Oracle Linux 10

Upgrading Systems With Leapp

G28064-01
June 2025

ORACLE®

Oracle Linux 10 Upgrading Systems With Leapp,

G28064-01

Contents

Preface

1   About Leapp

2   Preparing for the Upgrade

3   Upgrading the System

4   Completing Postupgrade Tasks

5   Troubleshooting Oracle Linux Upgrades

A Supported Repositories in Leapp Upgrades

Preface

Oracle Linux 10: Upgrading Systems With Leapp provides information about how to use the Leapp utility to perform system upgrades from Oracle Linux 9 to the current Oracle Linux 10 release.

Documentation License

The content in this document is licensed under the Creative Commons Attribution–Share Alike 4.0 (CC-BY-SA) license. In accordance with CC-BY-SA, if you distribute this content or an adaptation of it, you must provide attribution to Oracle and retain the original copyright notices.

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.

Documentation Accessibility

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

Access to Oracle Support for Accessibility

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

Diversity and Inclusion

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

mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About Leapp

The Leapp utility is a framework for updating and upgrading operating systems and applications. The utility's component packages enable the creation of different workflows into profiles for updating software.

Leapp operations consist of two phases:

- The preupgrade phase, where system checks are performed to verify if the software can be upgraded.
- The actual upgrade, which process is based on configuration files that map packages between previous and current versions of the software packages.

> ⚠️ **Caution:**
>
> The Leapp utility is used to upgrade the OSs only from the *current* Oracle Linux 9 release to the *current* Oracle Linux 10 version. The procedures in this document *don't apply to* and *are unsupported on* any other OSs or versions.

Supported Leapp Features

The Leapp utility can be used to upgrade local or remote Oracle Linux 9 systems

Upgrading Oracle Linux 9 Systems

For Oracle Linux systems, the following table lists supported and unsupported features by the Leapp utility.

Upgradeable With Leapp	Not Upgradeable With Leapp
Platforms (latest shipping updates) • x86_64 (RHCK, UEK7, and UEK8 kernels)[1] • Arm (aarch64) (UEK7 and UEK8 kernel) **Operating Systems** • Current Oracle Linux 9 version only **Profiles** • Server with GUI • Workstation • Server • Custom Operating System • Minimal Install	• Oracle applications • Oracle RDMA stack • Oracle DB products • Anything not installed by using an ISO image (Ceph, GlusterFS, OCNE, OCI image, and so on) • Migration of disks that are encrypted with LUKS • ULN integration • Upgrading with FIPS mode (not supported upstream)

[1]Latest shipping kernel versions

Requirements for Upgrading

To upgrade an Oracle Linux 9 system or instance, ensure that either one meets the following requirements:

- The minimum installation requirements as listed in System Requirements in Oracle Linux 10: Installing Oracle Linux are met.

  In particular, ensure that the system has disk space to complete the Leapp upgrade. Disk space in the `/boot` partition is especially paramount. The partition must have at least 1 GB (optimally 2 GB) of disk space with at least 320 MB of free disk space within it to accommodate the installation of the Red Hat Compatible Kernel (RHCK) and Unbreakable Enterprise Kernel (UEK), `initramfs`, `kdump` images, and so on. Examine the preupgrade report which might notify you if insufficient disk space is detected. For more information about the preupgrade phase, see Assessing the Capability of the System for Upgrading.

- Only packages provided by Oracle are installed. Upgrade stability isn't guaranteed if third-party packages are present in the system.

- Oracle Linux yum server at https://yum.oracle.com or a corresponding yum mirror is accessible.

  If accessing repositories from a mirror or a local repository, ensure that both Oracle Linux9 and Oracle Linux 10 channels are mirrored.

- x86_64 deployments are running Unbreakable Enterprise Kernel Release 7, 8, or the Red Hat Compatible Kernel (RHCK).

- aarch64 deployments are running the Unbreakable Enterprise Kernel Release 7 or 8.

Check the following references for information that might have an impact on the upgrade process:

- Oracle Linux 10: Release Notes for Oracle Linux 10
- Known Issues

Kernels Upgradeable With Leapp

The following table provides guidance about which kernel upgrades can be performed with the Leapp utility. The table assumes that the Oracle Linux 9 host satisfies the requirements listed in Requirements for Upgrading.

