname db -dbtype RAC
   ```

   The command creates an Oracle RAC database instance `db` . You can use the `add database` command repeatedly to create more instances on the working copy.

Watch a video Video

# Provisioning a Pluggable Database

You can provision a Pluggable Database (PDB) on an existing Container Database (CDB) running in a provisioned database working copy.

After you create a working copy of a gold image, provision that working copy to an Oracle FPP Client or `rhpclient`-less target, and create a database as a multitenant CDB, you can add a PDB to the CDB using the `rhpctl addpdb database` command.

- The following command example creates a PDB called `pdb23ai` on a CDB called `raccdb23ai`, which is on a working copy called `wc_db23ai`:

  ```
  $ rhpctl addpdb database -workingcopy wc_db23ai -cdbname raccdb23ai -
  pdbName pdb23ai
  ```

- Use the `rhpctl deletepdb database` command to delete a PDB from an existing CDB on a working copy.

  The following command example deletes a PDB called `pdb23ai` on a CDB called `raccdb23ai`, which is on a working copy called `wc_db23ai`:

  ```
  $ rhpctl deletepdb database -workingcopy wc_db23ai -cdbname raccdb23ai -
  pdbName pdb23ai
  ```

# Upgrading to Oracle Grid Infrastructure 23ai

This procedure uses Fleet Patching and Provisioning to upgrade your Oracle Grid Infrastructure cluster from 21c to 23ai.

**Before You Begin**

To upgrade to Oracle Grid Infrastructure 23ai, your source must be Oracle Grid Infrastructure 12c release 2 (12.2.0.1), Oracle Grid Infrastructure 18c, Oracle Grid Infrastructure 19c, or Oracle Grid Infrastructure 21c.

Ensure that groups configured in the source home match those in the destination home.

Ensure that you have an image `GI_HOME_23ai` of the Oracle Grid Infrastructure 23ai software to provision your working copy.

`GI_21c` is the active Grid Infrastructure home on the cluster being upgraded. It is a working copy because in this example, Fleet Patching and Provisioning provisioned the cluster. Fleet Patching and Provisioning can also upgrade clusters whose Grid Infrastructure homes are unmanaged that is, homes that Fleet Patching and Provisioning did not provision.

**Procedure**

1. Provision a working copy of the Oracle Grid Infrastructure 23ai software:

   ```
   $ rhpctl add workingcopy -workingcopy GI23ai -image GI_HOME_23ai
   {authentication_option}
   ```

   `GI23ai` is the working copy based on the image `GI_HOME_23ai`.
2. Upgrade your destination cluster to the `GI23ai` working copy:

   ```
   rhpctl upgrade gihome -sourcewc GI21c -destwc GI23ai
   ```

   Oracle Fleet Patching and Provisioning identifies the cluster to upgrade based on the name of the source working copy, and upgrades to the working copy `GI23ai`.

# Patching Oracle Grid Infrastructure and Oracle Databases Simultaneously

This procedure patches Oracle Grid Infrastructure and Oracle Databases on the cluster to the latest patch level without cluster downtime.

**Before You Begin**

In this procedure, Oracle Grid Infrastructure 23ai is running on the destination cluster. Working copy `GI_HOME_23ai_WCPY` is the active Grid home on this cluster. Working copy `DB_HOME_23ai_WCPY` runs an Oracle RAC 23ai Database with running database instance `db1`. Working copy `DB_HOME_23ai_WCPY` runs an Oracle RAC 21c Database with running database instance `db2`

Ensure that you have images `GI_HOME_23ai_PSU1`, `DB_HOME_23ai_PSU1`, `DB_HOME_21c_PSU5` with the required patches for Oracle Grid Infrastructure and Oracle RAC Database on the Fleet Patching and Provisioning Server.

The groups configured in the source home must match with those in the destination home.

**Procedure**

1. Prepare Oracle FPP Client or `rhpclient`-less target Oracle homes as follows:
   a. Provision software-only Grid home on the cluster to be patched:

```
$ rhpctl add workingcopy -workingcopy GI_HOME_23ai_PATCHED_WCPY
  -image GI_HOME_23ai_PSU1 -client CLUSTER_005 -softwareonly
```

   b. Provision each release Database home, without database instances, to be patched:

```
$ rhpctl add workingcopy -workingcopy DB_HOME_23ai_PATCHED_WCPY
  -image DB_HOME_23ai_PSU1
$ rhpctl add workingcopy -workingcopy DB_HOME_21c_PATCHED_WCPY
  -image DB_HOME_21c_PSU5
```

2. Patch Oracle Grid Infrastructure and all Oracle RAC Databases on `node1` as follows:

```
$ rhpctl move gihome -sourcewc GI_HOME_23ai_WCPY -destwc
GI_HOME_23ai_PATCHED_WCPY -auto
  -dbhomes
DB_HOME_21c_WCPY=DB_HOME_21c_PATCHED_WCPY,DB_HOME_23ai_WCPY=DB_HOME_23ai_PA
TCHED_WCPY  -targetnode node1 {authentication_option}
```

When you run the command, you move your active Oracle Grid Infrastructure from working copy `GI_HOME_23ai_WCPY` to `GI_HOME_23ai_PATCHED_WCPY`, Oracle RAC Database `db1` from `DB_HOME_23ai_WCPY` to `DB_HOME_23ai_PATCHED_WCPY`, and Oracle RAC Database `db2` from `DB_HOME_21c_WCPY` to `DB_HOME_21c_PATCHED_WCPY`.

# Patching Oracle Data Guard Database Automatically

Starting with Oracle Grid Infrastructure 23ai, Oracle FPP enables you to automatically patch Oracle Data Guard standby databases before patching their primary databases.

Oracle Fleet Patching and Provisioning (Oracle FPP) automatically discovers the standby databases, associated with primary databases, and patches the standby databases before patching the primary database using a single command.

> **Note:**
>
> If the Oracle Database Guard databases are registered on a different client, which can be in different data center, then those servers must be register peers of each other.

1. Check the Oracle Data Guard configuration of a database to identify the primary database and its standby databases.

```
rhpctl query workingcopy -workingcopy dbwc2360 -details
DataGuard Configuration:
Primary Database : orcl : (Site name)
Standby Database : orcl_stby : stbm000040-vm19 ...
```

2. Schedule a job for each standby database associated with the primary database using a single job ID.

```
rhpctl move database -dbname orcl -sourcewc dbwc2330 -patchedwc dbwc2360 -
patchdataguard YES -schedule NOW
```

   The above command provides a parent job ID, which you can use as a single point of reference to monitor the progress of standby and primary database patching.

3. Monitor the progress of the entire patching operation.

```
rhpctl query job -jobid 73
Wed Jun 26 15:11:07][giusr@grid:bin] $ ./rhpctl query job -jobid 73
server1: Audit ID: 871
Job ID: 73
User: grid
Client: client1
Scheduled job command: "rhpctl move database -dbname orcl -
sourcehome /u01/app/oracle/product/23.0.0/dbhome_1 -patchedwc
dbwc2360 -patchdataguard YES -schedule NOW"
...
...
Job execution start time: 2024-06-26 12:41:43
Job execution end time: 2024-06-26 13:29:18
Job execution elapsed time: 47 minutes 35 seconds
...
...
```

4. Optionally, you can monitor the progress of the individual standby database patching operations.

```
rhpctl query job -jobid 53
```

**Related Topics**

- rhpctl query workingcopy
  Displays the configuration information of an existing working copy.
- rhpctl move database
  Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.
- rhpctl query job
  Queries the current status of a scheduled job with a specific job ID.

# Patching Oracle Database 23ai Without Downtime

This procedure explains how to patch Oracle Database 23ai with the latest patching without bringing down the database.

### Before You Begin

You have an Oracle Database `db23ai` that you want to patch to the latest patch level.

Ensure that the working copy `db23ai_psu` based on the image `DB23ai_PSU` contains the latest patches and is available.

### Procedure

From the Fleet Patching and Provisioning Server, run one of the following commands as per your source and destination database:

1. To patch an Oracle Database home managed by Fleet Patching and Provisioning, and there exist working copies of the source and destination databases, run:

   ```
   rhpctl move database -sourcewc db23ai -patchedwc db23ai_psu
   ```

   `db23ai` is the source working copy of the database being patched

   `db23ai_psu` is the working copy of the Oracle Database software with patches applied, based on the image `DB23ai_PSU`.

2. To patch an unmanaged Oracle Database home (source working copy does not exist because the Oracle home is not managed by Fleet Patching and Provisioning), run:

   ```
   rhpctl move database -sourcehome /u01/app/orabase/product/23.0.0/dbhome_1
    -patchedwc db23ai_psu -targetnode node1
   ```

   `targetnode` specifies the node on which the database to be upgraded is running, because the source Oracle Database is on a 23ai cluster.

   `/u01/app/orabase/product/23.0.0/dbhome_1` is the path of the database being patched

   `db23ai_psu` is the working copy of the Oracle Database software with patches applied, based on the image `DB23ai_PSU`.

   Use the saved gold image for standardized patching of all your databases of release 23ai to the same patch level.

3. If for some reason, you want to rollback the patches applied to a managed Oracle Database home, run:

   ```
   rhpctl move database -sourcewc db23ai_psu
   -patchedwc db23ai -ignorewcpatches
   ```

   `db23ai` is the working copy of the unpatched database to which you want to roll back.

   `db23ai_psu` is the working copy of the Oracle Database software with patches applied, based on the image `DB23ai_PSU`.

For all Oracle Databases, you can also specify these additional options with the `move database` command:

- `-keepplacement`: For admin-managed Oracle RAC Databases (not Oracle RAC One Node Database), Fleet Patching and Provisioning retains the services on the same nodes after the move.
- `-disconnect`: Disconnects all sessions before stopping or relocating services.
- `-drain_timeout`: Specify the time, in seconds, allowed for resource draining to be completed for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. This option is available only with Oracle Database 12*c* release 2 (12.2) or later.
- `-stopoption`: Stops the database.
- `-nodatapatch`: Ensures `datapatch` is not run for databases you are moving.

Watch a video Video

# Upgrading to Oracle Database 23ai

This procedure describes how to upgrade an Oracle database from Oracle Database 21c to 23ai with a single command, using Fleet Patching and Provisioning, both for managed and unmanaged Oracle homes.

**Before you Begin**

- To upgrade to Oracle Database 23ai, your source database must be either Oracle Database 12c release 2 (12.2.0.1), Oracle Database 18c, Oracle Database 19c, or Oracle Database 21c.
- Oracle Grid Infrastructure on which the pre-upgrade database is running must be of the same release or later than the database release to which you are upgrading.
- The source Oracle home to be upgraded can be a managed working copy, that is an Oracle home provisioned using Fleet Patching and Provisioning, or an unmanaged home, that is, an Oracle home not provisioned using Fleet Patching and Provisioning. If you are upgrading an unmanaged Oracle home, provide the complete path of the database for upgrade.

**Procedure to Upgrade Oracle Database using Fleet Patching and Provisioning**

- From the Fleet Patching and Provisioning Server, run one of the following commands as per your source and destination database:

  1. To upgrade an Oracle home managed by Fleet Patching and Provisioning, and there exist working copies of the source and destination databases, run:

     ```
     $ rhpctl upgrade database -dbname test_database -sourcewc db21c -destwc
     db23ai
       {authentication_option}
     ```

     `test_database` is the name of the database being upgraded.

     `db21c` is the source working copy of the pre-upgrade database.

     `db23ai` is the working copy of the upgraded Oracle Database software.

2. To upgrade an unmanaged Oracle home (source working copy does not exist because the Oracle home is not managed by Fleet Patching and Provisioning), run:

```
$ rhpctl move database -sourcehome /u01/app/orabase/product/21.0.0/
dbhome_1
  -destwc db23ai -targetnode node1 {authentication_option}
```

`/u01/app/orabase/product/21.0.0/dbhome_1` is the path of the database being upgraded.

`db23ai` is the working copy of the upgraded Oracle Database software.

`targetnode` specifies the node on which the database to be upgraded is running, because the source Oracle Database is on a 21c cluster.

The upgraded database is now managed by Fleet Patching and Provisioning. You can ensure that your database is patched to the latest level, using Fleet Patching and Provisioning.

> **Note:**
>
> During upgrade, if an error occurs, the procedure stops and allows you to fix the error. After fixing the error, you can resume the upgrade operation from where it last stopped.

Watch a video Video

**Related Topics**

- rhpctl upgrade database
  Upgrades a database to the version of the destination working copy.
- rhpctl move database
  Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.

# Performing Oracle Exadata Database Node Live Update

Oracle Fleet Patching and Provisioning (Oracle FPP) supports Exadata Live Update, which was introduced in Oracle Exadata System Software release 24.1.0.

After you create a working copy of a gold image that you want to use for Exadata live update, provision that working copy to the Oracle Exadata database server using the `rhpctl update exadata -dbnodes` command.

1. Evaluate the current configuration and perform pre-upgrade checks.

```
rhpctl update exadata -dbnodes node1,node2 -client client1 -image
dblive_exadata
-iso_repo p28802055_236000_Linux-x86-64.zip -liveupdate full -eval
```

2. Create a backup of the current configuration.

```
rhpctl update exadata -dbnodes node1,node2 -client client1 -image
dblive_exadata
-iso_repo p28802055_236000_Linux-x86-64.zip -backup
```

3. Update Oracle Exadata Database node with the new image.

```
rhpctl update exadata -dbnodes node1,node2 -client client1 -image
dblive_exadata
-iso_repo p28802055_236000_Linux-x86-64.zip -liveupdate full
```

4. Query the Oracle Exadata client to check the client configuration and Exadata live update information.

```
rhpctl query client -client client1 -exadata

Audit ID: 90
Site: client1
Rapid Home Provisioning Client Version: 23.0.0.0.0
Rapid Home Provisioning Client Full Version: 23.6.0.0.0
Enabled: true
Host from which RHPC last registered: client1vm01.example.com
Port number last registered by RHPC: 8896
...
...
Image status: success
Exadata software version: 23.1.7.0.0.231012
Exadata Live Update Type: full
Exadata Live Update Version: 23.1.90.0.0.240226 (Live Update applied.
Reboot at any time to finalize outstanding items.)
Node type: GUEST
...
...
```

# Provisioning an Oracle Database Using Zip Copy

Starting with Oracle Grid Infrastructure 19c Release Update (19.11), Oracle FPP allows you to install the gold images without transferring them to the destination host.

This feature is known as *zipcopy* and you can use it to provision Oracle Database homes. You can also use this feature to provision Oracle Grid Infrastructure homes that exists on the destination hosts, but not to provision new Oracle Grid Infrastructure homes.

The ZipCopy feature has a requirement of shared storage between the Oracle FPP Server and the targets, including Oracle FPP Client and `rhpclient`-less targets. When importing an image to the Oracle FPP Server, you must provide to a zip file location of the Oracle Home software as input. You also need to specify a mount path on which this zip file is accessible on all targets where you want to provision this image. When provisioning using the ZipCopy feature, the Oracle FPP Server uses the zip file of the image already available on the mount path locally on the target hosts, instead of the transferring the software from the Oracle FPP server to the target hosts.

1. Import a new image from a zip file using the `-zip` and `-location` parameters.

```
$ rhpctl import image -image DB_1914 -zip /orastage/db1914000.zip -
location /orastage/db1914000.zip
```

   - `-zip` specifies the location from which you can import the image to the Oracle FPP server.
   - `-location` specifies a location where the image is available on the destination host as a zip file. Oracle FPP does not copy the zip file from the Oracle FPP server to the destination host.

> **Note:**
>
> You can make the image zip files available on the destination hosts using either local or shared storage.

2. Inspect the images on your Oracle Fleet Patching and Provisioning Server.

```
$ rhpctl query image -image DB_1914
```

   Make sure the `Location on target` value is available in the image query results.

3. Provision the working copy created using the zip file.

```
$ rhpctl add workingcopy -image DB_1914 -workingcopy fppc01_DB_1914 -user
oracle -oraclebase /u01/app/oracle
-client  dbSyslrfe3mla -path /u01/app/oracle/product/19.0.0.0/dbhome_2 -
localmount -location /orastage/db1914000.zip
-groups osdba=dba,osoper=dbaoper,osdg=dba,osbackup=dba,oskm=dba,osrac=dba -
storagetype LOCAL
```

   `fppc01_DB_1914` is the working copy based on the image `DB_1914`.

# Adding a Node to a Cluster and Scaling an Oracle RAC Database to the Node

You can add a node to your two-node cluster by using Fleet Patching and Provisioning to add the node, and then extend an Oracle RAC database to the new node.

**Before You Begin**

In this procedure, Oracle Grid Infrastructure 21c is running on the cluster. Working copy `GI_HOME_21c_WCPY` is the active Grid home on this cluster.

The Oracle RAC database home runs on the working copy `DB_HOME_21c_WCPY`.

Ensure that you have storage, network, and operating system requirements configured for the new node as stated in *Oracle Grid Infrastructure Installation Guide*.

**Procedure**

1. From the Fleet Patching and Provisioning Server, run the following command to add a node to the existing Oracle Grid Infrastructure working copy:

   ```
   rhpctl addnode gihome -workingcopy GI_HOME_21c_WCPY -newnodes n3:n3-vip
   {authentication_option}
   ```

   The command extends the cluster by adding `node3`.

2. Add instances to the administrator-managed Oracle RAC database on the new node:

   ```
   rhpctl addnode database -workingcopy DB_HOME_21c_WCPY -dbname db321 -node
   n3 {authentication_option}
   ```

   The command extends the database home on the `node3` and creates database `db321` on this node.

**Related Topics**

- rhpctl addnode gihome
- rhpctl addnode database
  Adds instances to an administrator-managed Oracle RAC database.

# Adding Gold Images for Fleet Patching and Provisioning

Create gold images of software home and store them on the Fleet Patching and Provisioning Server, to use later to provision Oracle homes.

**Before You Begin**

The Oracle home to be used for creating the gold image can be on the Fleet Patching and Provisioning Server, or Fleet Patching and Provisioning Client, or any `rhpclient`-less target machine that the Fleet Patching and Provisioning Server can communicate with.

**Procedure**

Create gold images of Oracle homes in any of the following ways and store them on the Fleet Patching and Provisioning server:

1. Import an image from an installed Oracle home on the Fleet Patching and Provisioning Server:

   ```
   rhpctl import image -image db23ai -path /share/software/23ai/dbhome -
   imagetype ORACLEDBSOFTWARE
   ```

   The gold image of `imagetype` Oracle Database 23ai software is created and stored on the Fleet Patching and Provisioning Server.

   You can also create gold images of Oracle Grid Infrastructure or any other software by specifying `-imagetype` as `ORACLEGISOFTWARE`, `ORACLEGGSOFTWARE`, or `SOFTWARE` respectively.

2. Import an image from an installed Oracle home on a Fleet Patching and Provisioning Client by running the following command from the Fleet Patching and Provisioning Client:

```
rhpctl import image -image db23ai -path /u01/app/dbusr/product/23.0.0/
```

The command creates and adds the image `db23ai` based on the local Oracle home installed in the specified path.

> **Note:**
>
> You cannot directly use images as software homes. Use images to create working copies of software homes.

# User Actions for Common Fleet Patching and Provisioning Tasks

You can use Fleet Patching and Provisioning user actions to perform many tasks, such as installing and configuring any type of software and running scripts.

**Deploying a Web Server**

The following procedure demonstrates automated deployment of Apache Web Server using Fleet Patching and Provisioning:

1. Create a script to install Apache Web server, as follows:
   a. On the Fleet Patching and Provisioning Server, download and extract the Apache Web server installation kit.
   b. Create the script to install, configure, and start the Apache Web server.
2. Register the script as a user action with Fleet Patching and Provisioning by running the following command on the Fleet Patching and Provisioning Server:

```
rhpctl useraction -useraction apachestart
-actionscript /user1/useractions/apacheinstall.sh
-post -optype ADD_WORKINGCOPY -onerror ABORT
```

The preceding command adds the `apachestart` user action for the action script stored in the specified directory. As per the specified properties, the user action runs after the `ADD_WORKINGCOPY` operation and terminates if there is any error.

3. Create an image type and associate the user action with the image type, as follows:

```
rhpctl add imagetype -imagetype apachetype -basetype SOFTWARE
-useraction "apachestart"
```

The preceding command creates a new image type called `apachetype`, a derivative of the basic image type, `SOFTWARE`, with an associated user action `apachestart`.

4. Create a gold image of the image type, as follows:

```
rhpctl import image -image apacheinstall -path /user1/apache2219_kit/
-imagetype apachetype
```

The preceding command creates a gold image, `apacheinstall`, with the script for Apache Web server installation, in the specified path, based on the `imagetype` you created earlier.

To view the properties of this image, run the `rhpctl query image -image apacheinstall` command.

5. Deploy a working copy of the gold image on the destination host, as follows:

```
rhpctl add workingcopy -workingcopy apachecopy -image apacheinstall
-path /user1/apacheinstallloc -sudouser user1
-sudopath /usr/local/bin/sudo -node node1 -user user1
-useractiondata "/user1/apachehome:1080:2.2.19"
```

Oracle FPP provisions the software to the destination host and runs the `apachestart` script specified in the user action. You can provide the Apache Web server configuration details such as port number with the `useractiondata` option. If the destination host is an Oracle FPP Client, then you need not specify `sudo` credentials.

### Registering Multiple Scripts Using a Single User Action

Run multiple scripts as part of a user action plug-in by registering a wrapper script and bundled custom scripts. The wrapper script extracts the bundled scripts, which are copied under the directory of the wrapper script, and then runs those extracted scripts as necessary, similar to the following procedure:

1. The following command creates a user action called `ohadd_ua`, and associates a wrapper script, `wc_add.sh`, with a zip file containing other scripts:

```
rhpctl add useraction -useraction ohadd_ua -actionscript
/scratch/crsusr/wc_add.sh -actionfile /scratch/crsusr/pack.zip -pre -
runscope
ALLNODES -optype ADD_WORKINGCOPY
```

The wrapper script, `wc_add.sh`, extracts the `pack.zip` file into the script path, a temporary path to which the user action scripts are copied. The wrapper script can invoke any scripts contained in the file.

2. The following command creates an image type, `sw_ua`, for the `ohadd_ua` user action:

```
rhpctl add imagetype -imagetype sw_ua -useractions ohadd_ua -basetype
SOFTWARE
```

3. The following command creates an image called `swimgua` from the software specified in the path:

```
rhpctl import image -image swimgua -path /tmp/custom_sw -imagetype sw_ua
```

4. The following command adds a working copy called `wcua` and runs the `wc_add.sh` script:

```
rhpctl add workingcopy -workingcopy wcua -image swimgua -client
destination_cluster
```

# A
# RHPCTL Command Reference

Use the Oracle Fleet Patching and Provisioning Control (RHPCTL) utility to manage Oracle Fleet Patching and Provisioning in your cluster.

This appendix contains reference information for Oracle Fleet Patching and Provisioning commands, including utility usage information and a comprehensive listing of the RHPCTL commands.

- RHPCTL Overview
  RHPCTL is a command-line utility with which you perform Oracle Fleet Patching and Provisioning operations and manage Oracle Fleet Patching and Provisioning Servers and Clients.
- Using RHPCTL Help
  You can use the content sensitive help with RHPCTL to get uses and syntax information of various commands.
- RHPCTL Command Reference
  This section describes RHPCTL command usage information, and lists and describes RHPCTL commands.

> **See Also:**
>
> *Oracle Database REST API Reference* for information about REST APIs for many common RHPCTL operations, including provisioning, patching, upgrading, and query operations.

## RHPCTL Overview

RHPCTL is a command-line utility with which you perform Oracle Fleet Patching and Provisioning operations and manage Oracle Fleet Patching and Provisioning Servers and Clients.

RHPCTL uses the following syntax:

```
rhpctl command object [parameters]
```

In RHPCTL syntax:

- *command* is a verb such as `add`, `delete`, or `query`
- *object* (also known as a noun) is the object on which RHPCTL performs the command, such as client or image.
- *parameters* extend the use of a preceding command combination to include additional parameters for the command. Specify parameters as *-keyword value*. If the *value* field contains a comma-delimited list, then do not use spaces between the items in the list.

You can use RHPCTL commands to perform several Oracle Fleet Patching and Provisioning operations, including:

- Oracle Fleet Patching and Provisioning Client operations, such as creating an Oracle Fleet Patching and Provisioning Client configuration.
- Role operations, such as adding and deleting roles, and granting and revoking roles for users.
- Site operations, such as obtaining configuration information for Oracle Fleet Patching and Provisioning Servers.
- Image operations, such as adding, deleting, and importing images.
- Image series operations, such as adding and deleting image series.
- Working copy operations, such as adding and deleting working copies.

# Using RHPCTL Help

You can use the content sensitive help with RHPCTL to get uses and syntax information of various commands.

To see help for all RHPCTL commands, from the command line enter:

```
rhpctl -help
```

To see the command syntax and a list of parameters for each RHPCTL command, from the command line enter:

