Oracle Linux 9

Upgrading Systems With Leapp

F57075-32
February 2026

Oracle Linux 9 Upgrading Systems With Leapp,

F57075-32

Contents

A Supported Repositories in Leapp Upgrades

Preface

Oracle Linux 9: Upgrading Systems With Leapp provides information about how to use the Leapp utility to perform system upgrades from Oracle Linux 8 to the current Oracle Linux 9 release.

1

About Leapp

> ⚠ **Caution**
>
> Oracle Linux 7 is now in Extended Support. See Oracle Linux Extended Support and Oracle Open Source Support Policies for more information. Migrate applications and data to Oracle Linux 8 as soon as possible.

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 8 release to the *current* Oracle Linux 9 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 8 systems and instances on Oracle Cloud Infrastructure that are based on the Oracle Linux 8 image.

Upgrading Oracle Linux 8 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, UEKR6 , and UEK7 kernels)[1] • Arm (aarch64) (UEK kernel)[2] **Operating Systems** • Current Oracle Linux 8 version only **Profiles** • Server with GUI • Workstation • Server • Custom Operating System • Minimal Install • Virtualization Host **Supported Stacks** • Oracle KVM Stack	• 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

[1]Latest shipping kernel versions

[2]Limits exist on auto upgrading for Arm with UEK because the kernel page size changes from UEKR6 to UEKR7.

Upgrading Oracle Linux 8 Oracle Cloud Infrastructure Instances

The Leapp utility can also upgrade both the x86_64 and Arm (aarch64) platforms that are running Oracle Linux 8 instances on Oracle Cloud Infrastructure.

The following table lists available and unavailable features:

Supported with Leapp	Unsupported with Leapp
Images • Oracle Linux 8 See https://docs.oracle.com/iaas/Content/Compute/References/images.htm	**Images** • Oracle Autonomous Linux 8 • • Bring Your Own (BYOI) Images See https://docs.oracle.com/iaas/Content/Compute/References/bringyourownimage.htm. • Oracle Cloud Infrastructure Marketplace images
Shapes • All Flexible Shapes See https://docs.oracle.com/iaas/Content/Compute/References/computeshapes.htm#flexible. • All Virtual Machine Shapes See https://docs.oracle.com/iaas/Content/Compute/References/computeshapes.htm#vmshapes.	**Shapes** • Bare Metal Shapes See, https://docs.oracle.com/iaas/Content/Compute/References/computeshapes.htm#baremetalshapes..
Features • Instances managed with the Oracle OS Management Hub service See https://docs.oracle.com/en-us/iaas/osmh/doc/overview.htm.	**Features** • Any unupgradeable features listed in the table in Upgrading Oracle Linux 8 Systems.

Upgrading Oracle Linux 8 KVM Hosts

The Leapp utility can be used to upgrade Oracle Linux 8 systems that host KVM virtual machines. Systems must fulfill the other Leapp criteria listed in the previous sections. The following table lists the scope of KVM host support.

> **Note**
>
> The Oracle Linux KVM Image isn't an Oracle Cloud Infrastructure platform image and not supported by Leapp.

Supported with Leapp	Unsupported with Leapp
• Upgrading the Oracle Linux 8 Latest KVM packages to the Oracle Linux 9 KVM AppStream • Upgrading the Oracle Linux 8 KVM Utilities to the Oracle Linux 9 KVM AppStream	• Switching between default KVM stack and Oracle KVM stacks • KVM packages or Appstream packages from developer repositories • Packages or features not included with the shipping product • Upgrading KVM hosts while KVM virtual machines (guests) are running.

For repository mappings between preupgrade stage and postupgrade stage that involve KVM clients, see Supported Repositories in Leapp Upgrades.

Requirements for Upgrading

To upgrade an Oracle Linux 8 system or instance, ensure that either one meets the following requirements:

- The minimum installation requirements as listed in System Requirements in Oracle Linux 9: 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 250 MB of disk space 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 Linux 8 and Oracle Linux 9 channels are mirrored.
- x86_64 deployments are running at least Unbreakable Enterprise Kernel Release 6 or the Red Hat Compatible Kernel (RHCK).
- aarch64 deployments are running at least the Unbreakable Enterprise Kernel Release 6.