	Starting Kernel (Oracle Linux 9)	Ending Kernel (Oracle Linux 10)	Supported
x86_64	RHCK	RHCK	Yes[1]
	RHCK	UEK8	No
	UEK7 or UEK8	UEK8	Yes
	UEK7 or UEK8	RHCK	No
aarch64[2]	UEK7 or UEK8	UEK8	Yes

[1]Unbreakable Enterprise Kernel Release 8 remains on the system or instance after the upgrade. If preferred, the administrator can remove this kernel.

[2]RHCK isn't distributed nor available for the aarch64 platform.

2
Preparing for the Upgrade

Complete the steps as applicable to prepare for an upgrade from Oracle Linux 9 to Oracle Linux 10.

1. Set up a means to connect remotely through a console.

   This document assumes that you're performing a Leapp upgrade remotely. In this case, a console is necessary so you can monitor the progress of the upgrade process, especially as the upgrade performs automatic reboots.

   The following list shows console connection options you can use:

   - Oracle Linux server: Use Oracle Integrated Lights Out Manager (ILOM). See https://docs.oracle.com/en/servers/management/ilom/index.html.
   - Oracle Private Cloud Appliance: Use the Instance Console Connection. See https://docs.oracle.com/en/engineered-systems/private-cloud-appliance/index.html.
   - Oracle Linux Virtualization Manager or Oracle Linux Kernel based Virtual Machines (KVM): User `virt-viewer`, `virt-manager`, or Cockpit Web Console. See Oracle Linux Virtualization Manager documentation.

   > **Note:**
   >
   > If you connect to the system by using SSH or by using VNC to a VNC service running on the system, you're disconnected during the upgrade process and are unable to log in until the upgrade is completed.

2. Perform a backup.

   Always back up a system so that the system can be restored to its former state if the upgrade fails.

3. Shut down all production workloads that have been set up to run on the system, as the upgrade is intrusive and requires several reboots.

4. If the system has network mounted file systems, unmount them, and then insert related entries in the `/etc/fstab` file inside comment marks.

   See File Systems and Storage Issues.

5. If the system is behind a proxy, configure the proxy settings in `/etc/dnf/dnf.conf`, for example:

   ```
   proxy=proxy-url:port
   ```

   See Oracle Linux: Managing Software on Oracle Linux.

6. If you installed the package, clear any packages with locked versions.

7. Obtain the latest Oracle Linux 9 packages.

   ```
   sudo dnf update -y
   ```

8. If the system is registered with ULN or a ULN mirror, unregister the system.

ORACLE®

See the following documentation for this step.

- Removing a System From ULN in Oracle Linux: Managing Software on Oracle Linux
- Checking Yum Configuration in https://yum.oracle.com/getting-started.html#checking-yum-configuration.

9. Reboot the system.

```
sudo reboot
```

10. Ensure that the appstream and baseos_latest repositories are enabled.

```
dnf repolist
```

11. Install the Leapp utility using the following command:

```
sudo dnf install -y leapp-upgrade
```

3

Upgrading the System

This chapter discusses the different stages of a system upgrade, which are the assessment phase and the upgrade phase. The main commands to use for these stages are `leapp preupgrade` and `leapp upgrade`, and followed by command arguments. For a list of these arguments, use the `-h` or `--help` argument, for example:

```
sudo leapp preupgrade --help
```

Assessing the Capability of the System for Upgrading

The preupgrade phase checks whether the system is fully ready for the upgrade.

> ⊘ **Important:**
>
> Refer also to Known Issues to better prepare the system for a Leapp upgrade.

Running the Preupgrade

Through the preupgrade phase, you can check whether the system is ready for the upgrade.

Running the preupgrade phase is recommended to ensure that the system is cleared of issues that might impede the upgrade. In this phase, you generate an assessment report that identifies risks to upgrading. The report also provides recommendations for resolving those risks.

1. If you're using a proxy server, edit the `/etc/yum.repos.d/leapp-upgrade-repos-ol10.repo` by adding the proxy setting for each repository entry.

   To add the setting in a single operation, you can run the following command:

   ```
   sudo sed -i '/^enabled=0.*/a proxy=http://proxy-host:proxy-port' /etc/yum.repos.d/
   leapp-upgrade-repos-ol10.repo
   ```

2. Run the `preupgrade` command.

   Use the appropriate command argument for a system.