```
rhpctl command (or verb) object (or noun) -help
```

# RHPCTL Command Reference

This section describes RHPCTL command usage information, and lists and describes RHPCTL commands.

> **Note:**
>
> A multitenant container database is the only supported architecture in Oracle Database 21c and later releases. While the documentation is being revised, legacy terminology may persist. In most cases, "database" and "non-CDB" refer to a CDB or PDB, depending on context. In some contexts, such as upgrades, "non-CDB" refers to a non-CDB from a previous release.

- archive Commands
  Use commands with the `archive` keyword to archive gold images and `unarchive` keyword to unarchive gold images.
- audit Commands
  Use commands with the `audit` keyword to delete, modify, and query audit records.
- backup and restore Commands
  Use commands with the `backup` and `restore` keywords to backup Oracle FPP Server and restore from the backup.
- client Commands
  Use commands with the `client` keyword to add, delete, and manage Oracle Fleet Patching and Provisioning clients.

- credentials Commands
  Use commands with the `credentials` keyword to add credentials to and delete credentials from the Oracle Cluster Registry (OCR).
- database Commands
  Use commands with the `database` keyword to add, delete, move, and upgrade databases.
- datapatch Commands
  Use commands with the `datapatch` keyword to apply patches to the specified Oracle Database.
- exadata Commands
  Use commands with the `exadata` keyword to patch an Oracle Exadata system.
- gihome Commands
  Use commands with the `gihome` keyword to add or delete nodes to Oracle Grid Infrastructure home and, move and upgrade Oracle Grid Infrastructure home.
- home Commands
  Use commands with the `home` keyword to deploy an Oracle Grid Infrastructure or Oracle Database home using Oracle FPP Local Mode.
- image Commands
  Use commands with the `image` keyword to add, delete, import, and manage gold images.
- imagetype Commands
  Use commands with the `imagetype` keyword to add, delete, modify, and manage an image type.
- job Commands
  Use commands with the `job` keyword to delete or query schedule jobs.
- osconfig Commands
  Use commands with the `osconfig` keyword to backup, compare, and manage operating system configuration information.
- patch Commands
  Use commands with the `patch` keyword to apply or rollback one-off patches to the specified working copy using the in-place patching method.
- peerserver Commands
  Use commands with the `peerserver` keyword to display information for a peer server.
- role Commands
  Use commands with the `role` keyword to add, delete, and manage roles.
- series Commands
  Use commands with the `series` keyword to add, delete, subscribe, and manage a series.
- server Commands
  Use commands with the `server` keyword to export, register, unregister, and query Oracle Fleet Patching and Provisioning Server.
- tag Commands
  Use commands with the `tag` keyword to add, query, modify, delete, set, and unset tags.
- user Commands
  Use commands with the `user` keyword to delete, modify, register, and unregister users.
- useraction Commands
  Use commands with the `useraction` keyword to add, delete, and modify user actions.

- workingcopy Commands
  Use commands with the `workingcopy` keyword to create, update, extend, and delete working copies.

# job Commands

Use commands with the `job` keyword to delete or query schedule jobs.

- rhpctl delete job
  Deletes a specific scheduled job from the repository.
- rhpctl query job
  Queries the current status of a scheduled job with a specific job ID.
- rhpctl delete job
  Deletes a specific scheduled job from the repository.

## rhpctl delete job

Deletes a specific scheduled job from the repository.

### Syntax

```
rhpctl delete job [-jobid job_id] [-force]
```

### Parameters

**Table A-1 rhpctl delete job Command Parameters**

| Parameter | Description |
|---|---|
| `-jobid job_id` | Optionally, you can specify the job ID value for the job you want to delete that you obtained while scheduling the job. If you choose not to use this parameter, then RHPCTL deletes all jobs. |
| `-force` | Use this parameter to forcibly delete a job. |

### Usage Notes

You must run this command on the Fleet Patching and Provisioning Server.

### Example

To delete a job with a job ID of `1`:

```
$ rhpctl delete job -jobid 1
```

## rhpctl query job

Queries the current status of a scheduled job with a specific job ID.

### Syntax

```
rhpctl query job [-jobid job_id] [-status {FAILED | SUCCEEDED | SCHEDULED
   | EXECUTING | UNKNOWN | TERMINATED }] [-client client_name] [-user
```

```
user_name]
  [-since timer_value] [-summary] [-eval] [-migrate]
```

### Parameters

**Table A-2 rhpctl query job Command Parameters**

| Parameter | Description |
|---|---|
| `-jobid job_id` | Optionally, you can specify the job ID value for the job you want to query. The job Id is obtained while scheduling the job.<br>If you choose this parameter, then the only other option you can specify is `-summary`. If you do not choose this parameter, then all jobs are queried. |
| `-status {EXECUTED \| TIMER_RUNNING \| EXECUTING \| UNKNOWN \| TERMINATED }` | Optionally, you can specify any of the following states of a job that you want to query:<br>• `EXECUTED`: The job is complete.<br>• `TIMER_RUNNING`: The timer for the job is still running.<br>• `EXECUTING`: The timer for the job has expired and is running.<br>• `UNKNOWN`: There is an unexpected failure due to issues such as a server going down, nodes going down, or any resource failures.<br>• `TERMINATED`: There is an abrupt failure or the operation has stopped. |
| `-client client_namek` | Optionally, you can specify the name of a client cluster for which you want to query jobs. |
| `-user user_name` | Optionally, you can specify the user name of the user for whom a software home is being provisioned. |
| `-since timer_value` | Optionally, you can specify a date from which to query the jobs, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05` |
| `-summary` | Optionally, you can use this parameter to return only job details. |
| `-eval` | Optionally, you can use this parameter to query only evaluation jobs. |
| `-migrate` | Optionally, you can use this parameter to query only migration jobs. |

### Usage Notes

You must run this command on the Fleet Patching and Provisioning Server.

### Example

To query a specified scheduled job:

```
$ rhpctl query job -jobid 1
```

This command returns output similar to the following:

```
Job ID: 1
User: fred
Client: fredlinux4
Scheduled job command: "rhpctl import image -image DB-Image1 -imagetype
ORACLEDBSOFTWARE -path /ade/fred_linux4/esw1 -schedule 2018-07-27T13:38:57Z"
Scheduled job execution start time: 2018-07-27T05:38:57-08. Equivalent local
```

```
time: 2018-07-27 05:38:57
Current status: EXECUTED
Result file path: "/scratch/rhp_storage/chkbase/scheduled/
job-1-2017-11-27-05:39:14.log"
Job execution start time: 2018-07-27 05:39:14
Job execution end time: 2018-07-27 05:43:09
Job execution elapsed time: 3 minutes 55 seconds

Result file "/scratch/rhp_storage/chkbase/scheduled/
job-1-2018-07-27-05:39:14.log" contents:
slc05amw.example.com: Audit ID: 4
slc05amw.example.com: Creating a new ACFS file system for image "DB-
Image1" ...
slc05amw.example.com: Copying files...
slc05amw.example.com: Copying home contents...
slc05amw.example.com: Changing the home ownership to user fred...
slc05amw.example.com: Changing the home ownership to user fred...
```

## rhpctl delete job

Deletes a specific scheduled job from the repository.

### Syntax

```
rhpctl delete job [-jobid job_id] [-force]
```

### Parameters

**Table A-3 rhpctl delete job Command Parameters**

| Parameter | Description |
|---|---|
| `-jobid job_id` | Optionally, you can specify the job ID value for the job you want to delete that you obtained while scheduling the job. If you choose not to use this parameter, then RHPCTL deletes all jobs. |
| `-force` | Use this parameter to forcibly delete a job. |

### Usage Notes

You must run this command on the Fleet Patching and Provisioning Server.

### Example

To delete a job with a job ID of `1`:

```
$ rhpctl delete job -jobid 1
```

# archive Commands

Use commands with the `archive` keyword to archive gold images and `unarchive` keyword to unarchive gold images.

- rhpctl archive image
  Archives the Oracle Fleet Patching and Provisioning (Oracle FPP) gold images.

- rhpctl unarchive image
  Unarchives the archived Oracle Fleet Patching and Provisioning (Oracle FPP) gold images.

## rhpctl archive image

Archives the Oracle Fleet Patching and Provisioning (Oracle FPP) gold images.

**Syntax**

```
rhpctl archive image -image image_name
```

`image_name` is the name of the gold image that you want to archive. This image must not have a working copy created using it.

## rhpctl unarchive image

Unarchives the archived Oracle Fleet Patching and Provisioning (Oracle FPP) gold images.

**Syntax**

```
rhpctl unarchive image -image image_name
```

`image_name` is the name of the archived gold image that you want to unarchive.

# audit Commands

Use commands with the `audit` keyword to delete, modify, and query audit records.

- rhpctl delete audit
  Deletes the Fleet Patching and Provisioning audit records.
- rhpctl modify audit
  Modifies the maximum number of audit records to store.
- rhpctl query audit
  Displays the Fleet Patching and Provisioning audit records.

## rhpctl delete audit

Deletes the Fleet Patching and Provisioning audit records.

**Syntax**

```
rhpctl delete audit [-to timestamp]
```

**Usage Notes**

Optionally, you can specify a date up to which audit records will be deleted, in the format *YYYY-MM-DD*. Otherwise, this command deletes all audit records.

## rhpctl modify audit

Modifies the maximum number of audit records to store.

### Syntax

```
rhpctl modify audit -maxrecord number
```

### Usage Notes

Specify the maximum number of audit records to store.

## rhpctl query audit

Displays the Fleet Patching and Provisioning audit records.

### Syntax

```
rhpctl query audit [[[-operation {add | delete | modify | grant | revoke |
move | verify | discover
  | upgrade | allow | disallow | deleteimage | insertimage | promote |
addnode | deletenode | register | unregister | export | import | query
  | subscribe | unsubscribe}]
  [-entity {client | role | audit | image | imagetype | useraction | series |
workingcopy | database | server | user | audit | imagetype | useraction}]
  | [-user user_name] [-client cluster_name] | [-from timestamp -to timestamp]
  | -before timestamp | -since timestamp | -first number | -last number]
  | -record record_id | -config]
```

### Parameters

**Table A-4 rhpctl query audit Command Parameters**

| Parameter | Description |
|---|---|
| `-operation {add \| delete \| modify \| grant \| revoke \| move \| verify \| discover \| upgrade \| allow \| disallow \| deleteimage \| insertimage \| promote \| addnode \| deletenode \| register \| unregister \| export \| import \| query \| subscribe \| unsubscribe}` | Specify the type of operation for which you want an audit query. |
| `-entity {client \| role \| image \| series \| workingcopy \| database \| server \| user \| audit \| imagetype \| useraction}` | Specify the entity for which you want an audit query. |
| `-user user_name` | Optionally, you can choose to run a query audit on a particular user who performed Fleet Patching and Provisioning operations. |

**Table A-4 (Cont.) rhpctl query audit Command Parameters**

| Parameter | Description |
|---|---|
| `-client` *`cluster_name`* | Optionally, you can choose to run a query audit on a particular client cluster where Fleet Patching and Provisioning operations were performed. |
| `-from` *`timestamp`* `-to` *`timestamp`* | Optionally, you can specify a time interval for which to run an audit query. Timestamps must be in the format *`YYYY-MM-DD`*. |
| `-before` *`timestamp`* | Optionally, you can specify a time before which to run an audit query. Timestamp must be in the format *`YYYY-MM-DD`*. |
| `-since` *`timestamp`* | Optionally, you can specify a time after which to run an audit query. Timestamp must be in the format *`YYYY-MM-DD`*. |
| `-first` *`number`* | Optionally, you can specify a number of the first audit records for a given time. |
| `-last` *`number`* | Optionally, you can specify a number of the last audit records for a given time. |
| `-record` *`record_id`* | Optionally, you can specify a particular audit record ID. |
| `-config` | You can choose this parameter to show the maximum record configuration. |

# backup and restore Commands

Use commands with the `backup` and `restore` keywords to backup Oracle FPP Server and restore from the backup.

- rhpctl backup server
  Enables a user to perform a full backup of the Oracle FPP Server.
- rhpctl restore server
  Enables you to restore or relocate an Oracle FPP Server for the specified backup.

## rhpctl backup server

Enables a user to perform a full backup of the Oracle FPP Server.

**Syntax**

```
rhpctl backup server -path backup_path
```

**Parameters**

**Table A-5 rhpctl backup server Command Parameters**

| Parameter | Description |
|---|---|
| `-path` *`backup_path`* | Specify an absolute path of a writable directory in which you want to store your backup. |

## rhpctl restore server

Enables you to restore or relocate an Oracle FPP Server for the specified backup.

### Syntax

```
rhpctl restore server -backup backup_location
```

### Parameters

**Table A-6 rhpctl restore server Command Parameters**

| Parameter | Description |
|---|---|
| -backup *backup_location* | Specify the backup location where your backup file is stored. |

## client Commands

Use commands with the `client` keyword to add, delete, and manage Oracle Fleet Patching and Provisioning clients.

- rhpctl add client
  Adds a Fleet Patching and Provisioning Client to the Fleet Patching and Provisioning Server configuration.
- rhpctl allow client
- rhpctl delete client
- rhpctl disallow client
- rhpctl discover client
- rhpctl export client
- rhpctl modify client
  Modifies an Oracle Fleet Patching and Provisioning Client.
- rhpctl query client
  Displays the configuration information of a specific Fleet Patching and Provisioning Client cluster.
- rhpctl update client
- rhpctl verify client

## rhpctl add client

Adds a Fleet Patching and Provisioning Client to the Fleet Patching and Provisioning Server configuration.

### Syntax

```
rhpctl add client -client cluster_name [-clusternamealias cluster_name_alias]
[-toclientdata path] [-targetnode node_name {-sudouser sudo_user_name
-sudopath sudo_binary_location | -root | -cred cred_name} | -auth plugin_name
[-arg1 name1:value1...]] [-maproles role=user_name
[,role=user_name[,...]]] [-version version]
```

### Parameters

**Table A-7 rhpctl add client Command Parameters**

| Parameter | Description |
|---|---|
| `-client client_name` | Specify the name of the cluster in which you want to create the client.<br><br>**Note:**<br>Oracle recommends that you specify a unique name for the client cluster. |
| `-clusternamealias` | Optionally, you can specify the client cluster alias if the client cluster name is not unique. |
| `-toclientdata path` | Optionally, you can specify the path to the XML file that is created by the Fleet Patching and Provisioning Server (specific to the client cluster), which contains the information the client needs to configure its connection to the server. |
| `-targetnode node_name` | Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client. |
| `-sudouser sudo_user_name -sudopath sudo_binary_location \| -root \| -cred cred_name` | If you choose to use the `-targetnode` parameter, then you must choose either `sudo` or `root` to access the remote node.<br><br>If you choose `sudo`, then you must use the `-sudouser` parameter and specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br><br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node. |
| `-auth plugin_name [-arg1 name1:value1...]` | Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-maproles role=user_name[,...]` | You can specify either built-in roles or roles that you have defined, and you can assign multiple uses to each role. Use commas to separate multiple roles and users.<br><br>**Note:**<br>Starting with Oracle Grid Infrastructure 21c, the `-maproles` parameter is deprecated. This parameter can be desupported in a future release. |

**Table A-7 (Cont.) rhpctl add client Command Parameters**

| Parameter | Description |
|---|---|
| `-version version` | Optionally, you can specify the Oracle FPP Client software version, such as `21.0`.<br><br>**Note:**<br>If the Oracle FPP Client version is lower than the Oracle FPP Server version, then you must specify this parameter. For example, if the client version is 12.2, then specify the *-version* parameter as *-version 12.2*. |

### Usage Notes

- Only clusters running Oracle Grid Infrastructure 12c release 2 (12.2) or later can be configured and added as Fleet Patching and Provisioning Clients. Clusters running earlier versions of Oracle Grid Infrastructure, and servers running no Oracle Grid Infrastructure, can be managed directly by the Fleet Patching and Provisioning Server.
- You can only run this command on the Fleet Patching and Provisioning Server.

### Examples

To add a client to the Fleet Patching and Provisioning Server:

```
$ rhpctl add client -client ClientCluster3 -toclientdata Grid_home/RHPserver/
info -version 12.2
```

## rhpctl allow client

Enables a user or a role to perform an operation on an Oracle Fleet Patching and Provisioning client or remote cluster.

### Syntax

```
rhpctl allow client -client client_name
   {-user username [-cluster  cluster_name] | -role role_name}
```

### Parameters

**Table A-8 rhpctl allow client Command Parameters**

| Parameter | Description |
|---|---|
| `-client client_name` | Specify the name of the FPP client or remote cluster. |
| `-user username` | Specify the name of the operating system (OS) user |
| `-cluster cluster_name` | Optionally, you can specify the name of a cluster. |

**Table A-8 (Cont.) rhpctl allow client Command Parameters**

| Parameter | Description |
|---|---|
| `-role role_name` | Specify the name of the role. |

**Usage Notes**

- Specify the name of an Oracle Fleet Patching and Provisioning client or a remote cluster.
- Specify the OS user and optionally a name of a cluster, or the name of a role.

  For a list of roles, refer to rhpctl add role
- This command can only be run on an Oracle Fleet Patching and Provisioning server

**Examples**

**Example A-1 Using rhpctl allow client**

```
$ rhpctl add role -role SECURE_CLIENT_USER -hasRoles GH_CLIENT_ACCESS

$ rhpctl disallow client -client secure_rhpc1 -role GH_CLIENT_ACCESS

$ rhpctl allow client -client secure_rhpc1 -role SECURE_CLIENT_USER

$ rhpctl grant role -role SECURE_CLIENT_USER -user secure_user1
```

## rhpctl delete client

Deletes a specific Fleet Patching and Provisioning Client from the configuration.

**Syntax**

```
rhpctl delete client -client cluster_name [-force]
```

**Usage Notes**

- Specify the name of the client cluster that you want to delete from the configuration.
- You must stop the Fleet Patching and Provisioning Client before you run this command or use the `-force` option.

**Example**

To delete the Fleet Patching and Provisioning Client `ClientCluster3`:

```
$ rhpctl delete client -client ClientCluster3
```

## rhpctl disallow client

Disables a user or a role to perform an operation on an Oracle Fleet Patching and Provisioning client or remote cluster.

### Syntax

```
rhpctl disallow client -client client_name
    {-user username [-cluster  cluster_name] | -role role_name}
```

### Parameters

**Table A-9 rhpctl disallow client Command Parameters**

| Parameter | Description |
|---|---|
| `-client client_name` | Specify the name of the FPP client or remote cluster. |
| `-user username` | Specify the name of the operating system (OS) user |
| `-cluster cluster_name` | Optionally, you can specify the name of a cluster. |
| `-role role_name` | Specify the name of the role. |

### Usage Notes

- Specify the name of an Oracle Fleet Patching and Provisioning client or a remote cluster.
- Specify the OS user and optionally a name of a cluster, or the name of a role.

  For a list of roles, refer to rhpctl add role
- This command can only be run on an Oracle Fleet Patching and Provisioning server

### Examples

**Example A-2 Using rhpctl disallow client**

```
$ rhpctl add role -role SECURE_CLIENT_USER -hasRoles GH_CLIENT_ACCESS

$ rhpctl disallow client -client secure_rhpc1 -role GH_CLIENT_ACCESS

$ rhpctl allow client -client secure_rhpc1 -role SECURE_CLIENT_USER

$ rhpctl grant role -role SECURE_CLIENT_USER -user secure_user1
```

## rhpctl discover client

Validates the input provided and discovers parameters on the given nodes, and generates a response file that you can use for configuring Oracle Clusterware.
After completing this command, use page A-20 to validate the response file and prepare the destination nodes for Oracle Clusterware deployment.

### Syntax

```
rhpctl discover client -image image_name -generatepath response_file_path
  {-responsefile response_file_name | -clusternodes node_list -client
cluster_name
   -oraclehome oracle_home_path} {-root | -sudouser sudo_username
   -sudopath sudo_binary_path | -cred cred_name | -auth plugin_name
```

```
[-arg1 name1:value1...]} [-user gi_user_name]
[-scan scan_name]
```

### Parameters

**Table A-10 rhpctl discover client Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of the Oracle Grid Infrastructure Gold Image which the resulting response file will support. |
| `-generatepath response_file_path` | Specify a file path where the response file that RHPCTL generates will be copied. The RHPCTL command generates name of the response file and displays it while the command is running. |
| `-responsefile response_file_name` | If you have a partially complete response file and you want it to be completed with reference to the destination nodes, then specify the response file name using this parameter.<br>**Note:** The response file must include the node list, client name, and Oracle home path. |
| `-clusternodes node_list` | Specify a comma-delimited list of nodes on which you plan to provision Oracle Clusterware (using the resulting response file) in the following format: `node_name:node_vip[:node_role] [,node_name:node_vip[:node_role]...]` |
| `-client cluster_name` | Specify the name of the Oracle FPP Client cluster to be probed. |
| `-oraclehome oracle_home_path` | Specify the location of the Oracle home. |
| `-root \| -sudouser sudo_username -sudopath sudo_binary_path \| -cred cred_name \| -auth plugin_name [-arg1 name1:value1...]` | You must choose either `sudo` or `root` to access the remote nodes.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-user gi_user_name` | Specify the name of the Oracle Grid Infrastructure installation user. |
| `-scan scan_name` | Specify the SCAN name. |

## rhpctl export client

Exports data from the repository on the Fleet Patching and Provisioning Server to a client data file.

### Syntax

```
rhpctl export client -client cluster_name -clientdata file_path
```

### Parameters

**Table A-11 rhpctl export client Command Parameters**

| Parameter | Description |
|---|---|
| `-client cluster_name` | Specify the name of the client cluster that you want to export. |
| `-clientdata file_path` | Specify the path to the location of the client data file. |

### Usage Notes

You can only run this command on the Fleet Patching and Provisioning Server.

### Example

To export repository data from a Fleet Patching and Provisioning Client named `mjk9394` to a client data file, `/tmp/mjk9394.xml`:

```
$ rhpctl export client -client mjk9394 -clientdata /tmp/mjk9394.xml
```

## rhpctl modify client

Modifies an Oracle Fleet Patching and Provisioning Client.

### Syntax

```
rhpctl modify client -client cluster_name [-enabled {TRUE | FALSE}]
  [-maproles role=user_name[+user_name...]
[,role=user_name[+user_name...],...]]] [-password]]
```

### Parameters

**Table A-12 rhpctl modify client Command Parameters**

| Parameter | Description |
|---|---|
| `-client cluster_name` | Specify the name of the client cluster that you want to modify. |
| `-enabled {TRUE \| FALSE}` | Specify whether the client is enabled. |

**Table A-12 (Cont.) rhpctl modify client Command Parameters**

| Parameter | Description |
|---|---|
| `-maproles role=user_name[+user_name...][,...]` | You can modify either built-in roles or roles that you have defined, and you can assign multiple uses to each role.<br>When you use the `-maproles` parameter, use a plus sign (+) to map more than one user to a specific role. Separate additional role/user pairs with commas.<br>**Note:**<br>Starting with Oracle Grid Infrastructure 21c, the `-maproles` parameter is deprecated. This parameter can be desupported in a future release. |
| `-password` | Optionally, you can recreate the Oracle Fleet Patching and Provisioning Client credentials. |

### Example

To disable an Oracle Fleet Patching and Provisioning Client named `RHPClient001`:

```
$ rhpctl modify client -client RHPClient001 -enabled FALSE
```

## rhpctl query client

Displays the configuration information of a specific Fleet Patching and Provisioning Client cluster.

### Syntax

```
rhpctl query client [-client cluster_name [-detail [-node node_name] [-
displayhtml]]]
  [-rhpserver rhps_regex] [-exadata]
```

### Parameters

**Table A-13 rhpctl query client Command Parameters**

| Parameter | Description |
|---|---|
| `-client cluster_name` | Specify the name of the client cluster in which the Fleet Patching and Provisioning Client resides. |
| `-detail` | Generate cluster information (HTML) for the cluster. |
| `-node node_name` | Specify the name of a node in a remote cluster. |
| `-displayhtml` | Display the HTML cluster information for the cluster. |

**Table A-13 (Cont.) rhpctl query client Command Parameters**

| Parameter | Description |
|---|---|
| `-rhpserver` *rhps_regex* | Specify a regular expression to match the cluster name of the servers where the operation must be performed. |
| `-exadata` | Specify the client type as Exadata to query an Exadata database node. |

### Usage Notes

Specify the name of the client cluster in which the Fleet Patching and Provisioning Client resides for which you want to display the configuration information.

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles
role=user_name[+user_name...][,role=user_name[+user_name...]...]
```

For information about granting roles with RHPCTL, refer to rhpctl grant role

### Examples

This command displays output similar to the following:

```
/rhpctl query client -client mbcluster-13
Site: mbcluster-13
Fleet Patching and Provisioning Client Version: 12.2.0.1.0
Enabled: true
Host from which RHPC last registered: rhpserver01.example.com
Port number last registered by RHPC: 8896
RHP Enabled: true
Standalone: false
Managed: true
```

To query Oracle Exadata database server node and check live update status:

```
rhpctl query client -client client1 -exadata

Audit ID: 90
Site: client1
Rapid Home Provisioning Client Version: 23.0.0.0.0
Rapid Home Provisioning Client Full Version: 23.6.0.0.0
Enabled: true
Host from which RHPC last registered: client1vm01.example.com
Port number last registered by RHPC: 8896
```

```
...
...
Image status: success
Exadata software version: 23.1.7.0.0.231012
Exadata Live Update Type: full
Exadata Live Update Version: 23.1.90.0.0.240226 (Live Update applied. Reboot
at any time to finalize outstanding items.)
Node type: GUEST
...
...
```

## rhpctl update client

Updates an image on the Fleet Patching and Provisioning Client.

### Syntax

```
rhpctl update client -image image_name {-targetnode node_name
  | -batches '(node_name)'} -root
```

### Parameters

**Table A-14 rhpctl update client Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of the image that you want to update. |
| `-targetnode node_name` | Specify the name of the node on which you want to update the Fleet Patching and Provisioning Client. |
| `-batches '(node_name)'` | Alternative to specifying a node name, you can specify batches of nodes.<br>**Note:** If you use this parameter for Oracle Database Appliance nodes, then run the command twice, in succession, specifying any one Oracle Database Appliance node for the first run, and another Oracle Database Appliance node for the second run. |
| `-root` | You must specify this parameter if you use either the `-targetnode` or `-batches` parameters. |

### Usage Notes

You can only run this command from a Fleet Patching and Provisioning Server.