   • On a system:

   ```
   sudo leapp preupgrade --oraclelinux [--enablerepo repository]
   ```

   For detailed information about the arguments, see Using Command Arguments to Enable Repositories.

   This process generates a process log, a report, and a `answerfile` file.

Analyzing the Leapp Report

The `/var/log/leapp/leapp-report.txt` identifies potential risks to the upgrade. The risks are classified as high, medium, or low. A high risk that would prevent an upgrade is further

classified as an inhibitor. The report summarizes the issues behind the identified risk and also suggests remediations if any are needed.

Ensure that you complete the recommended remedies to clear risks that are labeled high and can inhibit the upgrade process.

After addressing the reported risks, run the `preupgrade` command again. In the regenerated report, verify that all serious risks are cleared.

To better illustrate the contents of the report, consider the examples in the following sections:

Legacy Network Configuration Issue

The report might warn about network configuration files using the legacy `ifcfg` format.

```
Risk Factor: high (inhibitor)
Title: Legacy network configuration found
Summary: Network configuration files in legacy "ifcfg" format are present.In Oracle
Linux 10, support for these files is no longer enabled and the configuration will be
ignored. The following files were found:
    - /etc/sysconfig/network-scripts/ifcfg-ens3
Related links:
    - nmcli(1) manual, describes "connection migrate" sub-command.: https://
networkmanager.dev/docs/api/latest/nmcli.html
    - nm-settings-ifcfg-rh(5), description of the "ifcfg" format: https://
networkmanager.dev/docs/api/latest/nm-settings-ifcfg-rh.html
    - nm-settings-keyfile(5), description of the "keyfile" format: https://
networkmanager.dev/docs/api/latest/nm-settings-keyfile.html
```

To resolve this issue, review the following instructions and suggestions:

```
Remediation: [hint] Convert the configuration into NetworkManager native "keyfile"
format.
[command] nmcli connection migrate /etc/sysconfig/network-scripts/ifcfg-ens3
```

ipa-server Package is Installed Without IdM Configured Issue

The report might warn about the free-ipa package.

```
Risk Factor: medium
Title: ipa-server package is installed but no IdM is configured
Summary: The ipa-server package is installed but neither IdM server nor client is
configured on this system.
Related links:
    - IdM migration guide: https://docs.oracle.com/en/operating-systems/oracle-linux/10/
leapp
```

To resolve this issue, review the following instructions and suggestions:

```
Remediation: [hint] Remove ipa-server package and install it after in-place upgrade, if
you need ipa-server functionality on upgraded system
```

Providing Information to the Leapp Answerfile

In addition to completing the recommendations of `/var/log/leapp/leapp-report.txt`, you must also provide answers to all the items in `/var/log/leapp/answerfile`.

An inhibitor might be reported both in `/var/log/leapp/answerfile` and `/var/log/leapp/leapp-report.txt`, with the latter file providing an alternative remedy. Despite overlapping contents, always examine both files to ensure a successful upgrade.

The `/var/log/leapp/answerfile` file consists of specific verification checks that Leapp performs on the system. A verification check contains information about the system and also prompts you for confirmation on the action to be performed. The file provides context and information to help guide you on the response required.

> **Note:**
>
> All verification checks listed in the `answerfile` must be answered. Unanswered items cause the upgrade process to halt.

To provide responses to `answerfile`, choose from one of the following methods:

- Use the `leapp answer` command.

  Run this command on the specific section that needs correcting. For example, to confirm the PAM module verification, you would type:

  ```
  sudo leapp answer --section remove_pam_pkcs11_module_check.confirm=True
  ```

- Edit the contents of `/var/log/leapp/answerfile`.

  Go to the specific section that you want to confirm, such as `[remove_pam_pkcs11_module_check]`, uncomment its `confirm =` line and specify the answer, for example:

  ```
  confirm = True
  ```

Performing the Upgrade

After you have completed the `/var/log/leapp/answerfile` and verified that `/var/log/leapp/leapp-report.txt` no longer reports risks, upgrade the system as follows:

1. Using a console, connect to the system that you're upgrading.

   - If you're upgrading a remote system configured with a VNC server, connect to the system by using a VNC client.

2. On a separate terminal window of the system or instance to be upgraded, run the `upgrade` command with the appropriate command argument.

   - On a system:

     ```
     sudo leapp upgrade --oraclelinux [--enablerepo repository]
     ```

   For detailed information about the command arguments, see Using Command Arguments to Enable Repositories.

3. Verify that the report summary returns no errors or inhibitors. For example, the following report shows no errors or inhibitors:

   ```
   ============================================================
                          REPORT OVERVIEW
   ============================================================

   HIGH and MEDIUM severity reports:
       1. Packages not signed by Oracle found on the system
       2. Default Boot Kernel
       3. Berkeley DB (libdb) has been detected on your system
   ```

```
    4. PostgreSQL (postgresql-server) has been detected on your system
    5. Manual migration of data from MySQL database might be needed

Reports summary:
    Errors:                      0
    Inhibitors:                  0
    HIGH severity reports:       1
    MEDIUM severity reports:     4
    LOW severity reports:        1
    INFO severity reports:       2

Before continuing, review the full report below for details about
discovered problems and possible remediation instructions:
    A report has been generated at /var/log/leapp/leapp-report.txt
    A report has been generated at /var/log/leapp/leapp-report.json


============================================================
                    END OF REPORT OVERVIEW
============================================================
```

If any errors or inhibitors appear, resolve them before rebooting the system and rerun the Leapp upgrade.

4. Reboot the system.

   ```
   sudo reboot
   ```

5. While the system reboots, monitor the progress on the console.

   At the completion of the boot process, the utility automatically proceeds with upgrading packages. This operation takes awhile to complete and also includes multiple automatic reboots.

   > ⚠️ **Caution:**
   >
   > Do *not* interrupt the ongoing processes at this stage. Wait until the login screen appears, which indicates that the entire upgrade process has completed. Only then can you begin to use the system.

6. When the login screen appears on the console, log in with the proper credentials.

   > ❗ **Important:**
   >
   > See Oracle Linux 10 documentation for information about new features, changes, and deprecated items in Oracle Linux 10. Thus, you can identify post upgrade tasks that you might need to complete.

Verifying the Upgrade

Upon completion, the upgrade process generates the same files as the preupgrade phase: a process log, a report, and the `/var/log/leapp/answerfile`. On a terminal, perform the following steps:

1. Examine the `/var/log/leapp/leapp-report.txt` and fulfill any important recommendations to be completed after the upgrade process.

2. Perform the following verifications:

   To verify the system's new OS version, type:

   ```
   cat /etc/oracle-release
   ```

   To check the system's kernel version, type this command to verify that the kernel contains the `el10` substring:

   ```
   uname -r
   ```

   You can also identify the system's default kernel with the following command:

   ```
   sudo grubby --default-kernel
   ```

4
Completing Postupgrade Tasks

> **⚠ Important:**
>
> The following tasks aren't comprehensive. Depending on the setup, you might need to perform other procedures to return the newly upgraded system back into operation. Review the `/var/log/leapp/leapp-report.txt` that's generated after the upgrade. This report might contain more recommendations to ensure that the upgraded system remains in a supported state.

1. Enable the firewall.

   ```
   sudo systemctl start firewalld
   sudo systemctl enable firewalld
   ```

2. Check that the network connections are operational, for example, by pinging the system and see if connectivity is obtained with the system.

3. If you had `dnf` customizations before the upgrade, restore them in the upgraded system's `/etc/dnf/dnf.conf` file, for example:

   ```
   proxy=proxy-url:port
   ```

4. Restore network mounted file systems that you unmounted before the upgrade. See File Systems and Storage Issues.

5. Set SELinux to run in `Enforcing` mode.

   During the upgrade, the Leapp utility sets SELinux to run in `Permissive` mode. To restore the setting: To revert to `Enforcing` mode and verify the setting, type:

   ```
   sudo setenforce enforcing
   ```

   You can verify the mode of SELinux as follows:

   ```
   getenforce
   ```

   ```
   Enforcing
   ```

   To make this setting persist across system reboots, add the following line to `/etc/selinux/config`:

   ```
   SELINUX=enforcing
   ```

   Then run the following command:

   ```
   sudo grubby --update-kernel=ALL --remove-args="enforcing=0"
   ```

6. Reevaluate then reapply the security policies such as setting cryptographic policies.

7. Inspect the system for unneeded configurations and files.

> **Note:**
>
> Some of these unneeded files might be reported in the generated `/var/log/leapp/leapp-report.txt` after the upgrade. Ensure that you review this report and complete its post upgrade recommendations.