### Examples

The following example uses the `-targetnode` parameter:

```
$ rhpctl update client -image ODA1 -targetnode rac07box1 -root
```

The two following examples use the `-batches` parameter:

```
$ rhpctl update client -image ODA1 -batches '(rac07box1)' -root
$ rhpctl update client -image ODA1 -batches '(rac07box2)' -root
```

## rhpctl verify client

Validates the input provided and creates or completes and verifies the values in a response file that you can use to configure Oracle Clusterware.

### Syntax

```
rhpctl verify client -image image_name -responsefile response_file_name
  [-clusternodes node_list] {-root | -sudouser sudo_username -sudopath
   sudo_binary_path | -cred cred_name} | -auth plugin_name [-arg1
name1:value1...]
  [-user gi_user_name] [-client cluster_name] [-scan scan_name]
  [-oraclehome oracle_home_path] [-ignorewarn] [-fixup [-setupSSH]]
```

### Parameters

**Table A-15 rhpctl verify client Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of the image. |
| `-responsefile response_file_name` | Specify a response file to be used to provision Oracle Grid Infrastructure. |
| `-clusternodes node_list` | Specify a comma-delimited list of nodes on which Oracle Clusterware will be provisioned in the following format: `node_name:node_vip[:node_role] [,node_name:node_vip[:node_role]...]` |
| `-root \| -sudouser sudo_username -sudopath sudo_binary_path \| -cred cred_name \| -auth plugin_name [-arg1 name1:value1...]` | You must choose either `sudo` or `root` to access the remote nodes.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-user gi_user_name` | Specify the name of the Oracle Grid Infrastructure installation user. |
| `-client cluster_name` | Specify the name of the cluster you want to verify. |
| `-scan scan_name` | Specify the SCAN name. |
| `-oraclehome oracle_home_path` | Specify the location of the Oracle home. |
| `-ignorewarn` | Use this parameter to ignore warnings during validation. |
| `-fixup [-setupSSH]` | Use this parameter to run a fixup script, which automatically applies changes to the nodes to satisfy changes that CVU recommends.<br>Optionally, you can use the `-setupSSH` parameter to set up passwordless SSH user equivalence on the remote nodes for the provisioning user. |

## credentials Commands

Use commands with the `credentials` keyword to add credentials to and delete credentials from the Oracle Cluster Registry (OCR).

- rhpctl add credentials
- rhpctl delete credentials

## rhpctl add credentials

Adds credentials to the Oracle Cluster Registry (OCR).

### Syntax

```
rhpctl add credentials -cred cred_name {-root | -sudouser sudo_user_name
  -sudopath sudo_binary_location}
```

### Parameters

**Table A-16 rhpctl add credentials Command Parameters**

| Parameter | Description |
|---|---|
| `-cred` *cred_name* | Specify a credential name to associate the user and password credentials to access a remote node. |
| `-root | -sudouser` *sudo_user_name* `-sudopath` *sudo_binary_location* | You must choose either to provide `root` access to access a remote node or a `sudo` user name and path to the `sudo` binary to perform super user operations. |

### Usage Notes

After you add credentials they can be used in the `-cred` *cred_name* parameter of other commands to avoid those commands prompting for a password.

## rhpctl delete credentials

Deletes credentials from the Oracle Cluster Registry (OCR).

### Syntax

```
rhpctl delete credentials -cred cred_name
```

### Usage Notes

Specify only the name of the credentials you want to delete.

# database Commands

Use commands with the `database` keyword to add, delete, move, and upgrade databases.

- rhpctl add database
  Creates a database using a specific working copy.
- rhpctl addnode database
  Adds instances to an administrator-managed Oracle RAC database.
- rhpctl addpdb database

- rhpctl deletepdb database
  Deletes a Pluggable Database (PDB) to an existing Container Database (CDB) on a working copy.
- rhpctl delete database
  Deletes a database that was created on a working copy.
- rhpctl deletenode database
  Deletes an instance, which contracts an administrator-managed Oracle RAC database.
- rhpctl move database
  Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.
- rhpctl movepdb database
  Moves specified Pluggable Databases (PDBs) from the source multitenant Container Database (CDB) to the destination multitenant CDB.
- rhpctl upgrade database
  Upgrades a database to the version of the destination working copy.
- rhpctl zdtupgrade database
  The Zero Downtime Upgrade command `rhpctl zdtupgrade database` enables zero downtime database upgrades for Oracle RAC and Oracle RAC One Node Oracle Database instances.

## rhpctl add database

Creates a database using a specific working copy.

### Syntax

```
rhpctl add database -workingcopy workingcopy_name
  {-dbname unique_db_name {-node node_list |
         [-dbtype {RACONENODE | RAC | SINGLE}]
         [-dbtemplate file_path | image_name:relative_file_path]
         [-cdb] [-pdbName pdb_prefix [-numberOfPDBs pdb_count]]
         [{-sudouser user_name -sudopath sudo_binary_path |
          -root | -cred cred_name |
          -auth plugin_name [-arg1 name1:value1...]}]
          [-targetnode node_name]
          [-ignoreprereq]
          [-fixup]}
  [-datafileDestination datafileDestination_path]
  [-useractiondata user_action_data]
  [-eval] [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name]
```

### Parameters

**Table A-17 rhpctl add database Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy` *workingcopy_name* | Specify the name of an existing working copy for the database that you want to add. |
| `-dbname` *unique_db_name* | Specify the unique name of the database (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you are adding. |

**Table A-17 (Cont.) rhpctl add database Command Parameters**

| Parameter | Description |
|---|---|
| -datafileDestination *datafileDestination_path* | Specify the data file destination location or the name of the Oracle Automatic Storage Management (Oracle ASM) disk group.<br>**Note:** You cannot specify a disk group for Oracle Database versions before Oracle Database 11*g* release 2 (11.2). |
| -node *node_list* | Specify a node or comma-delimited list of several nodes on which to create the database. |
| -dbtype {RACONENODE \| RAC \| SINGLE} | Specify whether the database is Oracle RAC One Node, Oracle RAC, or a nonclustered database. |
| -dbtemplate *file_path* \| *image_name*:*relative_file_path* | Specify the absolute file path to a database template or the relative path to the image home directory on a Fleet Patching and Provisioning Server. |
| -cdb | Optionally, use this parameter to create a database as a Container Database (CDB). |
| -pdbName *pdb_prefix* | If you are creating one or more Pluggable Databases (PDBs), then specify a PDB name prefix. |
| -numberOfPDBs *pdb_count* | Specify the number of PDBs you want to create. |
| -sudouser *user_name* -sudopath *sudo_binary_path* \| -root \| -cred *cred_name* \| -auth *plugin_name* [-arg1 *name1*:*value1*...] | If you choose to use the -targetnode parameter, then you must choose either sudo or root to access the remote node.<br>If you choose sudo, then you must specify a user name to run super-user operations, and a path to the location of the sudo binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to -sudouser, -root, or -cred, you can use -auth to specify an authentication plugin to access a remote node. |
| -targetnode *node_name* | Optionally, you can specify the name of a node in a remote cluster that has no Oracle Fleet Patching and Provisioning Client. |
| -ignoreprereq | Ignore prerequisites. |
| -fixup | Run a fixup script. This option is valid for Oracle Grid Infrastructure and database provisioning. |
| -useractiondata *user_action_data* | Optionally, you can pass a value to the useractiondata parameter of the user action script. |
| -schedule {*timer_value* \| NOW \| PAUSE} | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>2018-07-25T19:13:17+05<br>If NOW is specified or the option is omitted, then the job is scheduled immediately.<br>If PAUSE is specified, then the job starts in the paused state and you need to resume the job using the rhpctl resume job -jobid *job_id* command. |
| -jobtag *tag_name* | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| -eval | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |

### Usage Notes

If you choose to use the -schedule parameter, then you must run this command on the Fleet Patching and Provisioning Server.

Authentication plugins enable the addition of authentication methods without changes in command line interfaces (CLIs). For information about authentication options, refer to Authentication Options for Oracle Fleet Patching and Provisioning Operations.

### Examples

To create a database on a working copy named `prodhome`:

```
$ rhpctl add database -workingcopy prodhome -dbname proddb -
datafileDestination /acfs/proddata -dbtype RAC
```

> **Note:**
>
> You can create multiple databases on a working copy.

## rhpctl addnode database

Adds instances to an administrator-managed Oracle RAC database.

### Syntax

```
rhpctl addnode database -workingcopy workingcopy_name
  -dbname unique_db_name -node node_list
   [-root | -cred cred_name | -sudouser sudo_user_name
    -sudopath sudo_binary_location |
    -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
   [-useractiondata user_action_data] [-eval] [-schedule {timer_value | NOW |
PAUSE}] [-jobtag tag_name]
   [-revert]
```

### Parameters

**Table A-18 rhpctl addnode database Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of a working copy. |
| `-dbname unique_db_name` | Specify the unique name of the database (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you are adding. |
| `-node node_list` | Specify a node or comma-delimited list of several nodes on which to create the database. |
| `-root \| -cred cred_name \| -sudouser sudo_user_name -sudopath sudo_binary_location \| -auth plugin_name plugin_args` | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |

**Table A-18 (Cont.) rhpctl addnode database Command Parameters**

| Parameter | Description |
|---|---|
| `-useractiondata` *user_action_data* | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-eval` | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |
| `-schedule {`*timer_value* `| NOW | PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br><br>`2018-07-25T19:13:17+05`<br><br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid` *job_id* command. |
| `-jobtag` *tag_name* | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-revert` | On rerun, roll back the failed actions performed on the first run. |

### Usage Notes

- If the specified working copy is not installed on the nodes in the node list, then you must first run `rhpctl addnode workingcopy`.
- If the working copy is on a Fleet Patching and Provisioning Client or on the Fleet Patching and Provisioning Server, then credentials are not required. This is true whether you run the command on the Server or the Client. Credentials are required when you run the command on the Server and the working copy is on an `rhpclient`-less target.
- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

## rhpctl addpdb database

Adds a Pluggable Database (PDB) to an existing Container Database (CDB) on a working copy.
After you create a working copy of a gold image and provision that working copy to a node, and create a database as a multitenant CDB, then you can add a PDB to the CDB on the working copy.

### Syntax

```
rhpctl addpdb database -workingcopy workingcopy_name -cdbname cdb_name
  -pdbName new_pdb_name
   [-pdbDatafileDestination pdb_datafile_destination_path]
   [-pdbadminusername pdb_admin_user_name]
   [-dbcaresponsefile responsefilename]
   [-root | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]
      | -sudouser user_name -sudopath sudo_binary_path]
   [-targetnode node_name]
```

```
[-useractiondata user_action_data]
[-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name]
```

### Parameters

**Table A-19 rhpctl addpdb database Command Parameters**

| Parameter | Description |
|---|---|
| -workingcopy *workingcopy_name* | Specify the name of an existing working copy for the PDB that you want to add. |
| -cdbname *cdb_name* | Specify the name of the multitenant CDB to which you want to add the PDB. |
| -pdbName *new_pdb_name* | Specify a name for the PDB you are adding. |
| -pdbDatafileDestination *pdb_datafile_destination* | Optionally, you can specify the path to the data file destination location for the PDB. |
| -pdbadminusername *pdb_admin_user_name* | Optionally, you can specify the name of the PDB's local administrator. |
| -dbcaresponsefile *responsefilename* | Optionally, you can specify the name of the response file. |
| -root \| -cred *cred_name* \| -auth *plugin_name* [-arg1 *name1*:*value1*...] \| -sudouser *user_name* -sudopath *sudo_binary_path* | If you choose to use the -targetnode parameter, then you must choose either sudo or root to access the remote node.<br>If you choose sudo, then you must specify a user name to run super-user operations, and a path to the location of the sudo binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to -sudouser, -root, or -cred, you can use -auth to specify an authentication plugin to access a remote node. |
| -targetnode *node_name* | Optionally, you can specify the name of a node in the cluster on which you want to run this operation. |
| -useractiondata *user_action_data* | Optionally, you can pass a value to the useractiondata parameter of the user action script. |
| -schedule {*timer_value* \| NOW \| PAUSE} | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>2018-07-25T19:13:17+05<br>If NOW is specified or the option is omitted, then the job is scheduled immediately.<br>If PAUSE is specified, then the job starts in the paused state and you need to resume the job using the rhpctl resume job -jobid *job_id* command. |
| -jobtag *tag_name* | Optionally, you can associate a user-defined tag with the scheduled jobs. |

### Usage Notes

The working copy can be on Fleet Patching and Provisioning Server, a Fleet Patching and Provisioning Client, or an rhpclient-less target.

### Example

The following example creates a PDB called `pdb183` on a CDB called `raccdb183` on a working copy called `wc_db183`:

```
$ rhpctl addpdb database -workingcopy wc_db183 -cdbname raccdb183 -pdbName
pdb183
```

## rhpctl deletepdb database

Deletes a Pluggable Database (PDB) to an existing Container Database (CDB) on a working copy.

### Syntax

```
rhpctl deletepdb database -workingcopy workingcopy_name -cdbname cdb_name -
pdbName pdb_name
  [-root | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]
   | -sudouser user_name -sudopath sudo_binary_path] [-targetnode node_name]
  [-useractiondata user_action_data] [-schedule {timer_value | NOW | PAUSE}]
[-jobtag tag_name]
```

### Parameters

**Table A-20 rhpctl deletepdb database Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of an existing working copy for the PDB that you want to delete. |
| `-cdbname cdb_name` | Specify the name of the multitenant CDB from which you want to delete the PDB. |
| `-pdbName pdb_name` | Specify the name of the PDB you want to delete. |
| `-root \| -cred cred_name \| -auth plugin_name [-arg1 name1:value1...] \| -sudouser user_name -sudopath sudo_binary_path` | If you choose to use the `-targetnode` parameter, then you must choose either `sudo` or `root` to access the remote node.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-targetnode node_name` | Optionally, you can specify the name of a node in the cluster on which you want to run this operation. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |

**Table A-20 (Cont.) rhpctl deletepdb database Command Parameters**

| Parameter | Description |
|---|---|
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br><br>`2018-07-25T19:13:17+05`<br><br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |

### Usage Notes

The working copy can be on Fleet Patching and Provisioning Server, a Fleet Patching and Provisioning Client, or an `rhpclient`-less target.

### Examples

The following example deletes a PDB called `pdb183` from a CDB called `raccdb183` on a working copy called `wc_db183`:

```
$ rhpctl deletepdb database -workingcopy wc_db183 -cdbname raccdb183 -pdbName
pdb183
```

## rhpctl delete database

Deletes a database that was created on a working copy.

> **Note:**
>
> If the database is hosted on a working copy that is on the Oracle Fleet Patching and Provisioning Server or on an Oracle Fleet Patching and Provisioning Client, then credentials are not required. This is true whether you run the command on the Server or the Client. Credentials are required when you run the command on the Server and the working copy is on an `rhpclient`-less target.

### Syntax

```
rhpctl delete database -workingcopy workingcopy_name -dbname unique_db_name
  {-sudouser sudo_user_name -sudopath sudo_binary_path |
  -root | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]}
  [-targetnode node_name] [-useractiondata user_action_data]
  [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name]
```

### Parameters

**Table A-21 rhpctl delete database Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy` *workingcopy_name* | Specify a name for the working copy for the database that you want to delete. |
| `-dbname` *unique_db_name* | Specify the unique name of the database (`DB_UNIQUE_NAME`without `DB_DOMAIN`) that you are deleting. |
| `-sudouser` *user_name* `-sudopath` *sudo_binary_path* `\| -root \| -cred` *cred_name* `\| -auth` *plugin_name* `[-arg1` *name1*`:`*value1*`...]` | If you choose to use the `-targetnode` parameter, then you must choose either `sudo` or `root` to access the remote node.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-targetnode` *node_name* | Optionally, you can specify the name of an `rhpclient`-less target. |
| `-useractiondata` *user_action_data* | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-schedule {`*timer_value* `\| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid` *job_id* command. |
| `-jobtag` *tag_name* | Optionally, you can associate a user-defined tag with the scheduled jobs. |

## rhpctl deletenode database

Deletes an instance, which contracts an administrator-managed Oracle RAC database.

### Syntax

```
rhpctl deletenode database -workingcopy working_copy_name -dbname
unique_db_name
 -node node_list {-root | -sudouser sudo_username -sudopath sudo_binary_path
 | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]} [-force]
 [-failover] [-drain_timeout timeout] [-stopoption stop_option] [-
useractiondata user_action_data]
 [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name] [-eval]
```

### Parameters

**Table A-22 rhpctl deletenode database Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy working_copy_name` | Specify the name of a working copy. |
| `-dbname unique_db_name` | Specify the unique name of the database (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you are deleting. |
| `-node node_list` | Specify a node or comma-delimited list of several nodes from which to delete the database instance. |
| `-root \| -sudouser sudo_username -sudopath sudo_binary_path \| -cred cred_name \| -auth plugin_name [-arg1 name1:value1...]` | Choose either `sudo` or `root` to access the remote nodes.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-force` | Use `-force` to remove the instance after forcibly stopping the instance. |
| `-failover` | Optionally, you can use this parameter to attempt to have services running on the instance that want to delete fail over to another instance. |
| `-drain_timeout timeout` | Optionally, you can use `-drain_timeout` to specify the time, in seconds, allowed for resource draining to be completed. Accepted values are an empty string (""), 0, or any positive integer. The default value is an empty string, which means that this parameter is not set. If it is set to 0, then draining occurs, immediately.<br>The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. |
| `-stopoption stop_option` | Optionally, you can specify a stop option for the database. Options include: `ABORT`, `IMMEDIATE`, `NORMAL`, `TRANSACTIONAL`, and `TRANSACTIONAL_LOCAL`. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-eval` | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |

### Usage Notes

- If the working copy is on a Fleet Patching and Provisioning Client or on the Fleet Patching and Provisioning Server, then credentials are not required. This is true whether you run the command on the Server or the Client. Credentials are required when you run the command on the Server and the working copy is on an `rhpclient`-less target.
- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

## rhpctl move database

Moves one or more databases from a source working copy or any Oracle Database home to a patched working copy.

### Syntax

```
rhpctl move database -patchedwc workingcopy_name
   {{-sourcewc workingcopy_name | -sourcehome Oracle_home_path [-oraclebase Oracle_base_path]
   [-client cluster_name]} [-dbname db_name_list | -excludedblist db_name_list]
   [-nonrolling [-skipprereq] | -forcerolling | -batches list_of_batches | -smartmove [-saf
availability] [-stopbetweenbatches]]]
   [-eval] [-ignoremissingpatches patch_name1 [,patch_name2...]] [-ignorewcpatches] [-
keepplacement]
   [-disconnect [-noreplay]] [-drain_timeout time] [-stopoption stop_option] [-nodatapatch]
[-skipdatapatchcheck]
   [-targetnode node_name] [-notify [-cc user_list]] | -continue [-skip] | -revert | -abort}
   [-root | -cred cred_name | -sudouser sudo_user_name -sudopath sudo_binary_location |
   -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
   [-schedule {timer_value | NOW | PAUSE}] [-chainbatches] [-pausebetweenbatches] [-jobtag
tag_name]
   [-tags tags_criteria]    [-useractiondata user_action_data] [-dbsinparallel
<number_of_instances>] [-raconetimeout <timeout>]
```

### Parameters

**Table A-23 rhpctl move database Command Parameters**

| Parameter | Description |
|---|---|
| `-patchedwc` *workingcopy_name* | Specify the name of the working copy to where you want to move the database. |
| `-sourcewc` *workingcopy_name* | Specify the name of the working copy from which the database is to be moved. |
| `-sourcehome` *Oracle_home_path* | Alternatively, you can specify the source Oracle home path. |
| `-oraclebase` *Oracle_base_path* | Specify the *ORACLE_BASE* path for provisioning the Oracle database home (required only for `ORACLEDBSOFTWARE` image type). |
| `-client` *cluster_name* | Specify the name of the client cluster. |

**Table A-23 (Cont.) rhpctl move database Command Parameters**

| Parameter | Description |
|---|---|
| `-dbname db_name_list` | Specify the unique names of the databases (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you want to move to the patched working copy.<br>Specify `-dbname ALL-DBS` to move all the databases from the source home to the destination home. To exclude any databases from the move operation, use `-excludedblist`.<br>**Note:**<br>If you are moving a non-clustered (single-instance) database, then, for the value of the `-dbname` parameter, you must specify the *SID* of the database instead of the database name. |
| `-excludedblist db_name_list` | Alternative to using the `-dbname` parameter, you can use the `-excludedblist` parameter to patch all databases except specific databases. |
| `-nonrolling [-skipprereq] \| -forcerolling \| -batches list_of_batches \| -smartmove [-saf availability] [-stopbetweenbatches]` | Optionally, you can choose one of the three following methods to move a database:<br>• Use the `-nonrolling` parameter to move the database in a non-rolling mode. By default, databases move in a rolling mode. Use the `-skipprereq` option to skip the prerequisite checks and start the database in upgrade mode for patching.<br>• Use the `-forcerolling` parameter to force the Oracle home to move in rolling mode.<br>• Use the `-batches` parameter to specify a comma-delimited list of batches of nodes (where each batch is a comma-delimited list of node names enclosed in parentheses) enclosed in double quotation marks ("") in the format: `"(nA,nB,...),(...,nY,nZ)"`.<br>• Alternatively, use the `-smartmove` parameter. Use the `-saf availability` parameter to specify a service availability factor, which is the minimum percentage of instances on which a service must remain running during the move.<br>Use the `-stopbetweenbatches` parameter to process batches separately. Smart move operation pauses for user intervention before continuing with the next batch. |
| `-eval` | Use the `-eval` parameter to print auto-generated batches of nodes and sequence of moves without actually performing the move operation. |
| `-ignoremissingpatches patch_name1 [,patch_name2...]` | Perform the move and/or upgrade even though the specified patches, which are present in the source path or working copy, may be missing from the destination path or working copy. |
| `-ignorewcpatches` | Optionally, you can use this parameter to ignore if a patched working copy is missing some patches which are present in the source path or working copy. |
| `-keepplacement` | Use this parameter to ensure that services of administrator-managed Oracle RAC or Oracle RAC One Node databases are running on the same instances before and after the move operation. |
| `-disconnect [-noreplay]` | Optionally, use the `-disconnect` parameter to disconnect all sessions before stopping or relocating services. If you choose to use `-disconnect`, then you can choose to use the `-noreplay` parameter to disable session replay during disconnection. |

**Table A-23 (Cont.) rhpctl move database Command Parameters**

| Parameter | Description |
|---|---|
| `-drain_timeout` *timeout* | Specify the time, in seconds, allowed for resource draining to be completed. Accepted values are an empty string (""), 0, or any positive integer. The default value is an empty string, which means that this parameter is not set. If it is set to 0, then draining occurs, immediately.<br>The draining period is intended for planned maintenance operations. During the draining period, all current client requests are processed, but new requests are not accepted. |
| `-stopoption` *stop_option* | Optionally, you can specify a stop option for the database. Options include: `ABORT`, `IMMEDIATE`, `NORMAL`, `TRANSACTIONAL`, and `TRANSACTIONAL_LOCAL`. |
| `-nodatapatch` | Use this parameter to indicate not to run `datapatch` for databases you are moving. |
| `-skipdatapatchchecks` | Use this parameter to skip executing datapatch sanity checks. |
| `-targetnode` *node_name* | Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client. |
| `-notify [-cc` *user_list*`]` | Optionally, you can supply a list of users to whom email notifications of the move will be sent, in addition to the owner of the working copy. |
| `-continue [-skip]` | If a batch-mode `rhpctl move database` command fails at any point, then, after correcting the cause of the error, you can rerun the command with the `-continue` parameter to attempt to patch the failed batch. If you want to skip the failed batch and continue with the next batch, use the `-continue` and `-skip` parameters together. If you attempt to skip over the last batch, then the move operation is terminated. |
| `–revert` | If a batch-mode or non-batch-mode `rhpctl move database` command fails, then you can rerun the command with the `-revert` parameter to undo the changes that have been made, and return the configuration to its initial state. |
| `–abort` | If a batch-mode or non-batch-mode `rhpctl move database` command fails, then you can rerun the command with the `-abort` parameter to terminate the patching process and leave the cluster in its current state. |
| `-root` \| `-cred` *cred_name* \| `-sudouser` *sudo_user_name* `-sudopath` *sudo_binary_location* \| `-auth` *plugin_name plugin_args* | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-useractiondata` *user_action_data* | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |

**Table A-23 (Cont.) rhpctl move database Command Parameters**

| Parameter | Description |
|---|---|
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br><br>`2018-07-25T19:13:17+05`<br><br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command.<br><br>**Note:**<br>If the `-schedule` parameter is used with the `-batches` parameter, then the command stops after the first batch and you have to use the `-continue` parameter to run the next batch. A new job ID is generated for every batch processing. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-chainbatches` | Use this parameter to run the command on all batches without pausing after each batch. When this parameter is used, you do not have to use the `-continue` parameter after the processing is completed on each batch. |
| `-pausebetweenbatches` | Use this parameter to pause between two batches, which you can rerun using the `rhpctl resume -job` command. If this parameter is used, the all the batches are run using the same job ID. |
| `-tags tags_criteria` | Optionally, specify tags criteria to create child jobs internally in the PAUSED state. Use the `rhpctl resume job` command to continue the operations of all the child jobs. |
| `-dbsinparallel number_of_instances` | Number of database instances that can be started in parallel on a given node. |
| `-raconetimeout timeout` | RAC One Node database relocation timeout in minutes. |

### Usage Notes

- You can obtain context sensitive help for specific use cases for the `rhpctl move database` command, as follows:

```
$ rhpctl move database -help [EXISTING_PATCHEDWC | NEW_PATCHEDWC | SRCHOME
  | SINGLEINSTANCEDB | ROLLING | NONROLLING | BATCHES | SMARTMOVE]
```

- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

### Examples

To move all the databases running from one working copy to another in a rolling fashion:

```
$ rhpctl move database -sourcewc prodHomeV1 -patchedwc prodHomeV2 -client
prodcluster
```

In the preceding example, the patched working copy, `prodHomeV2`, must exist.