This step aims to ensure that the configurations are consistent with the new OS version. The completion of this step would vary, depending on what you deem is important to retain from the previous system's state. Consider the following guidelines:

- Remove kernels and kernel modules that are no longer applicable.

- If you remove kernels, you might also need to update the GRUB menu so that the menu options only reflect the actual kernels on the system.

- Review `/etc/yum.repos.d` for entries that might need to be addressed, such as customized repositories.

  For example, during system updates, `*.rpmnew` files might be created to prevent overwriting corresponding existing `*.rpm` files. You would need to use the contents of the `*.rpmnew` files to guide you when changing the corresponding `*.rpm` files.

- Remove residual packages from the previous Oracle Linux version.

  a. Edit `/etc/dnf/dnf.conf` by removing or commenting out `exclude=` lines that refer to `leapp` packages.

  b. Use commands such as `rpm -qa` to list packages that can be removed.

  ```
  rpm -qa | grep el9
  rpm -qa | grep leapp
  ```

  c. Use the `sudo dnf remove` command to remove the packages listed by the queries.

> **⚠ Caution:**
>
> Residual `el9` packages that remain on the system do not receive updates. Vulnerability scanners or other security audits might report warnings or failures about these packages.

8. Remove the `/root/tmp_leapp_py3` directory, which is no longer needed.

9. If you removed the system from ULN to perform the upgrade, register the system again and configure the appropriate channels.

   For more information, see Registering an Oracle Linux System With ULN and ULN Channel Subscription Management in Oracle Linux: Managing Software on Oracle Linux.

5
Troubleshooting Oracle Linux Upgrades

This chapter provides troubleshooting information and describes known issues that might affect the upgrade process.

Tools for Troubleshooting

Use the following options to generate more output when you are generating the preupgrade report or performing the actual upgrade:

- `--verbose` displays warnings, error messages, and other critical information.
- `--debug` adds debug information in addition to the same output as the `--verbose` option.

You can use the following resources and tools for obtaining troubleshooting information:

- `/var/log/leapp/leapp-report.txt`
- `/var/log/leapp/leapp-upgrade.log`
- `/var/log/leapp/dnf-debugdata/` a directory for debug information. Note that this directory is created only if you use the `--debug` option when issuing either the `preupgrade` or the `upgrade` command.
- `journalctl` command

Known Issues

The following are known issues that you might encounter when upgrading an Oracle Linux 9 system to Oracle Linux 10.

Upgrade Issues

- **Leapp might report missing packages that are marked for installation**

  The `/var/log/leapp/leapp-preupgrade.log` or `/var/log/leapp/leapp-upgrade.log` files might report a warning similar to the following:

  ```
  Warning: Packages marked by Leapp for install not found in repositories
  metadata: rpcgen python3-pyxattr libnsl2-devel rpcsvc-proto-devel
  ```

  These packages are in the `Oracle Linux10 Codeready Builder` repository, which is a developer repository and is disabled by default.

  If the system requires these packages, then during the preupgrade or the upgrade phase, add the `--enablerepo ol10_codeready_builder` option to the appropriate Leapp command, for example:

  ```
  sudo leapp upgrade --oraclelinux --enablerepo ol10_codeready_builder
  ```

  Repositories that have been enabled during the Leapp upgrade remain enabled on the Oracle Linux 10 system after the upgrade completes.

Alternatively, after completing the upgrade, you can manually install the packages required for your installation by using the `dnf` command.

Bug ID 32827043

- **Some `el9` packages might not be upgraded**

  The same `rpm -qa` command syntax in the previous item that detects MySQL-related `*.el9` packages might also list more `*el9` packages on the system that weren't upgraded. Packages might not be upgraded if they were installed from repositories that aren't supported by Leapp, such as developer repositories. For such packages, do the following:

  1. Go to https://yum.oracle.com and check the Oracle Linux 10 repositories that would serve the packages you need.

  2. After the upgrade is completed, manually install the packages from those Oracle Linux 10 repositories.

  3. After all the necessary packages have been installed, remove the residual `el9` packages from the system.

  Bug ID 32878386

- **(aarch64) Upgrade log might report errors related to the vmd module**

  After completing an upgrade on aarch64 systems, the Leapp upgrade log might report the following message:

  ```
  dracut-install: Failed to find module 'vmd'
  ```

  The VMD module doesn't apply to the Arm architecture and therefore, the error message can be safely ignored.

  Bug ID 34172552

- **The `signature is not alive` error appears during reboot phase of leapp upgrade**
  If the upgrade fails during a reboot phase with errors similar to `Signature ... created at ... invalid: signature is not alive`, verify that the system time is synchronized or try to upgrade at a later time. The in-place upgrade process might fail if the local time on the system is in the past compared to the build date of the latest release in the public RPM package.

  (Bug 38083008)

File Systems and Storage Issues

- **Systems with Btrfs in a RAID configuration can't be upgraded**

  A system that uses the Btrfs file system in a RAID configuration can't be upgraded. In the `/var/log/leapp/leapp-report.txt` that's generated by the `preupgrade` command, this configuration is flagged as an inhibitor and no remedy is provided. If you upgrade the system and that configuration is detected, the upgrade process halts.

- **Detected XFS filesystems without bigtime feature.**
  The XFS v5 file system format introduced the `bigtime` feature in Oracle Linux 9, to provides timestamps beyond the year 2038. XFS filesystems that don't have the `bigtime` feature enabled remain vulnerable to timestamp overflow issues. It is recommended to

enable this feature on all XFS filesystems to ensure long-term compatibility and prevent potential failures. Following XFS file systems have not enabled the `bigtime` feature:

```
- /kvm
- /boot
- /
```

To enable the `bigtime` feature on XFS v5 filesystems, use the following command:

```
xfs_admin -O bigtime=1 <filesystem_device>
```

For older XFS v5 filesystems this step can only be done offline (for example, without the filesystem mounted).

- **Hosts with network mounted file systems can't be upgraded**

  Leapp doesn't support upgrading systems with mounted file systems on network storage, NFS, or iSCSI. As a workaround, unmount the file systems and comment out their entries from `/etc/fstab`. After the upgrade is completed, you can restore the entries and remount the file systems.

Networking Issues

- **Possible upgrade error if system has several NICs with the same prefix as NIC that's used by kernel**

  The in-place upgrade process might cause an error if the system to be upgraded has more than one NIC that shares the same prefix as the NIC that's used by the kernel, for example `eth`. After the upgrade, the system's network connectivity is lost.

  For more information, see About Network Interface Names in Oracle Linux 10: Setting Up Networking With NetworkManager.

- **NetworkManager might not start after the upgrade completes**

  After the upgrade, the system's `NetworkManager` might not start because of the failure of its name resolution service. The failure can be verified by checking the status of the service.

  ```
  systemctl status systemd-resolved.service
  ```

  ```
  ● systemd-resolved.service - Network Name Resolution
     Loaded: loaded (/usr/lib/systemd/system/systemd-resolved.service;
  disabled; >
     Active: inactive (dead)
       Docs: man:systemd-resolved.service(8)
             https://www.freedesktop.org/wiki/Software/systemd/resolved
  ```

  The `/var/log/messages` file also reports the following error:

  ```
  dbus-daemon[742]: [system] Activation via systemd failed for unit
  'dbus-org.freedesktop.resolve1.service': Unit
  dbus-org.freedesktop.resolve1.service not found.
  ```

  To resolve this issue, choose one of the following workarounds:

  - Configure `NetworkManager` to not use `systemd-resolved.service`.

    Add the following entries to the `/etc/NetworkManager/conf.d/no-systemd-resolved.conf` file:

```
[main]
systemd-resolved=false
```

– Enable the `systemd-resolved.service` as follows:

```
systemctl enable systemd-resolved.service

Created symlink /etc/systemd/system/dbus-org.freedesktop.resolve1.service →
/usr/lib/systemd/system/systemd-resolved.service.
Created symlink
/etc/systemd/system/multi-user.target.wants/systemd-resolved.service →
/usr/lib/systemd/system/systemd-resolved.service.

systemctl start systemd-resolved.service
```

You can also adopt other methods that are more consistent with the network name resolution model that you're using for the specific setup. For useful information, see About Network Interface Names in Oracle Linux 10: Setting Up Networking With NetworkManager.