To move all databases running on a non-managed Oracle home at `/u01/app/product/12.1.0/dbhome` to a working copy named `myDB12Home1`:

```
$ rhpctl move database -sourcehome /u01/app/product/12.1.0/dbhome -
oraclebase /u01/app/product/12.1.0/obase -patchedwc myDB12Home1
```

To move a database named `SampleDB` from a working copy named `myDB12Home1` to a working copy named `myDB12Home1patched` (any other databases running on `myDB12Home1` are not affected by this move):

```
$ rhpctl move database -sourcewc myDB12Home1 -patchedwc myDB12Home1patched -
dbname SampleDB
```

To move all databases running on a working copy named `myDB12Home1` to a working copy named `myDB12Home1patched`:

```
$ rhpctl move database -sourcewc myDB12Home1 -patchedwc myDB12Home1patched
```

To move a non-clustered (single-instance) database with a SID of `SID101` running on a patched working copy named `myDB12Home1patched`:

```
$ rhpctl move database -patchedwc myDB12Home1patched -sourcehome
/u01/app/oracle/product/12.2.0/db_1 -targetnode vm02 -dbname SIDl01
-sudouser oracle -sudopath /usr/bin/sudo
```

The preceding examples are the basic form of the command. You can also move groups of databases in batches. The batch operations also support management of session connections and recovery options.

## rhpctl movepdb database

Moves specified Pluggable Databases (PDBs) from the source multitenant Container Database (CDB) to the destination multitenant CDB.

**Syntax**

```
rhpctl movepdb database -sourcecdb source_cdb_name -destcdb
destination_cdb_name
    {-pdbName pdb_name_list | -excludepdblist pdb_name_list}
    [-root | -cred cred_name | -sudouser sudo_user_name
     -sudopath sudo_binary_location |
     -auth plugin_name [-arg1 name1:value1
     [-arg2 name2:value2 ...]]] [-client client_name | -targetnode node_name]
    [-useractiondata user_action_data] [-schedule {timer_value | NOW |
PAUSE}] [-jobtag tag_name]
```

### Parameters

**Table A-24 rhpctl movepdb database Command Parameters**

| Parameter | Description |
|---|---|
| `-sourcecdb` *source_cdb_name* | Specify the name of the Oracle Multitenant CDB from which you want to move the PDB. |
| `-destcdb` *destination_cdb_name* | Specify the name of the multitenant CDB to which you want to move the PDB. |
| `-pdbName` *pdb_name_list* | Specify a comma-separated list of names of PDBs that you want to move. |
| `-excludepdblist` *pdb_name_list* | Specify a list of PDBs that you want to excluded from the move operation. |
| `-root \| -cred` *cred_name* `\| -sudouser` *sudo_user_name* `-sudopath` *sudo_binary_location* `\| -auth` *plugin_name plugin_args* | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-client` *client_name* `\| -targetnode` *node_name* | Optionally, you can specify either the name of the client cluster or the node on which the operation is to be run. |
| `-useractiondata` *user_action_data* | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-schedule {`*timer_value* `\| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid` *job_id* command. |
| `-jobtag` *tag_name* | Optionally, you can associate a user-defined tag with the scheduled jobs. |

### Usage Notes

- The source and destination CDBs can be running in a provisioned database working copy. The working copy can be on the Oracle Fleet Patching and Provisioning Server, an Oracle Fleet Patching and Provisioning Client, or an `rhpclient`-less target. The destination CDB can be at a higher patch level, which facilitates patching of a PDB to a higher patch level.

- You can only use this command if both the source and destination CDBs are on the same node.
- This command does not currently support for Oracle RAC databases.

**Examples**

To move a PDB from a source CDB to a destination CDB:

```
rhpctl movepdb database -sourcecdb srccdb -pdbName pdb1,pdb2,pdb3 -destcdb
dstcdb
```

## rhpctl upgrade database

Upgrades a database to the version of the destination working copy.

**Syntax**

```
rhpctl upgrade database [-sourcewc source_workingcopy_name | -sourcehome
oracle_home_path
   [-oraclebase Oracle_base_path] [{-client cluster_name | -targetnode
node_name}]]
   [-root | -cred cred_name | -sudouser sudo_username -sudopath
path_to_sudo_binary
   | -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
   -destwc destination_workingcopy_name [-image image_name [-path where_path]]
   -dbname unique_db_name [-useractiondata user_action_data] [-eval [-preupg]
   [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name]
   [-ignoremissingpatches patch_name1[,patch_name2...]
   [-dbuaargs dbua_arguments] [-method {AUTOUPG | DBUA}] [-upgtimezone { YES
| NO }]
   [-runutlrp { YES | NO }] [-grp {YES | NO}] [-tgtcdbname cdb_unique_name [-
tgtpdbname new_pdb_name]]
   [-fra db_recovery_file_dest] [-ignoregroupcheck]
   [-drain_timeout session_drain_time] [-abort] [-revert]
```

**Parameters**

**Table A-25 rhpctl upgrade database Command Parameters**

| Parameter | Description |
|---|---|
| `-sourcewc source_workingcopy_name` | Specify the name of the source working copy from which you want to upgrade the database. |
| `-sourcehome oracle_home_path` | Alternative to specifying the name of the source working copy, you can specify the path to the source Oracle home. |
| `-oraclebase oraclebase_path` | If you use the `-sourcehome` parameter, then you can, optionally, specify a different `ORACLE_BASE` from the source Home. |
| `-client cluster_name \| -targetnode node_name` | Specify either the name of the Oracle FPP Client cluster or the name of an `rhpclient`-less target on which to provision a working copy. |

**Table A-25 (Cont.) rhpctl upgrade database Command Parameters**

| Parameter | Description |
|---|---|
| -root \| -cred *cred_name* \| -sudouser *sudo_user_name* -sudopath *sudo_binary_location* \| -auth *plugin_name plugin_args* | If you choose to use the -targetnode parameter, then you must choose either root, a credential name, sudo, or an authentication plugin to access the remote node.<br>Choose -root to perform super user operations as root. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a sudo user by specifying a sudo user name and the path to the sudo binary, or to use an authentication plugin to access the remote node. |
| -destwc *destination_workingcopy_name* [-image *image_name* [-path *where_path*]] | Specify the name of the destination working copy to which the database is to be upgraded. If the destination working copy does not exist, then specify the gold image from which to create it, and optionally, the path to where to provision the working copy. |
| -dbname *unique_db_name* | Specify the name of the database you are upgrading. |
| -useractiondata *user_action_data* | Optionally, you can pass a value to the useractiondata parameter of the user action script. |
| -eval | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |
| -schedule {*timer_value* \| NOW \| PAUSE} | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>2018-07-25T19:13:17+05<br>If NOW is specified or the option is omitted, then the job is scheduled immediately.<br>If PAUSE is specified, then the job starts in the paused state and you need to resume the job using the rhpctl resume job -jobid *job_id* command. |
| -jobtag *tag_name* | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| -ignoremissingpatches | Performs the move and/or upgrade although the specified patches, which are present in the source path or working copy, could be missing from the destination path or working copy. |
| -dbuaargs *dbua_arguments* | Species a double-quoted string containing optional arguments for DBUA. |
| -method {AUTOUPG \| DBUA} | Performs the database upgrade operation using the AutoUpgrade or Oracle DBUA. The default method is AUTOUPG. |
| -upgtimezone {YES \| NO} | Enables or disables time zone upgrade as part of the AutoUpgrade process. Default is YES. |
| -runutlrp {YES \| NO} | Enables or disables the recompilation of invalid objects as part of the AutoUpgrade process. Default is YES. |
| -grp {YES \| NO} | AutoUpgrade generates a Guaranteed Restore Point (GRP) for restoring the database. Default is YES. |
| -tgtcdbname *cdb_unique_name* | Specifies DB_UNIQUE_NAME of the target CDB in which you want to plug-in the non-CDB Oracle database. |
| -tgtpdbname *new_pdb_name* | Specifies a new name to assign to the non-CDB source database when it is plugged in as a PDB in the target CDB. |
| -fra *db_recovery_file_dest* | Specifies the location of the flash recovery area. |
| -ignoregroupcheck | Skips the group check except for OSDBA and OSASM during the upgrade database process. |

**Table A-25 (Cont.) rhpctl upgrade database Command Parameters**

| Parameter | Description |
|---|---|
| `-drain_timeout` *`session_drain_time`* | Specifies the service drain timeout in seconds. |
| `-abort` | Terminate the ongoing upgrade operation. |
| `-revert` | Use this parameter to revert the failed upgrade of Oracle Database. |

### Usage Notes

If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

### Example

The following example upgrades a database, `testy`, from Oracle Database 11*g*, which is on working copy `db112mbc143` to Oracle Database 12*c*, which is on working copy `db12102mbc143`, both of which reside on the remote node `bposvr141`:

```
$ rhpctl upgrade database -dbname testy -sourcewc db112mbc143 -destwc
db12102mbc143 -root -targetnode bposvr141
```

## rhpctl zdtupgrade database

The Zero Downtime Upgrade command `rhpctl zdtupgrade database` enables zero downtime database upgrades for Oracle RAC and Oracle RAC One Node Oracle Database instances.

> **Note:**
>
> The Zero Downtime Upgrade (ZDU) feature of Oracle Fleet Patching and Provisioning (FPP) is deprecated in Oracle Database 23ai.

### Syntax

```
rhpctl zdtupgrade database -dbname unique_db_name -destwc
destination_workingcopy_name
   [converttopdb -cdbname cdb_name
         [-pdbName pdb_name]]
   [-sourcewc source_workingcopy_name | -sourcehome oracle_home_path]
   -ggsrcwc golden_gate_source_workingcopy_name
   -ggdstwc golden_gate_dest_workingcopy_name
   [-clonedatadg diskgroup_name
         [-cloneredodg diskgroup_name]
         [-clonerecodg diskgroup_name]|
   -clonedatafs acfs_mountpoint
         [-cloneredofs acfs_mountpont]
         [-clonerecofs acfs_mountpoint]]
   [-targetnode node_name
         {-root |
          -cred credential name|
```

```
        -sudouser sudouser_name -sudopath sudo_binary_location|
        -auth plugin_name
              [arg1 name1:value1
                    [arg2 name2:value2. . . ]]}]
  [-eval]
  [-ignoreprereq]
  [-useractiondata user_action_data]
  [-dbuaargs dbua_arguments
```

### Parameters

**Table A-26 rhpctl zdtupgrade database Command Parameters**

| Parameter | Description |
|---|---|
| `-dbname` *unique_db_name* | Specify the unique name of the database that you want to upgrade. |
| `-converttopdb -cdbname` *cdbname* `-pdbName` *pdb_name* | (Optional) Specifies that you want to convert a non-container (non-CDB) Oracle Database to a Pluggable Database (PDB) during the upgrade.<br>If you select this option, then provide the following arguments:<br>`-cdbname` *cdb_name*, which defines the name of the existing Container Database (CDB) on which you want to place the PDB<br>`-pdbName` *pdb_name*, which defines the name of the PDB that you want to plug in to the CDB.<br>After the non-CDB source database is upgraded on the destination node, if you specify `-converttopdb`, and provide the existing CDB's name, then Zero Downtime Upgrade performs the conversion. |
| `-destwc` *destination_workingcopy_name* | Specifies the name of the destination working copy to which the database is to be upgraded. |
| `-sourcewc` *source_workingcopy_name* | (Optional) You can specify the name of the source working copy from which you want to upgrade the database. |
| `-sourcehome` *oracle_home_path* | Alternative to specifying the name of the source working copy, you can specify the path to the source Oracle home. |
| `-ggsrcwc` *golden_gate_source_workingcopy_name* | Specify the name of the Oracle GoldenGate source working copy. |
| `-ggdstwc` *golden_gate_dest_workingcopy_name* | Specify the name of the Oracle GoldenGate destination working copy. |
| `-clonedatadg` *diskgroup_name* | (Optional) You can specify the name of an Oracle ASM disk group to use as a data file location for the cloned database. |
| `-cloneredodg` *diskgroup_name* | (Optional) You can specify the name of an Oracle ASM disk group to use as a redo log location for the cloned database. |
| `-clonerecodg` *diskgroup_name* | (Optional) You can specify the name of an Oracle ASM disk group to use as a recovery area for the cloned database. |
| `clonedatafs` *acfs_mountpoint* | (Optional) You can specify the mount point of an Oracle Advanced Cluster File System (Oracle ACFS) to use as a data file location for the cloned database. |
| `-cloneredofs` *acfs_mountpoint* | (Optional) You can specify the name of an Oracle ACFS file system to use as a redo log location for the clone database. |

**Table A-26 (Cont.) rhpctl zdtupgrade database Command Parameters**

| Parameter | Description |
|---|---|
| `-clonerecofs` *acfs_mountpoint* | (Optional) You can specify the name of an Oracle ACFS file system to use as a recovery area for the clone database. |
| `-rmanlocation` *backup_location* | (Optional) You can specify the source RMAN backup location. |
| `-sourcenode` *node_name* | (Optional) You can specify the name of a source node in a remote cluster with no Fleet Patching and Provisioning Client on which to provision a working copy, where the source database is running.<br>If you choose to use the `-sourcenode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-targetnode` *node_name*<br>`-root \| -cred` *cred_name* `\| -sudouser` *sudo_user_name* `-sudopath` *sudo_binary_location* `\| -auth` *plugin_name plugin_args* | (Optional) You can specify the name of an `rhpclient`-less target on which to provision a working copy, where you want to run the upgraded database.<br>If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `eval` | (Optional) You can specify `eval` to evaluate the Zero Downtime Upgrade operation to see if it can succeed, but does not perform the operation. |
| `-ignoreprereq` | (Optional) You can use this parameter to instruct the `zdtupgrade database` command to ignore system prerequisites during the upgrade. |
| `-useractiondata` *user_action_data* | (Optional) You can use this parameter to specify a value to be passed to the `useractiondata` parameter of a user action script |
| `-dbuaargs` | (Optional) If you do not specify the AutoUpgrade Utility for the upgrade with the `-autoupg` parameter, so that Database Upgrade Assistant (DBUA) is used for the upgrade, then you can specify arguments to pass to DBUA. If you specify `-autoupg`, then this argument is not available.<br>For example, if the user account with which you are running Zero Downtime Upgrade If your account does not have SYSDBA privileges, or you do not have operating system authentication set up, then you can use the following syntax to connect, where *mydb* is your Oracle Database SID, *username* is a user name with SYSDBA privileges, and *password* is that user name's password:<br>`-sysDBAUserName - username -sysDBAPassword - password` |

**Table A-26 (Cont.) rhpctl zdtupgrade database Command Parameters**

| Parameter | Description |
|---|---|
| `[ -autoupg [ -upgtimezone [yes\|no] [-runutlrp [yes\|no] ]` | (Optional) Specifies that you want to use the AutoUpgrade utility for the upgrade. If you select this option, then provide the following arguments:<br>`upgtimezone [yes\|no]` Specify `yes` to upgrade the timezone during the upgrade, or `no` to skip the timezone upgrade.<br>If the Pre-Upgrade Information Tool instructs you to upgrade the time zone files after completing the database upgrade, then you can select this argument to enable the Zero Downtime Upgrade to upgrade the time zone file after the upgrade completes.<br>`runutlrp [yes\|no]` Specify `yes` to run the `utlrp.sql` script after the upgrade to revalidate the packages again., or select `no` to skip the timezone upgrade.<br>Oracle recommends that you run the `utlrp.sql` script after you install, patch, or upgrade a database, to identify and recompile invalid objects. |

# datapatch Commands

Use commands with the `datapatch` keyword to apply patches to the specified Oracle Database.

- rhpctl apply datapatch
  Applies datapatch to the specified Oracle Database.

## rhpctl apply datapatch

Applies datapatch to the specified Oracle Database.

### Syntax

```
rhpctl apply datapatch -workingcopy workingcopy_name [-targetnode node_name]
  [-root | -cred cred_name | -sudouser sudo_user_name -sudopath sudo_binary_location |
  -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
  [-dbname db_name_list] [-excludedblist db_name_list] [-eval] [-schedule {timer_value | NOW
| PAUSE}]
  [-jobtag tag_name]
```

### Parameters

**Table A-27 rhpctl apply datapatch Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of the working copy to which you want to apply the datapatch. |
| `-targetnode node_name` | Optionally, you can specify the name of a node in a remote cluster that has no Fleet Patching and Provisioning Client. |

**Table A-27 (Cont.) rhpctl apply datapatch Command Parameters**

| Parameter | Description |
|---|---|
| `-root` \| `-cred` *cred_name* \| `-sudouser` *sudo_user_name* `-sudopath` *sudo_binary_location* \| `-auth` *plugin_name plugin_args* | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-dbname` *db_name_list* | Specify the unique names of the databases (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you want to move to the patched working copy.<br>**Note:** If you are moving a non-clustered (single-instance) database, then, for the value of the `-dbname` parameter, you must specify the *SID* of the database instead of the database name. |
| `-excludedblist` *db_name_list* | Alternative to using the `-dbname` parameter, you can use the `-excludedblist` parameter to patch all databases except specific databases. |
| `-eval` | Use the `-eval` parameter to print auto-generated batches of nodes and sequence of moves without actually performing the move operation. |
| `-schedule {`*timer_value* \| `NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid` *job_id* command. |
| `-jobtag` *tag_name* | Optionally, you can associate a user-defined tag with the scheduled jobs. |

# exadata Commands

Use commands with the `exadata` keyword to patch an Oracle Exadata system.

- rhpctl update exadata
  Patches an Oracle Exadata system.

## rhpctl update exadata

Patches an Oracle Exadata system.

**Syntax**

```
rhpctl update exadata {-dbnodes comma_separates_list_of_nodes {[-patchmgrloc
patch_mgr_loc] [-iso_repo iso_repo_name]
[-backup] [-batches list_of_batches | -continue | -abort] [-liveupdate
{ highcvss | allcvss | full } ] } | -cells list_of_cell_nodes [-rollback] |
-ibswitches -list_of_ibswitch_nodes [-path ibswitch_vm_path] | -roceswitches
list_of_roceswitches [-downgrade]}
[-image exadata_image_name] [-fromnode node_name] [-patchmgargs "-
patch_mgr_args"] [-client client_name | -server]
[-smtpfrom "email_address"] [-smtpto "email_address"] [-schedule {timer_value
```

```
| NOW | PAUSE} [-jobtag job_tag_name]
[-patchmgrdrivingsystem patchmgr_driving_node] [-user user_name] [-
useractiondata user_action_data] [-eval]
```

### Parameters

**Table A-28 rhpctl update exadata Command Parameters**

| Parameter | Description |
|---|---|
| `-dbnodes` | Comma-delimited list of Oracle Exadata database nodes. |
| `-patchmgrloc patch_mgr_loc` | Specifies the patch manager location. |
| `-iso_repo iso_image_name` | Specifies the ISO image name. |
| `-backup` | Performs backup of the Oracle Exadata database server. |
| `-batches list_of_batches` | Optionally, you can specify a comma-delimited list of batches of nodes where each batch is a comma-delimited list of node names enclosed in parentheses and node names are enclosed in double quotation marks ("") in the format: "`(nA,nB,...),(...,nY,nZ)`". |
| `-continue` | Update Oracle Exadata on the next batch of nodes. |
| `-abort` | Stop the ongoing update operation. |
| `-liveupdate { highcvss \| allcvss \| full }` | Perform live update of the Oracle Exadata database node. When using Exadata Live Update, you must choose from the following options:<br>• `highcvss`: Performs only critical security updates to address vulnerabilities with a Common Vulnerability Scoring System (CVSS) score of 7 or greater.<br>• `allcvss`: Performs only security updates to address vulnerabilities with a CVSS score of 1 or greater.<br>• `full`: Performs a full update, which includes all security-related updates and all other non-security updates. |
| `-cells list_of_cell_nodes` | Comma separated list of storage server cell nodes. |
| `-rollback` | Specify this option to roll back the patch. |
| `-ibswitches list_of_ibswitch_nodes` | Specifies to patch the InfiniBand Network Fabric switches. |
| `-path ibswitch_vm_path` | Specify the absolute path for deploying image for upgrading or downgrading InfiniBand switches in a VM environment. |
| `-roceswitches list_of_roceswitches` | Specifies to patch the RDMA over Converged Ethernet (RoCE) switches. |
| `-downgrade` | Specify this option to downgrade the InfiniBand or RoCE switches. |
| `-image` | Specify the name of an Oracle Exadata image.<br>This image should have already been deployed to all the database nodes of the destination machine. |
| `-fromnode` | Specify the name of the source node. |
| `-patchmgargs` | Optionally specify the patch manager arguments in double quoted string. |
| `-client client_name` | Specify the name of the client cluster on which you want to update Oracle Exadata components. |
| `-server server_name` | Specify the name of the Oracle FPP Server from which you want to update the Oracle Exadata components. |
| `-smtpfrom` | The email address from which you want to send patch manager notifications. |
| `-smtpto` | The email address to which you want to send patch manager notifications. |

**Table A-28 (Cont.) rhpctl update exadata Command Parameters**

| Parameter | Description |
|---|---|
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br><br>`2018-07-25T19:13:17+05`<br><br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command.<br><br>**Note:**<br>If the `-schedule` parameter is used with the `-batches` parameter, then the command stops after the first batch and you have to use the `-continue` parameter to run the next batch. A new job id is generated for every batch operation. |
| `-jobtag job_tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-patchmgrdrivingsystem patchmgr_driving_node` | Oracle Exadata system node from which you want to start the patch manager.<br><br>**Note:**<br>Specify this option only when patching the Oracle Exadata InfiniBand switch. |
| `-user user_name` | Specify name of the operating system user for whom you are provisioning the software home. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-eval` | Evaluate the command without actually running the command. |

### Usage Notes

Use the `rhpctl update exadata` command to perform only database node patching.

### Example

The following example performs database node patching on a client cluster.

```
$ rhpctl update exadata image -image EXADATAIMAGEV1
     -iso_repo p28802055_192000_Linux-x86-64.zip -patchmgrloc /patchMgr/
dbserver_patch_19.190306
     -patchmgrargs "-ignore_alerts" -client CLIENT1 -batches "(rac07box1)"
```

> **See Also:**
>
> Combined Oracle Exadata Database Server and Grid Infrastructure Update for information about Oracle Exadata database node patching

# gihome Commands

Use commands with the `gihome` keyword to add or delete nodes to Oracle Grid Infrastructure home and, move and upgrade Oracle Grid Infrastructure home.

- rhpctl addnode gihome
- rhpctl deletenode gihome
- rhpctl move gihome
  Moves the Oracle Grid Infrastructure software stack from one home to another.
- rhpctl upgrade gihome
  Upgrades the Oracle Grid Infrastructure from a source working copy or source home path to a destination working copy.

## rhpctl addnode gihome

Adds one or more nodes to an Oracle Grid Infrastructure installation.

### Syntax

```
rhpctl addnode gihome {-workingcopy workingcopy_name | -client cluster_name}
  -newnodes node_name:node_vip[,node_name:node_vip...]
  {-root | -cred cred_name | -sudouser sudo_user_name
    -sudopath sudo_binary_location |
    -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]}
  [-targetnode node_name] [-force] [-setupssh] [-useractiondata
user_action_data]
  [-eval] [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name]
```

### Parameters

**Table A-29 rhpctl addnode gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of the working copy of the active Oracle Grid Infrastructure home that you want to install and configure on the specified node. |
| `-client cluster_name` | Alternatively, you can specify the name of the client cluster to which to add cluster nodes. |
| `-newnodes node_name:node_vip` | Specify a comma-delimited list of nodes on which Oracle Clusterware will be provisioned in the following format: `node_name:node_vip` |

**Table A-29 (Cont.) rhpctl addnode gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-root \| -cred` *`cred_name`* `\| -sudouser` *`sudo_user_name`* `-sudopath` *`sudo_binary_location`* `\| -auth` *`plugin_name plugin_args`* | You must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-targetnode` *`node_name`* | Optionally, you can specify the name of an `rhpclient`-less target. |
| `-force` | Optionally, you can use this parameter to forcibly add nodes ignoring any previously failed add-node operation. |
| `-setupssh` | Sets up passwordless SSH user equivalence on the remote nodes for the provisioning user. |
| `-useractiondata` *`user_action_data`* | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-eval` | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |
| `-schedule {`*`timer_value`*`\| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid` *`job_id`* command. |
| `-jobtag` *`tag_name`* | Optionally, you can associate a user-defined tag with the scheduled jobs. |

### Usage Notes

- You can specify the `rhpclient`-less target for the operation using the working copy name or, if the destination cluster is a Oracle Fleet Patching and Provisioning Client, then using the client cluster name.
- You must provide either `root` credentials, a credential name, a `sudo` user, or an authentication plugin.
- A destination node is required if the destination cluster is an Oracle Clusterware 11*g* release 2 (11.2) or 12*c* release 1 (12.1) cluster and must be the node name of an existing cluster node.

## rhpctl deletenode gihome

Removes one or more nodes from an Oracle Grid Infrastructure installation.