Hardware Related Issues

- **Systems with unrecognized hardware can't be upgraded**

  Support for certain hardware, such as the `e1000` driver, has been removed from . The upgrade can't proceed on platforms that have such hardware installed. Even though UEK might continue to support the hardware, the upgrade procedure is still inhibited if the hardware is detected on the system.

Leapp Overlay Size Issues

- **Upgrading might require increased overlay size**

  Upgrading Oracle Linux 9 systems with a huge number of packages to Oracle Linux 10 might fail because of insufficient space in the Leapp overlay file systems that are used during the upgrade. You might see the following error message:

```
Error: Transaction test error:
  installing package package-name needs 4MB on the / filesystem
```

  As a workaround, increase the `LEAPP_OVL_SIZE` variable. The default size is 4096. The actual size you would need might be larger depending on the specific setup. Use the following command:

```
sudo export LEAPP_OVL_SIZE=new-size
```

A

Supported Repositories in Leapp Upgrades

This appendix shows repositories that are used in a system or instance upgrade that uses the Leapp utility.

Repository Mappings

The following table shows repository correspondences between Oracle Linux 9 and Oracle Linux 10. The table helps you to identify the corresponding repositories that the Leapp utility makes available after the host has completed the upgrade.

This table shows the repository mappings for the Oracle Linux KVM Stack.

Oracle Linux 9 DNF Repositories	Oracle Linux 10 DNF Repositories	Notes
ol9_baseos_latest	ol10_baseos_latest	All Oracle Linux 10 upgrades require the BaseOS and AppStream repositories.
ol9_appstream	ol10_appstream	
ol9_UEKR7 ol9_UEKR8	ol10_UEKR8	Oracle Linux 10 requires UEK R8 as a minimum UEK version.
ol9_addons	ol10_addons	
ol9_codeready_builder	ol10_codeready_builder	Suggested for developer systems only.

Using Command Arguments to Enable Repositories

As more products are upgradeable with future versions of the Leapp utility, the number of repositories that need to be enabled after the upgrade might also increase. The Leapp upgrade commands would become complicatedly long as you manually list the repositories to be enabled in the command syntax.

Oracle has provided the following convenience switches or arguments that can be used with the Leapp `preupgrade` or `upgrade` commands. When used, these arguments automatically apply the `--enablerepo` subcommand to repositories that are appropriate to the host that you're upgrading.

`--enablerepo`
Use the option to enable required repositories. You must use the option for every repository you want to enable, for example:

```
sudo leapp preupgrade --enablerepo 'ol10_addons' --enablerepo
'ol10_codeready_builder' ...
```

`--oraclelinux`

This argument is used on system upgrades that you perform either locally or remotely. The argument detects the system's architecture and automatically uses the repositories that are applicable to the architecture.

When you use this argument, the following repositories are automatically enabled:

- `ol10_baseos_latest`

- `ol10_appstream`

- `ol10_UEKR8`

Using this option is equal to using `--enablerepo` individually for each repository listed.

You can use `--enablerepo` to add any other required repositories not already included in `--oraclelinux`. For example:

```
sudo leapp preupgrade --oraclelinux --enablerepo 'ol10_addons' --enablerepo
'ol10_codeready_builder' ...
```

This command is equal to :

```
sudo leapp preupgrade --enablerepo 'ol10_baseos_latest' --enablerepo 'ol10_appstream' --
enablerepo 'ol10_UEKR8' --enablerepo 'ol10_addons' --enablerepo 'ol10_codeready_builder'
```

`--iso`

Specify an Oracle Linux installation image to use to perform an in place upgrade. You must specify the full path to the ISO image:

```
sudo leapp upgrade --iso <path-to-ISO>
```

The ISO image must be stored on the local partition, not on removable media or the local network.

> **Note:**
>
> Depending on the package composition on some custom Oracle Linux deployments, in-place Leapp upgrades using the `--iso` option might not include some packages. This is an expected and known limitation of `--iso` option. Use the `--iso` option only for isolated environments, where access to local repository mirrors or public Oracle repositories is impossible. Analyze the following files before proceeding with the upgrade to confirm what packages are to be installed, removed, or updated:
>
> - /var/log/leapp/leapp-preupgrade.log
>
> - /var/log/leapp/leapp-upgrade.log
>
> - /var/log/leapp/leapp-report.txt