### Syntax

```
rhpctl deletenode gihome {-workingcopy workingcopy_name | -client
cluster_name}
  -node node_list {-root | -sudouser sudo_username -sudopath sudo_binary_path
    -cred cred_name | -auth plugin_name [-arg1 name1:value1...]}
 [-targetnode node_name] [-useractiondata user_action_data]
 [-eval] [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name]
```

### Parameters

**Table A-30 rhpctl deletenode gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of a working copy of the Oracle Grid Infrastructure home that you want to remove from the specified node. |
| `-client cluster_name` | Alternatively, you can specify the name of the client cluster from which to remove cluster nodes. |
| `-node node_list` | Specify a comma-delimited list of node names from which to delete Oracle Grid Infrastructure. |
| `-root \| -sudouser sudo_username -sudopath sudo_binary_path \| -cred cred_name \| -auth plugin_name [-arg1 name1:value1...]` | You must choose either `sudo` or `root` to access the remote nodes.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-targetnode node_name` | Optionally, you can specify the name of an `rhpclient`-less target. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-eval` | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |

### Usage Notes

- You can specify the `rhpclient`-less target for the operation using the working copy name or, if the destination cluster is an Oracle Fleet Patching and Provisioning Client, then using the client cluster name.

- You must provide either `root` credentials or a `sudo` user.
- A destination node is required if the destination cluster is an Oracle Clusterware 11*g* release 2 (11.2) or 12*c* release 1 (12.1) cluster and must be the node name of an existing cluster node.

## rhpctl move gihome

Moves the Oracle Grid Infrastructure software stack from one home to another.

### Syntax

```
rhpctl move gihome -destwc destination_workingcopy_name
   {-sourcewc source_workingcopy_name | -sourcehome oracle_home_path}
   [-targetnode target_node_name] [-ignorewcpatches] [-nonrolling] [-
keepplacement]
   [-auto -dbhomes mapping_of_Oracle_homes] [-dblist db_name_list
   | -excludedblist db_name_list] [-nodatapatch] [-skipdatapatchcheck] [-
disconnect]
   [-stopoption stop_option] [-drain_timeout timeout]
   [-dbsinparallel number_of_instances] [-raconetimeout timeout]]
   [-batches list_of_batches [-chainbatches] [-noparallel {YES | NO}] | -
smartmove [-saf availability] [-stopbetweenbatches]] [-eval]
   [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name] [-
pausebetweenbatches] [-tags tags_criteria]
   [[-tgip [-nodriverupdate]]] [-ignoremissingpatches
patch_name1[,patch_name2...]]
   | -continue | -revert |-abort | -forcecomplete} [-root | -cred cred_name |
-sudouser sudo_username -sudopath path_to_sudo_binary
   | -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]]
   [-cleanpids] [-useractiondata user_action_data] [ualoc
user_action_script_location] [-image image_name] [-smtpfrom "address"]
   [-smtpto "addresse1 addresse2 ..."] [-iso_repo iso_image] [-patchmgrloc
patch_mgr_loc]
   [-patchmgrargs patch_mgr_arguments] [-usepatchedhome] [-
ignoredbstarterror] [-excludedbs] [-ignorecvuprecheck | -skipcvuprecheck]
   [-ignorecvupostcheck | -skipcvupostcheck] [-obfuscate] [-checkdbpatching]
[-tgip -nodriverupdate]
```

### Parameters

**Table A-31 rhpctl move gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-destwc` *destination_workingcopy_name* | Specify the name of the destination working copy to which you want to move Oracle Grid Infrastructure. |
| `-sourcewc` *working_copy_name* | If you want to move Oracle Grid Infrastructure from a working copy, then specify the name of the source working copy from which you want to move the Grid home. |
| `-sourcehome` *oracle_home_path* | If you are moving Oracle Grid Infrastructure from an unmanaged (not provisioned by Fleet Patching and Provisioning) Oracle home, then specify the path to the Oracle home from which you want to move Oracle Grid Infrastructure. |

**Table A-31 (Cont.) rhpctl move gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-targetnode` *target_node_name* | Optionally, you can specify the name of an `rhpclient`-less target. |
| `-ignorewcpatches` | Use this parameter to ignore if the patched working copy is missing some patches which are present in the source path or working copy. |
| `-nonrolling` | Use this parameter to move the Oracle home in a non-rolling fashion. |
| `-keepplacement` | Specify this parameter to ensure that services of administrator-managed Oracle RAC or Oracle RAC One Node databases are running on the same instances before and after the move operation. |
| `-auto -dbhomes` *mapping_of_Oracle_homes* | Specify this parameter to automatically patch databases when you patch Oracle Grid Infrastructure. |
| `-dblist` *db_name_list* | Specify the unique names of the databases (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you want to move to the patched working copy.<br>**Note:** If you are moving a non-clustered (single-instance) database, then, for the value of the `-dbname` parameter, you must specify the *SID* of the database instead of the database name. |
| `-excludedblist` *db_name_list* | Alternative to using the `-dbname` parameter, you can use the `-excludedblist` parameter to patch all databases except specific databases. |
| `-nodatapatch` | Optionally, you can use this parameter to indicate not to run `datapatch` for databases being moved. |
| `-skipdatapatchchecks` | Use this parameter to skip executing datapatch sanity checks. |
| `-disconnect` | Optionally, you can use this parameter to disconnect all sessions before stopping or relocating services. |
| `-stopoption` *stop_option* | Optionally, you can choose one of the following stop options for the database: ABORT, IMMEDIATE, NORMAL, TRANSACTIONAL, or TRANSACTIONAL_LOCAL. |
| `-drain_timeout` *session_drain_time* | Optionally, you can use this parameter to specify a service drain timeout, in seconds. |
| `-dbsinparallel` *number_of_instances* | Specifies the number of database instances that can be started in parallel on a given node. |
| `-raconetimeout` *timeout* | Specify the Oracle RAC One Node database relocation timeout in minutes. |
| `-batches` *list_of_batches* | Optionally, you can specify a comma-delimited list of batches of nodes (where each batch is a comma-delimited list of node names enclosed in parentheses) enclosed in double quotation marks ("") in the format: `"(nA,nB,...),(...,nY,nZ)"`. |
| `-chainbatches` | Use this parameter to run the command on all batches without pausing after each batch. When this parameter is used, you do not have to use the `-continue` parameter after the operation is completed on each batch. |
| `-noparallel {YES \| NO}` | Process the nodes in the input batch serially and exit after all nodes in the batch are patched. |

**Table A-31 (Cont.) rhpctl move gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-smartmove [-saf availability] [-stopbetweenbatches]` | Alternatively, you can use the `-smartmove` parameter to auto-generate a list of batches of nodes and move databases by restarting instances after each batch.<br>Optionally, you can use the `-saf` parameter to specify the service availability factor, which is the minimum percentage of instances on which a service must remain running during the move.<br>Use the `-stopbetweenbatches` parameter to process batches separately. Smart move operation pauses for user intervention before continuing with the next batch. |
| `-eval` | Use this parameter to evaluate the `rhpctl move gihome` command and print automatically generated batches of nodes and the sequence of moves without actually running the command. |
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command.<br>**Note:**<br>If the `-schedule` parameter is used with the `-batches` parameter, then the command stops after the first batch and you have to use the `-continue` parameter to run the next batch. A new job ID is generated for every batch operation. |
| `-tags tags_criteria` | Optionally, specify tags criteria to create child jobs internally in the PAUSED state. Use the `rhpctl resume job` command to continue the operations of all the child jobs. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-pausebetweenbatches` | Use this parameter to pause between two batches, which you can rerun using the `rhpctl resume -job` command. If this parameter is used, the all the batches are run using the same job ID. |
| `-tgip [-nodriverupdate]` | Performs a transparent move of the Oracle Grid Infrastructure home.<br>The optional `-nodriverupdate` option skips the patching of the drivers if the patch contains a driver patch. |
| `-ignoremissingpatches patch_name1[,patch_name2...]` | Proceed with the move and/or upgrade although the specified patches, which are present in the source path or working copy, may be missing from the destination path or working copy. |
| `-continue` | Use this parameter to continue restarting the Oracle Clusterware stack on the next batch of nodes. |

**Table A-31 (Cont.) rhpctl move gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-revert` | Use this parameter to revert back to before the move operation. |
| `-abort` | Use this parameter to stop an ongoing move operation. |
| `-forcecomplete` | Use this parameter to mark the move operation as complete after completing it manually. |
| `-root \| -cred` *cred_name* `\| -sudouser` *sudo_user_name* `-sudopath` *sudo_binary_location* | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or you can use `-auth` to use an authentication plugin to access the remote node. |
| `-auth` *plugin-name* `[-arg1` *name1*`:`*value1* `[-arg2` *name2*`:`*value2* `...]]` | Use an authentication plugin to access the remote node.<br>Optionally provide a list of arguments to the plugin. |
| `-cleanpids` | When using a persistent home path for both the source and destination working copies, specify `-cleanpids` to ensure processes are stopped completely on the source home. |
| `-useractiondata` *user_action_data* | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-ualoc` *user_action_script_location* | Optionally, you can specify the location for the user action scripts. |
| `-image` *image_name* | Specifies the name of the image. For Oracle Exadata, this is the Exadata image name. |
| `-smtpfrom "`*address*`"` | Optionally, you can specify an email address enclosed in double quotation marks ("") from which Oracle Fleet Patching and Provisioning sends patch manager notifications. |
| `-smtpto "`*addresse1 addresse2 ...*`"` | Optionally, you can specify several email address enclosed in double quotation marks ("") to which Oracle Fleet Patching and Provisioning sends patch manager notifications. |
| `-iso_repo` *iso_image* | Specifies the image in the ISO repository. |
| `-patchmgrloc` *patch_mgr_loc* | Specifies the patch manager location. |
| `-patchmgrargs` | Specifies the patch manager arguments. |
| `-usepatchedhome` | Specify this parameter to use patched home to run Oracle Fleet Patching Provisioning Server and Client for Oracle Grid Infrastructure patching. |
| `-ignoredbstarterror` | Use this parameter to ignore the database startup errors during Oracle Grid Infrastructure patching. |
| `-excludedbs` *file_path* | Use this parameter to start all patched databases except for the database names specified in the input file. This parameter accepts path of a file that contains comma-separated list of database names, which is `DB_UNIQUE_NAME`. |

**Table A-31 (Cont.) rhpctl move gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-ignorecvuprecheck` | Ignore errors during CVU pre-requisites Oracle Grid Infrastructure upgrade check. |
| `-skipcvuprecheck` | Skip checks during CVU pre-requisites Oracle Grid Infrastructure upgrade check. |
| `-ignorecvupostcheck` | Ignore errors during post Oracle Grid Infrastructure upgrade CVU check. |
| `-skipcvupostcheck` | Skip checks during post Oracle Grid Infrastructure upgrade CVU check. |
| `-obfuscate` | Obfuscate patch storage contents. |
| `-checkdbpatching` | Ensure that there is no ongoing Oracle Database patching operation in the target cluster. |
| `-tgip` | Performs transparent move of the Oracle Grid Infrastructure home.<br><br>**Note:**<br>This parameter is available only for Oracle FPP Local Mode. |
| `-nodriverupdate` | Skips patching the drivers if the patch contains a driver patch.<br><br>**Note:**<br>This parameter is available only for Oracle FPP Local Mode. |

### Usage Notes

If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

### Example

Assume there is an `rhpclient`-less target running Oracle Grid Infrastructure 19c from a working copy named `grid23aiwcpy`, and one of the nodes in the cluster is named `bposvr141`. After provisioning the patched working copy, called `grid23aiPSU` (using the `-softwareonly` parameter with the `rhpctl add workingcopy` command), move the Grid home to the patched working copy, as follows:

```
$ rhpctl move gihome -sourcewc grid23aiwcpy -destwc grid23aiPSU -root -
targetnode bposvr141
```

## rhpctl upgrade gihome

Upgrades the Oracle Grid Infrastructure from a source working copy or source home path to a destination working copy.

### Syntax

```
rhpctl upgrade gihome {-sourcewc source_workingcopy_name |
     -sourcehome oracle_home_path -targetnode target_node_name}
   -destwc destination_workingcopy_name
  [-revert {-root | -sudouser sudo_user_name -sudopath sudo_binary_location]
      -cred cred_name |
      -auth plugin_name [-arg1 name1:value1...]  [-arg2 name2:value2 …]]}
  [-ignoreprereq] [-useractiondata user_action_data]
  [-eval] [-batches list_of_batches] [-abort | -continue]
  [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name]
  [-ignoremissingpatches patch_name1[,patch_name2...]]
  [-wallet wallet_file | -dbuser db_user]
  [-ignorecvucheck]
```

### Parameters

**Table A-32 rhpctl upgrade gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-sourcewc` *source_workingcopy_name* | Specify the name of the source working copy from which the Oracle Grid Infrastructure home needs to be upgraded. |
| `-sourcehome` *oracle_home_path* | Alternative to specifying the name of the source working copy, you can specify the path to the unmanaged Oracle Grid Infrastructure home. |
| `-targetnode` *target_node_name* | In addition to specifying the source Oracle Grid Infrastructure home, you must also specify an `rhpclient`-less target. |
| `-destwc` *destination_workingcopy_name* | Specify the name of the destination working copy to which the Oracle Grid Infrastructure home is to be upgraded. |
| `-revert` | Specify this option to downgrade after a failed upgrade of Oracle Grid Infrastructure. |
| `-root \| -sudouser` *sudo_username* `-sudopath` *sudo_binary_path* `\| -cred` *cred_name* | If you choose to use the `-targetnode` parameter, then you must choose either `sudo` or `root` to access the remote node.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-auth` *plugin-name* `[-arg1` *name1*`:`*value1* `[-arg2` *name2*`:`*value2* `...]]` | Use an authentication plugin to access the remote node.<br>Optionally provide a list of arguments to the plugin. |
| `-ignoreprereq` | Use this parameter to ignore the CVU prerequisite checks. |

**Table A-32 (Cont.) rhpctl upgrade gihome Command Parameters**

| Parameter | Description |
|---|---|
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br><br>`2018-07-25T19:13:17+05`<br><br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command.<br><br>**Note:**<br>If the `-schedule` parameter is used with the `-batches` parameter, then the command stops after the first batch and you have to use the `-continue` parameter to run the next batch. A new job ID is generated for every batch operation. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-useractiondata user_action_data` | Value to be passed to `useractiondata` parameter of the `useraction` script. |
| `-eval` | Evaluate without running the command. |
| `-batches list_of_batches` | List of batches of nodes in the format: "`(Ba),...,(Bz)`". |
| `-abort \| -continue` | Abort the ongoing move operation or continue the aborted move operation and continue restarting the CRS stack on the next batch of nodes. |
| `-ignoremissingpatches patch_name1[,patch_name2...]` | Proceed with the move and/or upgrade although the specified patches, which are present in the source path or working copy, may be missing from the destination path or working copy. |
| `-wallet wallet_file` | Specify the wallet file to get credentials for the DB user. If this option is not specified, then you are prompted for a password every time you connect to the external metadata repository. |
| `-dbuser db_user` | Specify a database user with DBA privileges on the external metadata repository. The default user is `SYS`. This parameter is not required if wallet is specified. |
| `-ignorecvucheck` | Ignore errors during post Oracle Grid Infrastructure upgrade CVU check . |

**Note:**

For self upgrade of FPPS to 23ai, you must use the `rhpctl.sh` script from the Oracle Grid Infrastructure 23ai home.

# home Commands

Use commands with the `home` keyword to deploy an Oracle Grid Infrastructure or Oracle Database home using Oracle FPP Local Mode.

- rhpctl deploy home
  Use the `rhpctl deploy home` command to deploy Oracle Grid Infrastructure and Oracle Database homes from gold images in Oracle FPP Local Mode.

## rhpctl deploy home

Use the `rhpctl deploy home` command to deploy Oracle Grid Infrastructure and Oracle Database homes from gold images in Oracle FPP Local Mode.

### Syntax

```
rhpctl deploy home -path target_home_path -zip gold_image_zip_path
[-sourcehome source_home_path] [-ignoreprereq]
```

### Parameters

**Table A-33 rhpctl deploy home Command Parameters**

| Parameter | Description |
|---|---|
| `-path target_home_path` | Specify the absolute target home path for provisioning the software home. |
| `-zip gold_image_zip_path` | Specify the absolute path of the compressed gold image. |
| `-sourcehome source_home_path` | Specify the path to the source Oracle home. |
| `-ignoreprereq` | Specify this parameter to ignore the CVU prerequisite checks. |

### Examples

To provision Oracle Grid Infrastructure 23ai using the specified gold image:

```
$ rhpctl deploy home -path /u01/app/23.0.0/grid -zip /tmp/images/grid_home.zip
```

# image Commands

Use commands with the `image` keyword to add, delete, import, and manage gold images.

- rhpctl add image
  Use the `rhpctl add image` command to create an image from an existing working copy and add it to the list of existing images on the Fleet Patching and Provisioning Server configuration.
- rhpctl allow image
- rhpctl delete image
  Deletes a specific image.
- rhpctl deploy image
  Deploys an image to a specific node in a client cluster.

- rhpctl disallow image
- rhpctl import image
  Creates an image on the Fleet Patching and Provisioning Server.
- rhpctl instantiate image
- rhpctl modify image
- rhpctl query image
  Displays the configuration of an existing image.
- rhpctl promote image
- rhpctl register image
  Registers an image metadata to the Oracle FPP repository.
- rhpctl uninstantiate image

## rhpctl add image

Use the `rhpctl add image` command to create an image from an existing working copy and add it to the list of existing images on the Fleet Patching and Provisioning Server configuration.

### Syntax

```
rhpctl add image -image image_name -workingcopy working_copy_name
   [-imagetype image_type] [-series series_name] [-state {TESTABLE |
RESTRICTED | PUBLISHED}]
```

### Parameters

**Table A-34 rhpctl add image Command Parameters**

| Command Option | Description |
|---|---|
| `-image image_name` | Specify the name of the image that you want to add. |
| `-workingcopy working_copy_name` | Specify the name of the working copy from which to create the image.<br><br>**Note:**<br>The working copy must be stored on ACFS storage for this command to work. Oracle FPP returns an error if the working copy is stored on a local storage device. |
| `-imagetype image_type` | Specify the software type. `ORACLEDBSOFTWARE` (default) for Oracle Database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, `LINUXOS` for Linux operating system ISO, or `SOFTWARE` for all other software. If you use custom image types, then specify the name of your image type. |
| `-series series_name` | If you want to add an image to an image series, then specify the name of an image series. |
| `-state {TESTABLE \| RESTRICTED \| PUBLISHED}` | Specify the state of the image. |

### Usage Notes

> **See Also:**
>
> Patching Oracle Database for details about how to use this command in the workflow for creating patched Oracle Database software homes

### Example

An example of this command is:

```
$ rhpctl add image -image DB12201_PATCH -workingcopy temp_wcpy_db12201_patch
```

## rhpctl allow image

Allows access to an image by a user or a role.

### Syntax

```
rhpctl allow image -image image_name {-user user_name [-client cluster_name]
    | -role role_name}
```

### Parameters

**Table A-35 rhpctl allow image Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of the image to which you want to allow access. |
| `-user user_name [-client cluster_name \| -role role_name` | Specify the either of the following:<br>• A user for which you want to allow access to the image and, optionally, the cluster name of the client cluster with the user.<br>• The role for which you want to allow access to the image. |

### Examples

To allow access to an image named PRODIMAGE:

```
$ rhpctl allow image -image PRODIMAGE -user mjk -client GHC1
```

## rhpctl delete image

Deletes a specific image.

### Syntax

```
rhpctl delete image -image image_name [-schedule {timer_value | NOW | PAUSE}]
[-tag tag_name]
    [-client cluster_name | -server server_cluster_name | -local] [-archive
archive_path]
```

### Usage Notes

- Specify the name of the image you want to delete.
- Optionally, you can use the `-schedule` parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:

  ```
  2018-07-25T19:13:17+05
  ```

  If you choose to use this parameter, then you must run this command on the Fleet Patching and Provisioning Server. If `NOW` is specified, then the job is scheduled immediately. If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid` *`job_id`* command.
- Optionally, you can associate a user-defined tag with the scheduled jobs.
- This command will fail if the image belongs to one or more series
- This command will fail if there are any provisioned working copies based on this image
- Optionaly, you can use the `-archive` parameter to archive the image at the specified location, instead of permanently deleting the image.

### Example

The following example deletes an image named PRODIMAGEV0:

```
$ rhpctl delete image -image PRODIMAGEV0 -archive /scratch/imagearchive
```

## rhpctl deploy image

Deploys an image to a specific node in a client cluster.

### Syntax

```
rhpctl deploy image -image image_name -path path_to_dir [-targetnode
node_name {-root
    | -cred cred_name | -sudouser sudo_username -sudopath path_to_sudo_binary
| -auth plugin_name
    [-arg1 name1:value1 [-arg2 name2:value2 ...]]}] [-client cluster_name]
```

### Parameters

**Table A-36 rhpctl deploy image Command Parameters**

| Parameter | Description |
|---|---|
| `-image` *`image_name`* | Specify the name of the image you want to deploy. |
| `-path` | Specify the absolute location where you want to deploy the image. |
| `-targetnode` *`node_name`* | Optionally, you can specify the name of a node to which you want to deploy the image. This parameter is required if the node hosting the home is not a Fleet Patching and Provisioning Client. |

**Table A-36 (Cont.) rhpctl deploy image Command Parameters**

| Parameter | Description |
|---|---|
| `-root \| -cred` *`cred_name`* `\| -sudouser` *`sudo_user_name`* `-sudopath` *`sudo_binary_location`* `\| -auth` *`plugin_name plugin_args`* | Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-client` | Optionally, you can specify the name of the client cluster. |

### Usage Notes

You can only run this command from a Fleet Patching and Provisioning Server.

### Example

The following example deploys an Oracle Database Appliance image to a node:

```
$ rhpctl deploy image -image ODA1 -path /u01/app/dbusr/product/21.0.0/db21c -
targetnode racgbox1 -root
```

## rhpctl disallow image

Disallows access to an image by a user or a role.

### Syntax

```
rhpctl disallow image -image image_name {-user user_name [-client client_name]
    | -role role_name}
```

### Parameters

**Table A-37 rhpctl disallow image Command Parameters**

| Parameter | Description |
|---|---|
| `-image` *`image_name`* | Specify the name of the image to which you want to disallow access. |
| `-user` *`user_name`* `[-client` *`client_name`* `\| -role` *`role_name`* | Specify either of the following:<br>• A user for which you want to disallow access to the image and, optionally, the cluster name of the client cluster with the user.<br>• The role for which you want to disallow access to the image. |

### Examples

To disallow access to an image:

```
$ rhpctl disallow image -image PRODIMAGE -user mjk -client GHC1
```

# rhpctl import image

Creates an image on the Fleet Patching and Provisioning Server.

Use the `rhpctl import image` command to create an image by copying the entire software contents from the specified path to the Oracle Fleet Patching and Provisioning Server.

### Syntax

```
rhpctl import image -image image_name {-path path | -zip zipped_home_path
   | -store_as_zip | -location zipped_home_path | -notify [-cc user_list]}
   [-imagetype image_type] [-pathowner user_name] [-version software_version]
   [-state {TESTABLE | RESTRICTED | PUBLISHED}] [-location zipped_home_path]
[-client cluster_name]
   [-targetnode node_name [-sudouser sudo_user_name -sudopath
sudo_binary_path |
   -root | cred cred_name | -auth plugin_name [-arg1 name1:value1 [-arg2
name2:value2 ... ]]]]
   [-useractiondata user_action_data] [] [-series series_name]
```

### Parameters

**Table A-38 rhpctl import image Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of the image that you want to add. |
| `-path path` | Specify the absolute path location of the software home that you want to import. For Oracle Database images, this is the `ORACLE_HOME`.<br><br>**Note:**<br>For Oracle Grid Infrastructure image import on IBM AIX servers, you must not import image from a linked Oracle Grid Infrastructure home. |
| `-zip zipped_home_path` | Specify the absolute path of the compressed software home to be imported (a ZIP or TAR file).<br><br>**Note:**<br>Do not use this option when importing an image from another platform. This option works only on the same platform, for example, if you are on a Linux platform, then you can use the `-zip` option to import an image only from another Linux system. |
| `-store_as_zip` | Specify this option to store the compressed Oracle home as a zip file. This option works only if you import the image as a zip file. |

**Table A-38 (Cont.) rhpctl import image Command Parameters**

| Parameter | Description |
|---|---|
| `-location` *zipped_home_path* | Specify the location of the compressed image file on the destination cluster. |
| `-notify [-cc` *users_list*`]` | Send an email notification.<br>Specify a list of users to whom email notifications is sent (in addition to the owner of working copy). |
| `-imagetype` *image_type* | Specify the software type. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, `ODAPATCHSOFTWARE` for engineered systems (Oracle Data Appliance), `EXAPATCHSOFTWARE` for Oracle Exadata software, or `SOFTWARE` for all other software. For a custom image type, use the image type name.<br>**Note:**<br>For Oracle Grid Infrastructure image import on IBM AIX servers, specify `SOFTWARE` image type. |
| `-version` *software_version* | Optionally, you can specify the version of the software you are importing. |
| `-pathowner` *user_name* | Specify the user with read access to the files and directories under the specified path.<br>**Note:** This parameter is applicable only for non-Oracle database software homes. |
| `-state {TESTABLE \| RESTRICTED \| PUBLISHED` | Specify whether the state of the image is testable, restricted, or published. |
| `-location` *zipped_home_path* | Location of the compressed image file on the destination cluster. |
| `-client` *cluster_name* | Specify the name of the client cluster. |
| `-targetnode` *node_name* | Specify the name of the node from which you want to import the image. This parameter is required if the node hosting the home is an `rhpclient`-less target |
| `-sudouser` *sudo_user_name* `-sudopath` *sudo_binary_path* `\| -root]` | If you use the `-targetnode` parameter, then you must specify either sudo or `root` to perform super user operations. |
| `-auth` *plugin-name* `[-arg1` *name1*`:`*value1* `[-arg2` *name2*`:`*value2* `...]]` | Use an authentication plugin to access the remote node.<br>Optionally provide a list of arguments to the plugin. |
| `-useractiondata` *user_action_data* | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-series` *series_name* | The name of the series. |

### Usage Notes

- You can only run this command on a Fleet Patching and Provisioning Server.
- When you import an Oracle Database or Oracle Grid Infrastructure software home, the version of the home must be one of the versions that Fleet Patching and Provisioning supports for provisioning and patching.

### Examples

The following example imports an image:

```
$ rhpctl import image -image PRODIMAGEV1 -path /u01/app/product/12.1.0/dbhome
-pathowner orcl
```

The following example imports an engineered system image:

```
$ rhpctl import image -image ODA1 -imagetype ODAPATCHSOFTWARE -path /tmp/
ODAPatchBundle -version 12.1.2.8.0
```

## rhpctl instantiate image

Requests copies of gold images from a peer Fleet Patching and Provisioning Server.

### Syntax

```
rhpctl instantiate image -server server_cluster_name {-image image_name
  | -series series_name | -imagetype image_type | -all}
```

### Parameters

**Table A-39 rhpctl instantiate image Command Parameters**

| Parameter | Description |
|---|---|
| `-server` *`server_cluster_name`* | Specify a Fleet Patching and Provisioning Server cluster from which you want to request images. |
| `-image` *`image_name`* `\|` `-series` *`series_name`* `\|` `-imagetype` *`image_type`* `\|` `-all` | You can request copies of gold images from a peer Fleet Patching and Provisioning Server, specifically, by image name, series name, or image type. Alternatively, you can use the `-all` parameter to request copies of all gold images from the peer Fleet Patching and Provisioning server.<br>If you choose to request images by image type, then specify `ORACLEDBSOFTWARE` (default) for Oracle Database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, and `SOFTWARE` for all other software. For a custom image type, use the image type name. |

### Usage Notes

- User actions associated with an image being copied are not themselves copied.
- Groups configuration of a gold image is replicated in copies sent to peers.
- Copies of gold images are in the PUBLISHED state.

# rhpctl modify image

Modifies the configuration details of an image.

### Syntax

```
rhpctl modify image -image image_name -imagetype image_type
```

### Parameters

**Table A-40 rhpctl modify image Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of the image that you want to modify. |
| `-imagetype image_type` | You can modify the software type. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, or `SOFTWARE` for all other software. For a custom image type, use the image type name. |

# rhpctl query image

Displays the configuration of an existing image.

### Syntax

```
rhpctl query image {[[-image image_name [-dbtemplate]] | [[-imagetype
image_type]
  [-version version] [-platform platform] [-complete]]]
  [-server server_cluster_name | -client client_name | -local] | -drift}
  [-rhpserver rhps_regex]
```

### Parameters

**Table A-41 rhpctl query image Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name [-dbtemplate]` | Specify the name of the image you want to query.<br>Optionally, you can use the `-dbtemplate` parameter to display template file names in the default template directory. |
| `-imagetype image_type` | Specify the software type. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software,or `SOFTWARE` for all other software. For a custom image type, use the image type name. |
| `-version version` | Specify the version of the image software you are querying. |
| `-platform platform` | Specify the operating system platform to which the image corresponds. |
| `-complete` | Use this parameter to list all the images for which image completion is set to `TRUE`. |
| `-server server_cluster_name` | Specify the name of the server cluster to which the image corresponds. |

**Table A-41 (Cont.) rhpctl query image Command Parameters**

| Parameter | Description |
|---|---|
| `-client client_name` | If this paramenter is used with the `-image` option, the command checks if the specified client has the given image instantiated on it. Otherwise, the use of the parameter lists the images instantiated on the given client. |
| `-local` | List the images stored on the current client. |
| `-drift` | List the the bug fixes not included in the golden image. |
| `-rhpserver rhps_regex` | Specify a regular expression to match the cluster name of the servers where the operation must be performed. |

### Usage Notes

If you use the `-version` parameter, then the version must have five fields, such as 12.1.0.2.4.

If you use the `-platform` parameter, then you can use `Linux_AMD64`, `Linux_S390`, `Linux_PPC`, `IBM_AIX_PPC64`, `HP_IA64`, `Linux_Itanium`, `Solaris_SPARC64`, `Linux_LOP`, and `Intel_Solaris_AMD64`

The `-rhpserver` parameter indicates where the operation should be performed and the `-server` parameter indicates what the operation should be about. When `-rhpserver` is used, the command is run on the servers whose name matches the regular expression provided. When `-server` is used, the command is about the server provided. For example:

- The following command runs locally and lists the images that have been replicated to `peerA`

  ```
  rhpctl query image -server peerA
  ```

- The following command runs on `peerA` and lists the images have been configured on `peerA`

  ```
  rhpctl query image -rhpserver peerA
  ```

- The following command runs on all servers and lists the images that each server has replicated on `peerA`

  ```
  rhpctl query image -server peerA -rhpserver .+'
  ```

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles
role=user_name[+user_name...][,role=user_name[+user_name...]...]
```

For information about granting roles with RHPCTL, refer to rhpctl grant role

## rhpctl promote image

Promotes an image.

### Syntax

```
rhpctl promote image -image image_name -state {TESTABLE | RESTRICTED |
PUBLISHED}
```

### Parameters

**Table A-42 rhpctl promote image Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of the image that you want to promote. |
| `-state {TESTABLE \| RESTRICTED \| PUBLISHED}` | Specify one of the following as the name of the state of the image:<br>TESTABLE:<br>RESTRICTED:<br>PUBLISHED: |

### Example

To promote an image named PRODIMAGE:

```
$ rhpctl promote image -image PRODIMAGE -state RESTRICTED
```

## rhpctl register image

Registers an image metadata to the Oracle FPP repository.

### Syntax

```
rhpctl register image -image image_name
   {-path home_path | -zip zipped_home_path} [-imagetype image_type] [-
pathowner username]
   [-state {TESTABLE| RESTRICTED|PUBLISHED}] [-client cluster_name] [-
targetnode target_node_name [-sudouser sudo_username
   -sudopath path_to_sudo_binary | -root | -cred cred_name]] [-useractiondata
user_action_data]
```

### Parameters

**Table A-43 rhpctl register image Command Parameters**

| Parameter | Description |
|---|---|
| `-image image_name` | Specify the name of a configured image from which to register a working copy or the name of an image series from which RHPCTL takes the latest image when adding a working copy. |

**Table A-43 (Cont.) rhpctl register image Command Parameters**

| Parameter | Description |
|---|---|
| `-path home_path` | Specify the absolute path for provisioning the software home. For Oracle Database images, this becomes the `ORACLE_HOME`. |
| `-zip zipped_home_path` | Specify the absolute path of the compressed software home to be imported (a ZIP or TAR file). |
| `-imagetype image_type` | Specify the software type. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, `ODAPATCHSOFTWARE` for engineered systems (Oracle Data Appliance), `EXAPATCHSOFTWARE` for Oracle Exadata software, or `SOFTWARE` for all other software. For a custom image type, use the image type name. |
| `-pathowner user_name` | Specify the user with read access to the files and directories under the specified path.<br>**Note:** This parameter is applicable only for non-Oracle database software homes. |
| `-state {TESTABLE \| RESTRICTED \| PUBLISHED}` | Specify whether the state of the image is testable, restricted, or published. |
| `-client cluster_name` | Specify the name of the client cluster. |
| `-targetnode target_node_name` | Specify the name of an `rhpclient`-less target. |
| `-sudouser sudo_user_name -sudopath sudo_binary_path \| -root \| -cred cred_name` | If you use the `-targetnode` parameter, then you must specify either sudo or `root` to perform super user operations. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |

### Examples

- To register an Oracle Database image:

```
$ rhpctl register image -image PRODIMAGEV1 -path /u01/app/product/23.5.0/
dbhome_1 -pathowner orcl
```

## rhpctl uninstantiate image

Stops updates for previously requested images from a peer Fleet Patching and Provisioning Server.

### Syntax

```
rhpctl uninstantiate image -server server_cluster_name {-image image_name
  | -series series_name | -imagetype image_type | -all}
```

### Parameters

**Table A-44 rhpctl uninstantiate image Command Parameters**

| Parameter | Description |
|---|---|
| `-server` *server_cluster_name* | Specify a Fleet Patching and Provisioning Server cluster from which you want to stop updates. |
| `-image` *image_name* \| `-series` *series_name* \| `-imagetype` *image_type* \| `-all` | You can get updates from a peer Fleet Patching and Provisioning Server, specifically, by image name, series name, or image type. Alternatively, you can use the `-all` parameter to stop updates from the peer Fleet Patching and Provisioning server.<br>If you choose to stop updates by image type, then specify `ORACLEDBSOFTWARE` (default) for Oracle Database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, and `SOFTWARE` for all other software. For a custom image type, use the image type name. |

# imagetype Commands

Use commands with the `imagetype` keyword to add, delete, modify, and manage an image type.

- rhpctl add imagetype
  Configures a new image type and its associated user actions.
- rhpctl allow imagetype
  Grants access to an image type to a user or a role.
- rhpctl delete imagetype
- rhpctl disallow imagetype
- rhpctl modify imagetype
- rhpctl query imagetype

## rhpctl add imagetype

Configures a new image type and its associated user actions.

### Syntax

```
rhpctl add imagetype -imagetype image_type -basetype {SOFTWARE |
  ORACLEGISOFTWARE | ORACLEDBSOFTWARE | ORACLEGGSOFTWARE}
  [-useractions user_action_list]
```

### Parameters

**Table A-45 rhpctl add imagetype Command Parameters**

| Parameter | Description |
|---|---|
| `-imagetype` *image_type* | Specify the name of the image type you are creating. |

**Table A-45 (Cont.) rhpctl add imagetype Command Parameters**

| Parameter | Description |
|---|---|
| `-basetype {SOFTWARE \| ORACLEGISOFTWARE \| ORACLEDBSOFTWARE \| ORACLEGGSOFTWARE}` | Specify a base image type on which the image type you are creating is based. Use `ORACLEDBSOFTWARE` (default) for Oracle Database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, and `SOFTWARE` for all other software. |
| `-useractions` *`user_action_list`* | Specify a comma-delimited list of names of user actions |

### Example

To add a new image type:

```
rhpctl add imagetype -imagetype DB122_PATCH_TYPE -basetype ORACLEDBSOFTWARE
```

## rhpctl allow imagetype

Grants access to an image type to a user or a role.

### Syntax

```
rhpctl allow imagetype -imagetype image_type {-user user_name [-client
cluster_name] | -role role_name}
```

### Parameters

**Table A-46 rhpctl allow imagetype Command Parameters**

| Parameter | Description |
|---|---|
| `-imagetype` *`image_type`* | Specify the software type. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, `ODAPATCHSOFTWARE` for engineered systems (Oracle Data Appliance), `EXAPATCHSOFTWARE` for Oracle Exadata software, or `SOFTWARE` for all other software. For a custom image type, use the image type name. |
| `-user` *`user_name`* | Specify an operating system user to whom you are granting access to the image type. Either this parameter or the `-role` parameter is required. |
| `-client` *`cluster_name`* | Optionally, you can specify the name of the client cluster to which the operating system user belongs, if you choose to use the `-user` parameter. |
| `-role` *`role_name`* | Alternative to the `-user` parameter, you can specify a particular role to which to grant access to the image. |

## rhpctl delete imagetype

Deletes an existing image type.

### Syntax

```
rhpctl delete imagetype -imagetype image_type
```

### Usage Notes

Specify an image type to delete. You cannot delete any of the built-in image types.

## rhpctl disallow imagetype

Revokes access to an image type from a user or a role.

### Syntax

```
rhpctl disallow imagetype -imagetype image_type {-user user_name [-client
cluster_name] | -role role_name}
```

### Parameters

**Table A-47 rhpctl disallow imagetype Command Parameters**

| Parameter | Description |
|---|---|
| `-imagetype image_type` | Specify the software type. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, `ODAPATCHSOFTWARE` for engineered systems (Oracle Data Appliance), `EXAPATCHSOFTWARE` for Oracle Exadata software, or `SOFTWARE` for all other software. For a custom image type, use the image type name. |
| `-user user_name` | Specify an operating system user from whom you are revoking access to the image type. Either this parameter or the `-role` parameter is required. |
| `-client cluster_name` | Optionally, you can specify the name of the client cluster to which the operating system user belongs, if you choose to use the `-user` parameter. |
| `-role role_name` | Alternative to the `-user` parameter, you can specify a particular role from which to revoke access to the image. |

## rhpctl modify imagetype

Modifies the configuration of an image type.

### Syntax

```
rhpctl modify imagetype -imagetype image_type -useractions user_action_list
```

### Parameters

**Table A-48 rhpctl modify imagetype Command Parameters**

| Parameter | Description |
|---|---|
| `-imagetype image_type` | Specify the name of the image type you want to modify. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, or `SOFTWARE` for all other software. For a custom image type, use the image type name. |
| `-useractions user_action_list` | Specify a comma-delimited list of names of user actions |

## rhpctl query imagetype

Displays the configuration of an image type.

### Syntax

```
rhpctl query imagetype -imagetype image_type [-rhpserver rhps_regex]
```

### Parameters

**Table A-49 rhpctl query imagetype Command Parameters**

| Parameter | Description |
|---|---|
| `-imagetype image_type` | Specify the name of the image type you want to query. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, or `SOFTWARE` for all other software. For a custom image type, use the image type name. |
| `-rhpserver rhps_regex` | Specify a regular expression to match the cluster name of the servers where the operation must be performed. |

### Usage Notes

Specify the name of the image type you want to query. Use ORACLEDBSOFTWARE (default) for Oracle database software, ORACLEGISOFTWARE for Oracle Grid Infrastructure software, or SOFTWARE for all other software. For a custom image type, use the image type name.

The `-rhpserver` parameter indicates where the operation should be performed and the `-server` parameter indicates what the operation should be about. When `-rhpserver` is used, the command is run on the servers whose name matches the regular expression provided. When `-server` is used, the command is about the server provided. For example:

- The following command runs locally and lists the image types that have been replicated to `peerA`

  ```
  rhpctl query imagetype -server peerA
  ```

- The following command runs on `peerA` and lists the image types have been configured on `peerA`

  ```
  rhpctl query imagetype -rhpserver peerA
  ```

- The following command runs on all servers and lists the image types that each server has replicated on `peerA`

  ```
  rhpctl query imagetype -server peerA -rhpserver .+'
  ```

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles
role=user_name[+user_name...][,role=user_name[+user_name...]...]
```

For information about granting roles with RHPCTL, refer to rhpctl grant role

## osconfig Commands

Use commands with the `osconfig` keyword to backup, compare, and manage operating system configuration information.

- rhpctl collect osconfig
- rhpctl compare osconfig
- rhpctl disable osconfig
- rhpctl enable osconfig
- rhpctl query osconfig

## rhpctl collect osconfig

Collects a backup of the operating system configuration for a cluster.

**Syntax**

```
rhpctl collect osconfig -client cluster_name [-targetnode node_name
  {-sudouser sudo_user_name -sudopath sudo_binary_location | -root}]
```

### Parameters

**Table A-50 rhpctl collect osconfig Command Parameters**

| Parameter | Description |
|---|---|
| `-client cluster_name` | Specify the name of the client cluster. |
| `-targetnode node_name` | Optionally, you can specify the name of an `rhpclient`-less target. |
| `-sudouser sudo_user_name -sudopath sudo_binary_path \| -root]` | If you use the `-targetnode` parameter, then you must specify either sudo or `root` to perform super user operations. |

## rhpctl compare osconfig

Compares operating system configurations for a specific cluster.

### Syntax

```
rhpctl compare osconfig -client cluster_name -node node_name -id1 identifier -
id2 identifier
```

### Parameters

**Table A-51 rhpctl compare osconfig Command Parameters**

| Parameter | Description |
|---|---|
| `-client cluster_name` | Specify the name of the client cluster in which you want to compare operating system configurations. |
| `-node node_name` | Specify the name of a node in a remote cluster. |
| `-id1 identifier` | Specify an identifier of an operating system configuration to be considered as a reference. |
| `-id2 identifier` | Specify an identifier of an operating system configuration to be compared. |

## rhpctl disable osconfig

Disables a scheduled backup of the operating system configuration and gives the option to delete all collected configuration backups.

### Syntax

```
rhpctl disable osconfig [-client cluster_name] [-clean]
```

### Usage Notes

- Optionally, you can specify a client cluster name on which you want to disable collection of operating system configuration information.
- Optionally, you can use the `-clean` parameter to delete all operating system configuration backups.

# rhpctl enable osconfig

Enable operating system configuration information collection for the client cluster.

### Syntax

```
rhpctl enable osconfig -client cluster_name [-retaincopies count]
  [-start timer_value] [-frequency collect_frequency] [-collectnow
   [-targetnode node_name {-sudouser sudo_user_name -sudopath
sudo_binary_location
    | -root}]] [-force]
```

### Parameters

**Table A-52 rhpctl enable osconfig Command Parameters**

| Parameter | Description |
|---|---|
| `-client cluster_name` | Specify the name of the client cluster. |
| `-retaincopies count` | Optionally, you can specify the number of scheduled backups you want to be maintained. The default value is `37`. |
| `-start timer_value` | Optionally, you can specify a start date and time to run configuration collection according to the following example: 2018-07-23T00:00:00-07 |
| `-frequency collect_frequency` | Optionally, you can specify the configuration collection interval in number of days. |
| `-collectnow` | Optionally, you can use this parameter to collect configuration information, immediately. |
| `-targetnode node_name` | Optionally, you can specify the name of an `rhpclient`-less target. |
| `-sudouser sudo_user_name -sudopath sudo_binary_path \| -root]` | If you use the `-targetnode` parameter, then you must specify either sudo or `root` to perform super user operations. |
| `-force` | Optionally, you can use this parameter to forcibly modify the count for the `-retaincopies` parameter previously set. |

# rhpctl query osconfig

Provides historic operating system configuration collection information, such as the collection schedule, retention count, scheduled job for periodic collection, and collection data.

### Syntax

```
rhpctl query osconfig -client client_name
```

### Usage Notes

Provide the name of the client cluster that you want to query operating system configuration collection information.

### Example

This command returns output similar to the following:

```
$ rhpctl query osconfig -client rhpdemocluster

OSConfig Enabled: true
Collection start time: "00:00:00"
Collection frequency: "1"
retaincopies count: "35"
OSConfig periodic Job ID: "38"
Collection storage path: "/scratch/rhp_storage/chkbase/osconfig/
rhpdemocluster"
Latest list of nodes for collections : "mjk00fwc"
OSConfig ID: "22" Collected on: "Jul 27, 2018 22:00:58 PM"
OSConfig ID: "21" Collected on: "Jul 26, 2018 22:00:47 PM"
OSConfig ID: "20" Collected on: "Jul 25, 2018 22:00:29 PM"
```

# patch Commands

Use commands with the `patch` keyword to apply or rollback one-off patches to the specified working copy using the in-place patching method.

- rhpctl apply patch
  Applies one-off patches to the specified working copy using in-place patching method.
- rhpctl rollback patch
  Rolls back the specified patch in the specified working copy.

## rhpctl apply patch

Applies one-off patches to the specified working copy using in-place patching method.

### Syntax

```
rhpctl apply patch -workingcopy workingcopy_name
  [-image image_name | -patchloc patch_location_on_target] [-online [-dbname
unique_db_name]]
  [-eval] [-targetnode target_node_name {-root | -cred cred_name | -sudouser
sudo_username
  -sudopath path_to_sudo_binary | -auth plugin_name [-arg1 name1:value1
  [-arg2 name2:value2 ...]]}] [-schedule timer_value] [-jobtag tag_name]
```

### Parameters

**Table A-53 rhpctl apply patch Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify a name for the working copy that you want to patch. |
| `-image image_name` | Specify the name of a configured image that you want to patch. |

**Table A-53 (Cont.) rhpctl apply patch Command Parameters**

| Parameter | Description |
|---|---|
| `-patchloc patch_location_on_target` | Specify the absolute path for the unzipped patch directory on the target. |
| `-online` | Specify this parameter to indicate that the patch type is online. |
| `-dbname unique_db_name` | Specify the unique name of the database (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you are patching. |
| `-eval` | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |
| `-targetnode target_node_name` | Specify the name of an `rhpclient`-less target. |
| `-root \| -cred cred_name \| -sudouser sudo_user_name -sudopath sudo_binary_location \| -auth plugin_name plugin_args` | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-schedule timer_value` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2024-04-15T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |

**Examples**

To patch a working copy on a client cluster:

```
rhpctl apply patch -workingcopy workingcopy_name -image image_name -online
```

## rhpctl rollback patch

Rolls back the specified patch in the specified working copy.

**Syntax**

```
rhpctl rollback patch -patchid patch_ids -workingcopy workingcopy_name
  [-online [-dbname unique_db_name]] [-targetnode target_node_name {-root | -
cred cred_name
  | -sudouser sudo_username -sudopath path_to_sudo_binary | -auth plugin_name
  [-arg1 name1:value1 [-arg2 name2:value2 ...]]}]
  [-schedule timer_value] [-jobtag tag_name]
```

### Parameters

**Table A-54 rhpctl rollback patch Command Parameters**

| Parameter | Description |
|---|---|
| `-patchid patch_id` | Specify a comma-separated list of patch IDs that you want to roll back. |
| `-workingcopy workingcopy_name` | Specify a name for the working copy from which you want to rollback the patch. |
| `-online` | Specify this parameter to indicate that the patch type is online. |
| `-dbname unique_db_name` | Specify the unique name of the database (`DB_UNIQUE_NAME` without `DB_DOMAIN`) from which you want to rollback the patch. |
| `-targetnode target_node_name` | Specify the name of an `rhpclient`-less target. |
| `-root \| -cred cred_name \| -sudouser sudo_user_name -sudopath sudo_binary_location \| -auth plugin_name plugin_args` | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-schedule timer_value` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2024-04-15T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |

### Examples

To rollback a patch from a working copy on a client cluster:

```
rhpctl rollback patch -patchid patch_id -workingcopy workingcopy_name -online
```

# peerserver Commands

Use commands with the `peerserver` keyword to display information for a peer server.

- rhpctl query peerserver

## rhpctl query peerserver

Displays information for a registered peer Fleet Patching and Provisioning Server.

### Syntax

```
rhpctl query peerserver [-server server_cluster_name [-serverPolicy]]
```

### Parameters

**Table A-55 rhpctl query peerserver Command Parameters**

| Parameter | Description |
|---|---|
| `-server` *server_cluster_name* | Optionally, you can specify the name of the Fleet Patching and Provisioning Server cluster for which you want to view the information. |
| `-serverPolicy` | Optionally, you can specify the image policy for the peer Fleet Patching and Provisioning Server for which you want to view the information. |

# role Commands

Use commands with the `role` keyword to add, delete, and manage roles.

- rhpctl add role
- rhpctl delete role
  Deletes a role from the list of existing roles on the Fleet Patching and Provisioning Server configuration.
- rhpctl grant role
- rhpctl query role
- rhpctl revoke role

## rhpctl add role

Creates roles and adds them to the list of existing roles on the Fleet Patching and Provisioning Server configuration.

> **See Also:**
>
> Fleet Patching and Provisioning Roles

### Syntax

```
rhpctl add role -role role_name -hasRoles roles
```

### Parameters

**Table A-56 rhpctl add role Command Parameters**

| Parameter | Description |
|---|---|
| `-role` *role_name* | Specify a name for the role that you want to create. |

**Table A-56 (Cont.) rhpctl add role Command Parameters**

| Parameter | Description |
|---|---|
| `-hasRoles` *roles* | Specify a comma-delimited list of roles to include with the new role.<br>GH_ROLE_ADMIN<br>GH_AUDIT_ADMIN<br>GH_USER_ADMIN<br>GH_SITE_ADMIN<br>GH_WC_ADMIN<br>GH_WC_OPER<br>GH_WC_USER<br>GH_IMG_ADMIN<br>GH_IMG_USER<br>GH_SUBSCRIBE_USER<br>GH_SUBSCRIBE_ADMIN<br>GH_IMGTYPE_ADMIN<br>GH_IMGTYPE_ALLOW<br>GH_IMGTYPE_OPER<br>GH_SERIES_ADMIN<br>GH_SERIES_CONTRIB<br>GH_IMG_TESTABLE<br>GH_IMG_RESTRICT<br>GH_IMG_PUBLISH<br>GH_IMG_VISIBILITY<br>GH_JOB_USER<br>GH_JOB_ADMIN<br>GH_AUTHENTICATED_USER<br>GH_CLIENT_ACCESS<br>GH_ROOT_UA_CREATE<br>GH_ROOT_UA_ASSOCIATE<br>GH_ROOT_UA_USE<br>GH_OPER<br>GH_CA<br>GH_SA<br>OTHER |

### Usage Notes

- You can only run this command on the Fleet Patching and Provisioning Server.
- You must be assigned the GH_ROLE_ADMIN role to run this command.

### Example

To add a role on the Fleet Patching and Provisioning Server:

```
$ rhpctl add role -role hr_admin -hasRoles GH_WC_USER,GH_IMG_USER
```

## rhpctl delete role

Deletes a role from the list of existing roles on the Fleet Patching and Provisioning Server configuration.

### Syntax

```
rhpctl delete role -role role_name
```

### Usage Notes

- Specify the name of the role that you want to delete
- You *cannot* delete any built-in roles
- You can only run this command on the Fleet Patching and Provisioning Server

### Example

To delete a role from the Fleet Patching and Provisioning Server:

```
$ rhpctl delete role -role hr_admin
```

## rhpctl grant role

Grants a role to a client cluster.

### Syntax

```
rhpctl grant role {-role role_name {-user user_name [-client cluster_name]
  | -grantee role_name}} | {[-client cluster_name]
  [-maproles role=user_name[+user_name...][,role=user_name[+user_name...]
[,...]}
```

> **Note:**
>
> The `-client` option works only on the Oracle Fleet Patching and Provisioning Server.

### Parameters

**Table A-57 rhpctl grant role Command Parameters**

| Parameter | Description |
|---|---|
| `-role role_name` | Specify the name of the role that you want to grant clients or users. |

**Table A-57 (Cont.) rhpctl grant role Command Parameters**

| Parameter | Description |
|---|---|
| `-user user_name [-client cluster_name]` | Specify the name of a user. The user name that you specify must be in the form of `user@rhpclient`, where `rhpclient` is the name of the Fleet Patching and Provisioning Client.<br>Optionally, you can specify the name of the client cluster to which the user belongs.<br>**Note:**<br>The `-client` option works only on the Oracle Fleet Patching and Provisioning Server. |
| `-grantee role_name` | Use this parameter to specify a role to which you want to grant another role. |
| `[-client cluster_name] -maproles role=user_name[+user_name...] [,role=user_name[+user_name...][,...]` | You can map either built-in roles or roles that you have defined to either users on a specific client cluster or to specific users.<br>When you use the `-maproles` parameter, use a plus sign (+) to map more than one user to a specific role. Separate additional role/user pairs with commas.<br>**Note:**<br>Starting with Oracle Grid Infrastructure 21c, the `-maproles` parameter is deprecated. This parameter can be desupported in a future release. |

### Example

The following example grants a role, ABC, to four specific users.

```
$ rhpctl grant role -role ABC -maproles
ABC=mjk@rhpc1+dc@rhpc1+aj@rhpc1+jc@rhpc1
```

## rhpctl query role

Displays the configuration information of a specific role.

### Syntax

```
rhpctl query role [-role role_name]
```

### Usage Notes

- Specify the name of the role for which you want to display the configuration information

- You can only run this command on the Fleet Patching and Provisioning Server

**Example**

This command returns output similar to the following:

```
$ rhpctl query role -role GH_CA

Role name: GH_CA
Associated roles: GH_IMGTYPE_ADMIN, GH_IMGTYPE_ALLOW, GH_IMGTYPE_OPER,
GH_IMG_ADMIN,
GH_IMG_PUBLISH, GH_IMG_RESTRICT, GH_IMG_TESTABLE, GH_IMG_VISIBILITY,
GH_SERIES_ADMIN,
GH_SERIES_CONTRIB, GH_SUBSCRIBE_ADMIN, GH_WC_ADMIN
Users with this role: rhpusr@rwsdcVM13
```

## rhpctl revoke role

Revokes a role from a client user.

**Syntax**

```
rhpctl revoke role {-role role_name {-user user_name
  [-client cluster_name] | -grantee role_name}}
  | {[-client cluster_name] -maproles role=user_name[+user_name...]
  [,role=user_name[+user_name...]...]}
```

**Parameters**

**Table A-58 rhpctl revoke role Command Parameters**

| Parameter | Description |
|---|---|
| `-role role_name` | Specify the name of the role from which you want to revoke clients or users. |
| `-user user_name [-client cluster_name]` | Specify the name of a user and, optionally, a client cluster from which you want to revoke a role. The user name that you specify must be in the form of `user@rhpclient`, where `rhpclient` is the name of the Fleet Patching and Provisioning Client. |
| `-grantee role_name` | Specify the grantee role name. |
| `[-client client_name] -maproles role=user_name[+user_name...]` | You can map either built-in roles or roles that you have defined to specific users. Use a plus sign (+) to map more than one user to a specific role. Separate additional role/user pairs with commas. Optionally, you can also specify a client cluster. |

> **Note:**
>
> Starting with Oracle Grid Infrastructure 21c, the `-maproles` parameter is deprecated. This parameter can be desupported in a future release.

# series Commands

Use commands with the `series` keyword to add, delete, subscribe, and manage a series.

- rhpctl add series
- rhpctl delete series
  Deletes a series from the Fleet Patching and Provisioning Server configuration.
- rhpctl deleteimage series
- rhpctl insertimage series
  Inserts an existing image into a series.
- rhpctl query series
- rhpctl subscribe series
- rhpctl unsubscribe series

## rhpctl add series

Adds a series to the Fleet Patching and Provisioning Server configuration.

### Syntax

```
rhpctl add series -series series_name [-image image_name]
```

### Parameters

**Table A-59 rhpctl add series Command Parameters**

| Parameter | Description |
|---|---|
| `-series series_name` | Specify a name for the series that you want to add. |
| `-image image_name` | Optionally, you can specify the name of a configured image. This image becomes the first in the series. |

### Example

To add a series:

```
$ rhpctl add series -series DB12_series
```

## rhpctl delete series

Deletes a series from the Fleet Patching and Provisioning Server configuration.

### Syntax

```
rhpctl delete series -series series_name [-force]
```

### Usage Notes

- Specify the name of the series that you want to delete.

- Use `-force` to delete an image series even if the series includes images.
- Before deleting an image series, you must first remove all images from the series by using the rhpctl deleteimage series command.
- This command *does not* delete images, only series.

**Example**

The following example deletes a series called PRODDBSERIES:

```
$ rhpctl delete series -series PRODDBSERIES
```

## rhpctl deleteimage series

Deletes an image from a series.

**Syntax**

```
rhpctl deleteimage series -series series_name -image image_name
```

**Parameters**

**Table A-60 rhpctl deleteimage series Command Parameters**

| Parameter | Description |
|---|---|
| `-series series_name` | Specify the name of the series from which you want to delete an image. |
| `-image image_name` | Specify the name of the image that you want to delete from a series. |

**Example**

The following command deletes an image called PRODIMAGEV0 from a series called PRODDBSERIES:

```
$ rhpctl deleteimage series -series PRODDBSERIES -image PRODIMAGEV0
```

## rhpctl insertimage series

Inserts an existing image into a series.

> **Note:**
>
> A single image can belong to one or more series.

**Syntax**

```
rhpctl insertimage series -series series_name -image image_name
   [-before image_name]
```

### Parameters

**Table A-61 rhpctl insertimage series Command Parameters**

| Parameter | Description |
|---|---|
| `-series series_name` | Specify the name of the series into which you want to insert an image. |
| `-image image_name` | Specify the name of the image that you want to insert into a series. |
| `-before image_name` | Optionally, you can specify the name of an image before which you want to insert the new image. |

### Example

To insert an image into a series:

```
rhpctl insertimage series -series DB12_series -image DB12102_PSU
```

## rhpctl query series

Displays the configuration of a series.

### Syntax

```
rhpctl query series [-series series_name | -image image_name]
  [-server server_cluster_name] [-rhpserver rhps_regex]
```

### Parameters

**Table A-62 rhpctl query series Command Parameters**

| Parameter | Description |
|---|---|
| `-series series_name` | Specify the name of the series for which you want to display the configuration. |
| `-image image_name` | Alternatively, you can specify the name of a configured image. |
| `-server server_cluster_name` | Specify the name of the server cluster to which the image corresponds. |
| `-rhpserver rhps_regex` | Specify a regular expression to match the cluster name of the servers where the operation must be performed. |

### Usage Notes

If you do not specify a series or an image by name, then CRSCTL returns information for all series.

The `-rhpserver` parameter indicates where the operation should be performed and `-server` indicates who the operation should be about. When `-rhpserver` is used, the command is run

on the servers whose name matches the regular expression provided. When `-server` is used, the command will be about the server provided. For example:

- The following command runs locally and lists the series that have been replicated to `peerA`

  ```
  rhpctl query series -server peerA
  ```

- The following command runs on `peerA` and lists the series have been configured on `peerA`

  ```
  rhpctl query series -rhpserver peerA
  ```

- The following command runs on all servers and lists the series that each server has replicated on `peerA`

  ```
  rhpctl query series -server peerA -rhpserver .+'
  ```

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles
role=user_name[+user_name...][,role=user_name[+user_name...]...]
```

For information about granting roles with RHPCTL, refer to rhpctl grant role

**Example**

This command returns output similar to the following:

```
$ rhpctl query series

Image series: DB12_series
Image series: GRID_series
Image series: DB112_series
```

## rhpctl subscribe series

Subscribes a specific user to an image series.

**Syntax**

```
rhpctl subscribe series -series series_name [-user user_name [-client
cluster_name]]
```

### Parameters

**Table A-63 rhpctl subscribe series Command Parameters**

| Parameter | Description |
|---|---|
| `-series` *`series_name`* | Specify the image series to which you want to subscribe a user. |
| `-user` *`user_name`* | Specify an operating system user to whom you are subscribing the image series. |
| `-client` *`cluster_name`* | Optionally, you can specify the name of the client cluster to which the operating system user belongs. |

## rhpctl unsubscribe series

Unsubscribes a user from an image series.

### Syntax

```
rhpctl unsubscribe series -series series_name [-user user_name [-client
cluster_name]]
```

### Parameters

**Table A-64 rhpctl unsubscribe series Command Parameters**

| Parameter | Description |
|---|---|
| `-series` *`series_name`* | Specify the image series from which you want to unsubscribe a user. |
| `-user` *`user_name`* | Specify an operating system user from whom you are unsubscribing the image series. |
| `-client` *`cluster_name`* | Optionally, you can specify the name of the client cluster to which the operating system user belongs. |

# server Commands

Use commands with the `server` keyword to export, register, unregister, and query Oracle Fleet Patching and Provisioning Server.

- rhpctl export server
- rhpctl query server
  Displays the configuration of a server.
- rhpctl register server
- rhpctl unregister server

## rhpctl export server

Exports data from the repository to a Fleet Patching and Provisioning Server data file.

### Syntax

```
rhpctl export server -server peer_server_name -serverdata file_path
```

### Usage Notes

- Specify the name of a peer server cluster.
- Specify the path to the file containing the Fleet Patching and Provisioning Server data.

## rhpctl query server

Displays the configuration of a server.

### Syntax

```
rhpctl query server
```

### Usage Notes

This command has no parameters.

### Example

This command displays output similar to the following:

```
$ rhpctl query server

Fleet Patching and Provisioning Server (RHPS): rhps-myserver
Storage base path: /u01/app/RHPImages
Disk Groups: RHPDATA
Port number: 8896
```

## rhpctl register server

Registers the specific Fleet Patching and Provisioning Server as a peer server.

### Syntax

```
rhpctl register server -server server_cluster_name -serverdata file
  {-root | -cred cred_name | -sudouser sudo_username -sudopath
path_to_sudo_binary
   | -auth plugin_name [-arg1 name1:value1 [-arg2 name2:value2 ...]]}
```

### Parameters

**Table A-65 rhpctl register server Command Parameters**

| Parameter | Description |
|---|---|
| `-server server_cluster_name` | Specify the name of the Fleet Patching and Provisioning Server cluster that you want to register. |

**Table A-65 (Cont.) rhpctl register server Command Parameters**

| Parameter | Description |
|---|---|
| `-serverdata` *file* | Specify the path to the file containing the Fleet Patching and Provisioning Server data. |
| `-root \| -cred` *cred_name* `\| -sudouser` *sudo_user_name* `-sudopath` *sudo_binary_location* `\| -auth` *plugin_name plugin_args* | Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |

## rhpctl unregister server

Unregisters a specific Fleet Patching and Provisioning Server as a peer server.

### Syntax

```
rhpctl unregister server -server server_cluster_name [-force]
```

### Usage Notes

- Specify the name of the Fleet Patching and Provisioning Server you want to unregister as a peer.
- Optionally, you can use the `-force` parameter to forcibly unregister the server.

# tag Commands

Use commands with the `tag` keyword to add, query, modify, delete, set, and unset tags.

- rhpctl add tag
  Creates a new tag for the Oracle Fleet Patching and Provisioning (Oracle FPP) entities.
- rhpctl query tag
  Displays a list of existing tags or details of a specific tag of Oracle FPP entities.
- rhpctl modify tag
  Modifies the specified tag of Oracle FPP entities.
- rhpctl delete tag
  Deletes the specified tag of Oracle FPP entities.
- rhpctl set tags
  Sets the listed tags for each Oracle FPP entity.
- rhpctl unset tags
  Unsets the listed tags for each Oracle FPP entity.

## rhpctl add tag

Creates a new tag for the Oracle Fleet Patching and Provisioning (Oracle FPP) entities.

### Syntax

```
rhpctl add tag -tag tag_name [-description tag_description]
```

### Parameters

**Table A-66 rhpctl add tag Command Parameters**

| Parameter | Description |
|---|---|
| `-tag tag_name` | Specify the name of the tag you want to add. |
| `-description tag_description` | Optionally, provide a description of the tag you are adding. |

## rhpctl query tag

Displays a list of existing tags or details of a specific tag of Oracle FPP entities.

### Syntax

```
rhpctl query tag -tag tag_name
```

### Parameters

**Table A-67 rhpctl query tag Command Parameters**

| Parameter | Description |
|---|---|
| `-tag tag_name` | Specify the name of the tag you want to add. |

## rhpctl modify tag

Modifies the specified tag of Oracle FPP entities.

### Syntax

```
rhpctl modify tag -tag tag_name [-description tag_description]
```

### Parameters

**Table A-68 rhpctl modify tag Command Parameters**

| Parameter | Description |
|---|---|
| `-tag tag_name` | Specify the name of the tag you want to modify. |
| `-description tag_description` | Optionally, provide a description of the tag you are modifying. |

## rhpctl delete tag

Deletes the specified tag of Oracle FPP entities.

### Syntax

```
rhpctl delete tag -tag tag_name [-force]
```

### Parameters

**Table A-69 rhpctl delete tag Command Parameters**

| Parameter | Description |
|---|---|
| `-tag tag_name` | Specify the name of the tag you want to delete. |
| `-force` | Optionally, you can specify this parameter to delete the tag forcibly. |

## rhpctl set tags

Sets the listed tags for each Oracle FPP entity.

### Syntax

```
rhpctl set tags -tags tag_list -targettype DB|SITE
-targets <<site1:db1,db2,..>::<site2:db1,db2,...>...>|<id1,id2...>
```

### Parameters

**Table A-70 rhpctl set tag Command Parameters**

| Parameter | Description |
|---|---|
| `-tags tag_list` | Comma-separated list of the Oracle FPP tags. |
| `-targettype {DB\|SITE}` | Oracle FPP target for which you want to set the tag. |
| `-targets <<site1:db1,db2,..>::<site2:db1,db2,...>...>\|<id1,id2...>` | A list of target IDs, which include site names for Oracle Database IDs. |

### Examples

To set tag for a site:

```
rhpctl set tags -tags tag1,tag2 -targettype SITE -targets site1
```

To set tag for a database:

```
rhpctl set tags -tags tag1,tag2 -targettype DB -targets dbwc4
```

## rhpctl unset tags

Unsets the listed tags for each Oracle FPP entity.

### Syntax

```
rhpctl unset tags -tags tag_list -targettype DB|SITE
-targets <<site1:db1,db2,..>::<site2:db1,db2,...>...>|<id1,id2...>
```

### Parameters

**Table A-71 rhpctl unset tag Command Parameters**

| Parameter | Description |
|---|---|
| `-tags tag_list` | Comma-separated list of the Oracle FPP tags. |
| `-targettype {DB\|SITE}` | Oracle FPP target for which you want to unset the tag. |
| `-targets <<site1:db1,db2,..>::<site2:db1,db2,...>...>\|<id1,id2...>` | A list of target IDs, which include site names for Oracle Database IDs. |

# user Commands

Use commands with the `user` keyword to delete, modify, register, and unregister users.

- rhpctl delete user
- rhpctl modify user
- rhpctl register user
- rhpctl unregister user

## rhpctl delete user

Deletes a user from the Fleet Patching and Provisioning repository.

### Syntax

```
rhpctl delete user -user user_name [-client cluster_name]
```

### Parameters

**Table A-72 rhpctl delete user Command Parameters**

| Parameter | Description |
|---|---|
| `-user user_name` | Specify the name of the user you want to delete from a Fleet Patching and Provisioning Client. |
| `-client cluster_name` | Optionally, you can specify the name of the client cluster from which you want to delete from a specific user. |

### Usage Notes

- You can delete non built-in users only if that user does not own any working copies.
- If the user created an image or image series, then you can still delete the user, but the creator of the image or image series is changed to *internal-user@GHS*.

- If the user was the owner of an image series, then you can delete the user, but the owner of the image series will be changed to *internal-user@GHS*. You can still use the affected image series as normal, such that you can still provision a working copy from the affected image series, and you can still insert or delete images from the affected image series.

**Example**

The following example deletes the user named `scott` on the server cluster from the Fleet Patching and Provisioning repository:

```
$ rhpctl delete user -user scott
```

## rhpctl modify user

Modifies the email address of a specific user.

**Syntax**

```
rhpctl modify user -user user_name -email email_address [-client client_name]
```

**Parameters**

**Table A-73 rhpctl modify user Command Parameters**

| Parameter | Description |
|---|---|
| `-user user_name` | Specify an operating system user whose email address you want to modify. |
| `-email email_address` | Specify the email address of the operating system user in the RFC 822 format. |
| `-client client_name` | Optionally, you can specify the name of the client cluster to which the operating system user belongs. |

## rhpctl register user

Registers an email address for a specific user.

**Syntax**

```
rhpctl register user -user user_name -email email_address [-client
client_name]
```

**Parameters**

**Table A-74 rhpctl register user Command Parameters**

| Parameter | Description |
|---|---|
| `-user user_name` | Specify an operating system user whose email address you want to register. |
| `-email email_address` | Specify the email address of the operating system user in the RFC 822 format. |

**Table A-74 (Cont.) rhpctl register user Command Parameters**

| Parameter | Description |
|---|---|
| `-client client_name` | Optionally, if you run the command on the Fleet Patching and Provisioning Server, then you can specify the name of the client cluster to which the operating system user belongs. Otherwise, the command applies to a user on the cluster (either the Fleet Patching and Provisioning Server or Client) where the command is run. |

**Example**

An example of this command is:

```
$ rhpctl register user -user scott -email scott@example.com
```

## rhpctl unregister user

Unregisters an email address for a specific user.

**Syntax**

```
rhpctl unregister user -user user_name [-client client_name]
```

**Parameters**

**Table A-75 rhpctl unregister user Command Parameters**

| Parameter | Description |
|---|---|
| `-user user_name` | Specify an operating system user whose email address you want to unregister. |
| `-client client_name` | Optionally, you can specify the name of the client cluster to which the operating system user belongs. |

# useraction Commands

Use commands with the `useraction` keyword to add, delete, and modify user actions.

- rhpctl add useraction
  Configures a user action and its associated script and action file.
- rhpctl delete useraction
- rhpctl modify useraction
  Modifies the configuration of the specified user action name.
- rhpctl query useraction
  Displays the configuration of a user action.

## rhpctl add useraction

Configures a user action and its associated script and action file.

### Syntax

```
rhpctl add useraction -useraction user_action_name -actionscript script_name
  [-actionfile file_name] [-pre | -post | -eval] [-optype option] [-phase
operation_phase]
  [-onerror {ABORT | CONTINUE}] [-runscope {ONENODE | ALLNODES | AUTO |
FIRSTNODEONRHPS | LASTNODEONRHPS | ALLNODESONRHPS}]
  [-runasroot] [-runninginstance]
```

### Parameters

**Table A-76 rhpctl add useraction Command Parameters**

| Parameter | Description |
|---|---|
| `-useraction` *user_action_name* | Specify the name of the user action you want to add. |
| `-actionscript` *script_name* | Associate a specific action script to run with the user action. |
| `-actionfile` *file_name* | Optionally, you can specify an action file that is required by the user action. |
| `-pre \| -post \| -eval` | Use `-pre` to run the user action before the operation, use `-post` to run the user action after the operation, or use `-eval` to run the user action during evaluation. |
| `-optype` *option* | Optionally, you can specify the operation for which the user action is configured. Options include:<br>IMPORT_IMAGE<br>ADD_WORKINGCOPY<br>DELETE_WORKINGCOPY<br>ADD_DATABASE<br>DELETE_DATABASE<br>MOVE_DATABASE<br>ADD_PDB_DATABASE<br>DELETE_PDB_DATABASE<br>MOVE_GIHOME<br>UPGRADE_DATABASE<br>UPGRADE_GIHOME<br>ADDNODE_GIHOME<br>DELETENODE_GIHOME<br>ADDNODE_DATABASE<br>DELETENODE_DATABASE<br>ADDNODE_WORKINGCOPY<br>ZDTUPGRADE_DATABASE<br>ZDTUPGRADE_DATABASE_SNAPDB<br>ZDTUPGRADE_DATABASE_DBUA<br>ZDTUPGRADE_DATABASE_SWITCHBACK<br>MIGRATE_DATABASE<br>UPDATE_EXADATA |
| `-phase` | Specify the phase of the operation for which the user action is configured. |
| `-onerror {ABORT \| CONTINUE}` | Optionally, you can choose whether to stop or continue the operation if the user action encounters an error while it is running. |

**Table A-76 (Cont.) rhpctl add useraction Command Parameters**

| Parameter | Description |
|---|---|
| `-runscope {ONENODE \| ALLNODES \| AUTO \| FIRSTNODEONRHPS \| LASTNODEONRHPS \| ALLNODESONRHPS}` | Optionally, you can specify the nodes where the user action is run. Choose `ONENODE` to run the user action for each database on the node on which a patch was applied to the database. Choose `ALLNODES` to run the user action for each database on every cluster node. Choose `AUTO` for a run scope based on the other command options.<br>Specify `FIRSTNODEONRHPS`, `LASTNODEONRHPS`, or `ALLNODESONRHPS` to run the user action script on FPPS when the operation is performed on first node or last node or all nodes. |
| `-runasroot` | Run the `rhpctl add useraction` command as `root` user. |
| `-runninginstance` | Specify this parameter to run the user action only on the nodes that have a running database instance. You can use this parameter only with the `MOVE_DATABASE`, `MOVE_GIHOME`, and `UPGRADE_DATABASE` operation types. |

## rhpctl delete useraction

Deletes an existing user action configuration.

### Syntax

```
rhpctl delete useraction -useraction user_action_name
```

### Usage Notes

Specify the name of a user action you want to delete.

## rhpctl modify useraction

Modifies the configuration of the specified user action name.

### Syntax

```
rhpctl modify useraction -useraction user_action_name [-actionscript
script_name]
  [-actionfile file_name] [-pre | -post | -eval] [-optype option] [-onerror
{ABORT | CONTINUE}]
  [-runscope {ONENODE | ALLNODES | AUTO | FIRSTNODEONRHPS | LASTNODEONRHPS |
ALLNODESONRHPS}] [-runninginstance]
```

### Parameters

**Table A-77 rhpctl modify useraction Command Parameters**

| Parameter | Description |
|---|---|
| `-useraction user_action_name` | Specify the name of the user action you want to modify. |
| `-actionscript script_name` | Optionally, you can specify an action script to run. |

**Table A-77 (Cont.) rhpctl modify useraction Command Parameters**

| Parameter | Description |
|---|---|
| `-pre \| -post \| -eval` | Use `-pre` to run the user action before the operation, use `-post` to run the user action after the operation, or use `-eval` to run the user action during evaluation. |
| `-optype` *option* | Optionally, you can specify the operation for which the user action is configured. Options include:<br>IMPORT_IMAGE<br>ADD_WORKINGCOPY<br>DELETE_WORKINGCOPY<br>ADD_DATABASE<br>DELETE_DATABASE<br>MOVE_DATABASE<br>ADD_PDB_DATABASE<br>DELETE_PDB_DATABASE<br>MOVE_GIHOME<br>UPGRADE_DATABASE<br>UPGRADE_GIHOME<br>ADDNODE_GIHOME<br>DELETENODE_GIHOME<br>ADDNODE_DATABASE<br>DELETENODE_DATABASE<br>ADDNODE_WORKINGCOPY<br>ZDTUPGRADE_DATABASE<br>ZDTUPGRADE_DATABASE_SNAPDB<br>ZDTUPGRADE_DATABASE_DBUA<br>ZDTUPGRADE_DATABASE_SWITCHBACK<br>MIGRATE_DATABASE<br>UPDATE_EXADATA |
| `-onerror {ABORT \| CONTINUE}` | Optionally, you can choose whether to stop or continue the operation if the user action encounters an error while it is running. |
| `-runscope {ONENODE \| ALLNODES \| AUTO \| FIRSTNODEONRHPS \| LASTNODEONRHPS \| ALLNODESONRHPS}` | Optionally, you can specify the nodes where the user action is run. Optionally, you can specify the nodes where the user action is run. Choose `ONENODE` to run the user action for each database on the node on which a patch was applied to the database. Choose `ALLNODES` to run the user action for each database on every cluster node. Choose `AUTO` for a run scope based on the other command options.<br>Specify `FIRSTNODEONRHPS`, `LASTNODEONRHPS`, or `ALLNODESONRHPS` to run the user action script on FPPS when the operation is performed on first node or last node or all nodes. |
| `-runninginstance` | Specify this parameter to run the user action only on the nodes that have a running database instance. You can use this parameter only with the `MOVE_DATABASE`, `MOVE_GIHOME`, and `UPGRADE_DATABASE` operation types. |

## rhpctl query useraction

Displays the configuration of a user action.

### Syntax

```
rhpctl query useraction [-useraction user_action_name | -imagetype image_type]
  [-optype option]
```

### Parameters

**Table A-78 rhpctl query useraction Command Parameters**

| Parameter | Description |
|---|---|
| `-useraction` *user_action_name* | Specify the name of the user action you want to query. |
| `-imagetype` *image_type* | Specify the software type. Use `ORACLEDBSOFTWARE` (default) for Oracle database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, `ODAPATCHSOFTWARE` for engineered systems (Oracle Data Appliance), `EXAPATCHSOFTWARE` for Oracle Exadata software, or `SOFTWARE` for all other software. For a custom image type, use the image type name. |
| `-optype` *option* | Optionally, you can specify the operation for which to run the query. Options include:<br>IMPORT_IMAGE<br>ADD_WORKINGCOPY<br>DELETE_WORKINGCOPY<br>ADD_DATABASE<br>DELETE_DATABASE<br>MOVE_DATABASE<br>ADD_PDB_DATABASE<br>DELETE_PDB_DATABASE<br>MOVE_GIHOME<br>UPGRADE_DATABASE<br>UPGRADE_GIHOME<br>ADDNODE_GIHOME<br>DELETENODE_GIHOME<br>ADDNODE_DATABASE<br>DELETENODE_DATABASE<br>ADDNODE_WORKINGCOPY<br>ZDTUPGRADE_DATABASE<br>ZDTUPGRADE_DATABASE_SNAPDB<br>ZDTUPGRADE_DATABASE_DBUA<br>ZDTUPGRADE_DATABASE_SWITCHBACK<br>MIGRATE_DATABASE<br>UPDATE_EXADATA |

## workingcopy Commands

Use commands with the `workingcopy` keyword to create, update, extend, and delete working copies.

- rhpctl add workingcopy
  Creates a working copy on a client cluster.
- rhpctl addnode workingcopy

- rhpctl delete workingcopy
  Deletes an existing working copy.
- rhpctl query workingcopy
  Displays the configuration information of an existing working copy.
- rhpctl register workingcopy
  Registers working copy metadata to the Oracle FPP repository.

## rhpctl add workingcopy

Creates a working copy on a client cluster.

**Syntax**

> **Note:**
>
> The Highly Available Grid Naming Service feature of Grid Naming Service (GNS) in Oracle Grid Infrastructure is deprecated in Oracle Database 23ai.

To add a working copy to a client cluster:

```
rhpctl add workingcopy -workingcopy workingcopy_name
  {-image image_name | -series series_name}
  [-oraclebase oraclebase_path] [-path where_path]
  [-localmount [-location zipped_home_path] [-zip zipped_home_path] [-
transfer_as_zip] [-storagetype {LOCAL | RHP_MANAGED}]
  [-user user_name] [-dbname unique_db_name
           [-dbtype {RACONENODE | RAC | SINGLE}] [-datafileDestination
datafileDestination_path]
           [-dbtemplate { file_path | image_name:relative_file_path}]
           {-node node_list |}
           [-cdb] [-pdbName pdb_prefix [-numberOfPDBs pdb_count]]]
  [-client cluster_name] [-clusternamealias cluster_name_alias] [-
ignoreprereq | -fixup]
  [-responsefile response_file_path] [-clusternodes node_list]
  [-groups group_list]
  [-root | -cred cred_name | -sudouser sudo_username
    -sudopath path_to_sudo_binary | -auth plugin_name [-arg1 name1:value1
    [-arg2 name2:value2 ...]]]
  [-notify [-cc users_list]]
  [-asmclientdata data_path] [-softwareonly]
  [-local] [-inventory inventory_path] [-targetnode target_node_name]
  [-agpath read_write_path -aupath gold_image_path] [-setupssh]
  [-useractiondata user_action_data] [-eval] [-schedule {timer_value | NOW |
PAUSE}] [-jobtag tag_name]
  [-tags tags_criteria] [-checkwcpatches -sourcehome source_home_path] [-scan
scan_name]
  [-diskDiscoveryString disk_discovery_string] [-readonly]
```

### Parameters

**Table A-79 rhpctl add workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy` *workingcopy_name* | Specify a name for the working copy that you want to create. |
| {`-image` *image_name* \| `-series` *series_name*} | Specify the name of a configured image from which to create a working copy or the name of an image series from which RHPCTL takes the latest image when adding a working copy. |
| `-oraclebase` *oracle_base_path* | Specify an `ORACLE_BASE` path for provisioning an Oracle Database or Oracle Grid Infrastructure home. You can specify either an existing directory or a new directory.<br>**Note:** This parameter is required only for the `ORACLEDBSOFTWARE` and `ORACLEGISOFTWARE` image types. |
| `-inventory` *inventory_path* | Specify the location of the Oracle Inventory directory. |
| `-path` *absolute_path* | Specify the absolute path for provisioning the software home on the client side (this location must be empty). For Oracle Database images, this becomes the `ORACLE_HOME`.<br>**Note:** This parameter is required for `LOCAL` storage types, and is invalid for `RHP_MANAGED`. |
| `-localmount` | Specify this option to provision the working copy using the locally mounted compressed image file. |
| `-location` *zipped_home_path* | Specify the location of the compressed image file on the destination cluster. |
| `-zip` *zipped_home_path* | Specify the absolute path of the compressed software home. |
| `-transfer_as_zip` | Specify this option to transfer the working copy to the destination host as a zip file. This parameter works with the `-image` parameter. |
| `-storagetype` {`LOCAL` \| `RHP_MANAGED`} | Specify the type of storage for the software home. |
| `-user` *user_name* | Specify the name of the user who will own the working copy being provisioned.<br>If you do not specify this parameter, then the working copy is owned by the user running the command. If you are provisioning to a remote cluster, then the user name must be a valid user on the remote cluster. The user ID need not be the same between the two clusters, but the user name must exist on both.<br>**Note:** You cannot use `-user` simultaneously with the `-softwareonly` parameter. |
| `-dbname` *unique_db_name* | Specify the unique name of the database (`DB_UNIQUE_NAME` without `DB_DOMAIN`) that you are adding. |
| `-dbtype` {`RACONENODE` \| `RAC` \| `SINGLE`} | Specify whether the database is Oracle RAC One Node, Oracle RAC, or a nonclustered database. |
| `-datafileDestination` *datafileDestination_path* | Specify the data file destination location or the name of the Oracle Automatic Storage Management (Oracle ASM) disk group.<br>**Note:** You cannot specify a disk group for Oracle Database versions before Oracle Database 11*g* release 2 (11.2). |
| `-dbtemplate` *file_path* \| *image_name*`:`*relative_file_path* | Specify the absolute file path to a database template or the relative path to the image home directory on a Fleet Patching and Provisioning Server. |
| `-node` *node_list* | Specify a node or comma-delimited list of several nodes.<br>Enter a node name for a single-instance Oracle home. |
| `-cdb` | Optionally, use this parameter to create a database as a Container Database (CDB). |

**Table A-79 (Cont.) rhpctl add workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-pdbName` *pdb_prefix* | If you are creating one or more Pluggable Databases (PDBs), then specify a PDB name prefix. |
| `-numberOfPDBs` *pdb_count* | Specify the number of PDBs you want to create. |
| `-client` *cluster_name* | Specify the name of the client cluster.<br><br>**Note:**<br>Oracle recommends that you specify a unique name for the client cluster. |
| `-clusternamealias` | Optionally, you can specify the client cluster alias if the client cluster name is not unique. |
| `-ignoreprereq \| -fixup` | You can choose to ignore the Clusterware Verification Utility (CVU) checks or you can choose to run the recommended fixup script.<br>**Note:** These parameters are valid only when you are provisioning Oracle Grid Infrastructure. |
| `-responsefile` *response_file_path* | Specify a response file to use when you provision Oracle Grid Infrastructure. |
| `-clusternodes` *node_name*`:`*node_vip*`[:`*node_role*`]`<br>`[,`*node_name*`:`*node_vip*`[:`*node_role*`]...]` | Specify a comma-delimited list of cluster node information on which to provision Oracle Clusterware. |
| `-groups "OSDBA\|OSOPER\|OSASM\|OSBACKUP\|OSDG\|OSKM\|OSRAC=`*group_name*`[,...]"` | Specify a comma-delimited list of Oracle groups, enclosed in double quotation marks (""), that you want to configure in the working copy.<br>For example:<br>`-groups "OSDBA=dba,OSOPER=oper"`<br>When you create a gold image from a source home or working copy, the gold image inherits the groups configured in the source. When you create a working copy from that gold image using `rhpctl add workingcopy`, by default, the new working copy inherits the same groups as the gold image.<br>If you use the `-groups` parameter on the command line, then:<br>• Groups configured in the gold image that you do not specify on the command line are inherited by the working copy.<br>• Groups configured in the gold image that you also specify on the command line are set to the value that you specify on the command line (command line parameters override the gold image).<br>• Groups that you specify on the command line that are not in the gold image are added to the configured groups in the gold image (the command line adds new groups).<br>**Notes:**<br>• When you move or upgrade a source home (unmanaged or working copy), the groups in the destination working copy must match those of the source home.<br>• You cannot use `-groups` simultaneously with the `-softwareonly` parameter. |

**Table A-79 (Cont.) rhpctl add workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-root \| -cred cred_name \| -sudouser sudo_user_name -sudopath sudo_binary_location \| -auth plugin_name plugin_args` | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-notify [-cc user_list]` | Specify this parameter to have email notifications sent to the owner of the working copy. Optionally, you can include a list of additional users who will receive notifications. |
| `-asmclientdata data_path` | Specify the path to a file that contains Oracle ASM client data. |
| `-local` | Use this parameter to provision only Oracle Grid Infrastructure software on the local node.<br>**Note:** You can only use this parameter in conjunction with the `-softwareonly` parameter, and only when running the `rhpctl add workingcopy` command on a Fleet Patching and Provisioning Server. |
| `-softwareonly` | Use this parameter to provision only Oracle Grid Infrastructure software. |
| `-targetnode target_node_name` | Specify the name of an `rhpclient`-less target. |
| `-agpath read_write_path -aupath gold_image_path` | Use `-agpath` to specify the path to the read-write, site-specific configuration changes to set the persistent home path, and use `-aupath` to specify the path for the read-only gold image to set the persistent home path. |
| `-setupssh` | Use this parameter to set up passwordless SSH user equivalence on the remote nodes for the provisioning user. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-eval` | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-tags tags_criteria` | Optionally, specify tags criteria to create child jobs internally in the PAUSED state. Use the `rhpctl resume job` command to continue the operations of all the child jobs. |
| `-checkwcpatches -sourcehome source_home_path` | Optionally, you can use the `-checkwcpatches` and `-sourcehome` parameters to compare patches in a specific source home path with the patches in the working copy you want to add. |
| `-scan scan_name` | Optionally, you can use this parameter to specify a SCAN name. |
| `-diskDiscoveryString disk_discovery_string` | Optionally, you can use this parameter to specify a disk discovery string. |

**Table A-79 (Cont.) rhpctl add workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-readonly` | Optionally, you can use this parameter to add the database working copy as a read-only home. |

### Usage Notes

> **Note:**
>
> Member Clusters, which are part of the Oracle Cluster Domain architecture, are desupported in Oracle Grid Infrastructure 21c.

> **Note:**
>
> Starting with Oracle Grid Infrastructure 23ai, Domain Services Clusters (DSC), which is part of the Oracle Cluster Domain architecture, are desupported.
> Oracle Cluster Domains consist of a Domain Services Cluster (DSC) and Member Clusters. Member Clusters were deprecated in Oracle Grid Infrastructure 19c. The DSC continues to be available to provide services to production clusters. However, with most of those services no longer requiring the DSC for hosting, installation of DSCs are desupported in Oracle Database 23ai. Oracle recommends that you use any cluster or system of your choice for services previously hosted on the DSC, if applicable. Oracle will continue to support the DSC for hosting shared services, until each service can be used on alternative systems.

- You can obtain context sensitive help for specific use cases for the `rhpctl add workingcopy` command, as follows:

```
$ rhpctl add workingcopy -help [REMOTEPROVISIONING | STORAGETYPE | ADMINDB
  | GRIDHOMEPROV | SWONLYGRIDHOMEPROV  | STANDALONEPROVISIONING |
GGHOMEPROVISIONING]
```

- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

### Examples

- To create a working copy on a client cluster for yourself or another user:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
   -series series_name} -oraclebase oracle_base_path -client cluster_name
  [-user user_name]
```

- To create a working copy on storage that you specify:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
   -series series_name} -oraclebase oracle_base_path -storagetype
  {LOCAL | RHP_MANAGED} [-path absolute_path]
```

- To create and configure a working copy of Oracle Grid Infrastructure:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
   -series series_name} {-root | -cred cred_name | -sudouser sudo_user_name
   -sudopath sudo_binary_path} -responsefile response_file_path
   [-clusternodes node_information] [-user user_name] [-oraclebase oracle_base_path]
   [-path absolute_path] [-asmclientdata data_path]
   [-ignoreprereq | -fixup]
```

- To provision a software-only working copy of Oracle Grid Infrastructure:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
   -series series_name} -softwareonly -path Grid_home_path -oraclebase
   oracle_base_path [-local | -client cluster_name
   [-groups "Oracle_group=user_group[,...]"] [-node client_node_name] |
   {-root | -cred cred_name | -sudouser sudo_user_name -sudopath sudo_binary_path}
    -targetnode node_name]
```

- To provision a working copy on a node or a cluster where Oracle Fleet Patching and Provisioning does not exist:

```
rhpctl add workingcopy -workingcopy workingcopy_name {-image image_name |
   -series series_name} -oraclebase oracle_base_path -user user_name
    -node node_name [-path absolute_path]
    {-root | -cred cred_name | -sudouser sudo_user_name -sudopath sudo_binary_path}
```

> **Note:**
>
> If you are provisioning Oracle database software to a Fleet Patching and Provisioning Client that has been configured with an Oracle ASM disk group, then do not specify the `-path` parameter, so as to enable the Fleet Patching and Provisioning Client to use storage provided by Fleet Patching and Provisioning.
>
> If the Fleet Patching and Provisioning Client is not configured with an Oracle ASM disk group, then specify the `-storagetype` parameter with `LOCAL`, in addition to specifying the `-path` parameter.

## rhpctl addnode workingcopy

Extends an Oracle RAC database to another node or nodes in a cluster.

**Syntax**

```
rhpctl addnode workingcopy -workingcopy workingcopy_name -node node_list
  [-targetnode node_name {-root | -sudouser sudo_username -sudopath
sudo_binary_path
```

```
   | -cred cred_name | -auth plugin_name [-arg1 name1:value1...]} -setupssh]
  [-ignoreprereq] [-useractiondata user_action_data] [-eval] [-schedule
{timer_value | NOW | PAUSE}] [-jobtag tag_name]
```

### Parameters

**Table A-80 rhpctl addnode workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of a working copy that contains the Oracle database you want to extend. |
| `-node node_list` | Specify a node or a comma-delimited list of nodes to which you want to extend the database. |
| `-targetnode node_name` | Optionally, you can specify an `rhpclient`-less target on which to run this command. |
| `-root \| -sudouser sudo_username -sudopath sudo_binary_path \| -cred cred_name \| -auth plugin_name [-arg1 name1:value1...]` | If you choose to use the `-targetnode` parameter, then you must choose either `sudo` or `root` to access the remote nodes.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-ignoreprereq` | Use this parameter to ignore the CVU prerequisite checks. |
| `-setupssh` | Sets up passwordless SSH user equivalence on the remote nodes for the provisioning user. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |
| `-eval` | Optionally, you can use this parameter to evaluate the impact of this command on the system without actually running the command. |
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br>`2018-07-25T19:13:17+05`<br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |

### Usage Notes

- If you are extending an administrator-managed database, then you must also run the `rhpctl addnode database` command to start the instance.
- If the `rhpclient`-less target is an Oracle Clusterware 11*g* release 2 (11.2) or 12*c* release 1 (12.1) cluster, then you must provide either root credentials or provide a sudo user. You must also specify an `rhpclient`-less target that must be the node name of one of the cluster nodes.

# rhpctl delete workingcopy

Deletes an existing working copy.

### Syntax

```
rhpctl delete workingcopy -workingcopy workingcopy_name -targetnode node_name
  [-notify [-cc user_list]] [-force] {-root | -sudouser sudo_user_name -
sudopath sudo_binary_path
  -cred cred_name | -auth plugin_name [-arg1 name1:value1...]} [-
useractiondata user_action_data]
  [-schedule {timer_value | NOW | PAUSE}] [-jobtag tag_name] [-metadataonly]
```

### Parameters

**Table A-81 rhpctl delete workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of a working copy that you want to delete. |
| `-notify [-cc user_list]` | Name of a node in a remote cluster with no Fleet Patching and Provisioning Client. |
| `-targetnode node_name` | You must specify a target node when you delete an active working copy.<br>This parameter is optional when you delete a non-active software-only working copy.<br>**Note:**<br>Do not use this parameter in combination with the `metadataonly` parameter. |
| `-force` | Use this parameter to forcibly delete the database working copy. |
| `-root \| -sudouser sudo_username -sudopath sudo_binary_path \| -cred cred_name` | If you choose to use the `-targetnode` parameter, then you must choose either `sudo` or `root` to access the remote node.<br>If you choose `sudo`, then you must specify a user name to run super-user operations, and a path to the location of the `sudo` binary.<br>Optionally, you can choose to specify a credential name to associate the user and password credentials to access a remote node.<br>Alternative to `-sudouser`, `-root`, or `-cred`, you can use `-auth` to specify an authentication plugin to access a remote node. |
| `-useractiondata user_action_data` | Optionally, you can pass a value to the `useractiondata` parameter of the user action script. |

**Table A-81 (Cont.) rhpctl delete workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-schedule {timer_value \| NOW \| PAUSE}` | Optionally, you can use this parameter to schedule a time to run this operation, in ISO-8601 format, as in the following example:<br><br>`2018-07-25T19:13:17+05`<br><br>If `NOW` is specified or the option is omitted, then the job is scheduled immediately.<br>If `PAUSE` is specified, then the job starts in the paused state and you need to resume the job using the `rhpctl resume job -jobid job_id` command. |
| `-jobtag tag_name` | Optionally, you can associate a user-defined tag with the scheduled jobs. |
| `-metadataonly` | Use this parameter to delete only the working copy metadata, which is located in the metadata repository.<br><br>**Note:**<br>Do not use this parameter in combination with the `targetnode` parameter. |

### Usage Notes

- This command *will not* delete the working copy if there are any databases configured on it. Use the `-force` option to override this.
- This command *will not* not delete the working copy if there are any running databases on it. The `-force` option *will not* override this.
- This command *does not* delete the Oracle base that was created when you ran `rhpctl add workingcopy`.
- If you choose to use the `-schedule` parameter, then you must run this command on the Fleet Patching and Provisioning Server.

### Examples

To delete a working copy:

```
$ rhpctl delete workingcopy -workingcopy wc1
```

## rhpctl query workingcopy

Displays the configuration information of an existing working copy.

### Syntax

```
rhpctl query workingcopy [-workingcopy workingcopy_name [-details] [-
metadataonly] | -image image_name]
    [-drift] [-rhpserver <rhps_regex>] | [-client cluster_name [-path path | -
```

```
image image_name]] |
    [-imagetype image_type [-version image_version] [-client cluster_name]]
```

### Parameters

**Table A-82 rhpctl query workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-workingcopy workingcopy_name` | Specify the name of a working copy for which you want to display the configuration information. |
| `-metadataonly` | Use this paramter only when you use the `-workingcopyy` parameter to query only the metadata of the working copy, which is located in the repository and not run OPatch or connect to the target to query for extra information. |
| `-details` | Provide details of the move operation if the working copy was part of a move operation. |
| `-image image_name` | Alternatively, you can specify the name of a configured image you want to query.<br><br>**Note:**<br>If you specify an image name, then RHPCTL lists all the working copies based on that image. |
| `-drift` | List the the bug fixes not included in the golden image. |
| `-client cluster_name` | Optionally, you can specify a client cluster on which to query working copies. |
| `-path path` | Location of the working copy. |
| `-rhpserver rhps_regex` | Regular expression to match the cluster name of the servers where you want to run the command. |
| `-imagetype image_type` | Specify the software type. `ORACLEDBSOFTWARE` (default) for Oracle Database software, `ORACLEGISOFTWARE` for Oracle Grid Infrastructure software, `ORACLEGGSOFTWARE` for Oracle GoldenGate software, `LINUXOS` for Linux operating system ISO, or `SOFTWARE` for all other software. If you use custom image types, then specify the name of your image type. |
| `-version image_version` | Version of the image associated with the working copy. |

### Usage Notes

When issuing a command for a peer server using the `-rhpserver` option, the user running the command must be an existing user of the peer server and the user must have a required role. To enable a user from a peer server to run commands on the local server, run the `rhpctl`

`grant role` command to grant a required role to the peer server user and to specify the cluster name of the peer server to which the user belongs. For example:

```
$ rhpctl grant role -role role_name -user user_name -client cluster_name
```

To add multiple users, run the following command:

```
$ rhpctl grant role -client cluster_name -maproles
role=user_name[+user_name...][,role=user_name[+user_name...]...]
```

For information about granting roles with RHPCTL, refer to rhpctl grant role

## rhpctl register workingcopy

Registers working copy metadata to the Oracle FPP repository.

**Syntax**

```
rhpctl register workingcopy {-attrfile attribute_file | -workingcopy
workingcopy_name
   {-client cluster_name | -targetnode target_node_name {-root | -cred
cred_name |
   -sudouser sudo_username -sudopath path_to_sudo_binary | -auth plugin_name
   [-arg1 name1:value1 [-arg2 name2:value2 ...]]} [-sitetype site_type]}
   [-softwareonly]} -image image_name -path home_path [-groups key=value
[,key=value...]]
   [-user user_name] [-oraclebase oraclebase_path] [-node node_list]
```

**Parameters**

**Table A-83 rhpctl register workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-attrfile attribute_file` | Absolute file name on the Oracle FPP Server which provides information about the working copy. |
| `-workingcopy workingcopy_name` | Specify a name for the working copy that you want to register. |
| `-client cluster_name` | Specify the name of the client cluster.<br><br>**Note:**<br>Oracle recommends that you specify a unique name for the client cluster. |
| `-targetnode target_node_name` | Specify the name of an `rhpclient`-less target. |

**Table A-83 (Cont.) rhpctl register workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-root \| -cred cred_name \| -sudouser sudo_user_name -sudopath sudo_binary_location \| -auth plugin_name plugin_args` | If you choose to use the `-targetnode` parameter, then you must choose either `root`, a credential name, `sudo`, or an authentication plugin to access the remote node.<br>Choose `-root` to perform super user operations as `root`. Alternatively, you can choose either to specify a credential name to associate the user name and password credentials to access a remote node, to perform super user operations as a `sudo` user by specifying a `sudo` user name and the path to the `sudo` binary, or to use an authentication plugin to access the remote node. |
| `-sitetype {ORACLERESTART \| STANDALONE}` | Specify the type of a site that has no Oracle FPP client. The default value is `STANDALONE`. |
| `-softwareonly` | Use this parameter to provision only Oracle Grid Infrastructure software. |
| `-image image_name` | Specify the name of a configured image from which to register a working copy or the name of an image series from which RHPCTL takes the latest image when adding a working copy. |
| `-path absolute_path` | Specify the absolute path for provisioning the software home. For Oracle Database images, this becomes the `ORACLE_HOME`. |
| `-groups "OSDBA\|OSOPER\|OSASM\|OSBACKUP\|OSDG\|OSKM\|OSRAC=group_name[,...]"` | Specify a comma-delimited list of Oracle groups, enclosed in double quotation marks (""), that you want to configure in the working copy.<br>For example:<br>`-groups "OSDBA=dba,OSOPER=oper"`<br>When you create a gold image from a source home or working copy, the gold image inherits the groups configured in the source. When you create a working copy from that gold image using `rhpctl add workingcopy`, by default, the new working copy inherits the same groups as the gold image.<br>If you use the `-groups` parameter on the command line, then:<br>• Groups configured in the gold image that you do not specify on the command line are inherited by the working copy.<br>• Groups configured in the gold image that you also specify on the command line are set to the value that you specify on the command line (command line parameters override the gold image).<br>• Groups that you specify on the command line that are not in the gold image are added to the configured groups in the gold image (the command line adds new groups).<br>**Notes:**<br>• When you move or upgrade a source home (unmanaged or working copy), the groups in the destination working copy must match those of the source home.<br>• You cannot use `-groups` simultaneously with the `-softwareonly` parameter. |
| `-user user_name` | Specify the name of the user who will own the working copy being provisioned.<br>If you do not specify this parameter, then the working copy is owned by the user running the command. If you are provisioning to a remote cluster, then the user name must be a valid user on the remote cluster. The user ID need not be the same between the two clusters, but the user name must exist on both.<br>**Note:** You cannot use `-user` simultaneously with the `-softwareonly` parameter. |
| `-oraclebase oracle_base_path` | Specify an `ORACLE_BASE` path for provisioning an Oracle Database or Oracle Grid Infrastructure home. You can specify either an existing directory or a new directory.<br>**Note:** This parameter is required only for the `ORACLEDBSOFTWARE` and `ORACLEGISOFTWARE` image types. |

**Table A-83 (Cont.) rhpctl register workingcopy Command Parameters**

| Parameter | Description |
|---|---|
| `-node node_list` | Specify a node or comma-delimited list of several nodes.<br>Enter a node name for a single-instance Oracle home. |

### Examples

- To register an Oracle Database working copy:

```
$ rhpctl register workingcopy -workingcopy DBWC2304 -client client3
-image DB_2304 -path /u01/app/oracle/product/23.0.0/dbhome1 -user oracle
```

- To register an Oracle Grid Infrastructure working copy:

```
$ rhpctl register workingcopy -workingcopy GIWC2304 -client client3
 -image GI_2304 -path /u01/app/23.0.0/grid
```

- To register a software-only Oracle Grid Infrastructure working copy:

```
$ rhpctl register workingcopy -workingcopy SFWC2304 -client client4
-image GISF2304 -path /u01/app/23.0.0/grid -softwareonly
```