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

- Oracle Linux 9: Release Notes for Oracle Linux 9.4
- 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 8 host satisfies the requirements listed in Requirements for Upgrading.

	Starting Kernel (Oracle Linux 8)	Ending Kernel (Oracle Linux 9)	Supported
x86_64 not using Btrfs file system	RHCK	RHCK	Yes[1]
	RHCK	UEK	No
	UEK	UEK	Yes
	UEK	RHCK	No
x86_64 using Btrfs file system [2]	UEK	UEK	Yes
	UEK	RHCK	No
aarch64[3]	UEK R7	UEK[4]	Yes
aarch64 not using Btrfs file system	UEK R6	UEK[4]	Yes [5]
aarch64 using Btrfs file system	UEK R6	UEK[4]	No[5]

[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 in Oracle Linux 8 doesn't support the Btrfs file system.

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

[4]For aarch64 systems, Oracle Linux 9 ships with the latest UEK release.

[5]The Arm page size changes from UEK R6 to UEK R7. For details, see Btrfs File System Issue.

2 Preparing for the Upgrade

> **⚠ Caution**
>
> The SHA-1 hash algorithm is deprecated in Oracle Linux 9 and not used by default. This security policy affects RSA/SHA-1 signatures which can no longer be verified against the default cryptographic policy. Leapp upgrade is inhibited if packages are found with a RSA/SHA-1 digital signature. Before upgrading, consider the following options:
>
> - Contact the package vendor and ask for new builds that are signed with valid signatures, and then install these.
> - Alternatively, remove the incompatible packages.
>
> For more information about Oracle Linux security, see General Oracle Linux documentation.

Complete the steps as applicable to prepare for an upgrade from Oracle Linux 8 to Oracle Linux 9. Unless specified otherwise, all the procedures for upgrading an Oracle Linux 8 system also apply upgrading an Oracle Linux 8 instance on Oracle Cloud Infrastructure.

1. Review *Completing Postupgrade Tasks* in Oracle Linux 8: Upgrading Systems With Leapp and ensure that the Oracle Linux 8 system has been configured to be in a supported state.

   > **Important**
   >
   > You **MUST** perform this step. In particular, remove older packages to avoid errors and to ensure a successful upgrade. This step is critical especially if the system has been upgraded earlier from an Oracle Linux 7 installation where residual packages can be inhibitors to an upgrade.

2. If the Oracle Linux 8 system was upgraded earlier from Oracle Linux 7 and then retained the use of legacy network scripts, you must migrate these scripts, as they're no longer usable in Oracle Linux 9. The presence of these scripts inhibits upgrading Oracle Linux 8 to Oracle Linux 9. For example, you can run the following command:

   ```
   sudo nmcli connection migrate
   ```

   For more information about configuring the network by using `NetworkManager`, see Oracle Linux 8: Setting Up Networking.

3. Ensure that SSH root login is disabled in the `/etc/ssh/sshd_config` file by verifying that the comment mark (#) is at the beginning of the following line, as shown:

   ```
   #PermitRootLogin yes
   ```

   Alternatively, replace the `yes` value with `no`.

4. 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 Cloud Infrastructure instance: Create a console connection by following the instructions at https://docs.oracle.com/iaas/Content/Compute/References/serialconsole.htm#Instance_Console_Connections.
- 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.

5. If you are upgrading an Oracle Linux instance on Oracle Cloud Infrastructure, ensure that Oracle OS Management Hub and Oracle Autonomous Linux are disabled.
6. If you are upgrading an Oracle Linux instance on Oracle Cloud Infrastructure, verify if Oracle OS Management Hub is running on the instance. Do the following:
   a. From Oracle Cloud Infrastructure, open the navigation menu and click **Compute**. Under Compute, click **Instances**.
   b. Select the instance you want to upgrade.
   c. Click the **OS Management** tab.
      - If the Oracle OS Management description specifies "OS Management Hub is not enabled for this instance," then the instance isn't managed by the OS Management Hub.
      - If the description provides information about the instance, then it is an OS managed instance. You can use Leapp with OS Management Hub to upgrade the instance.

      An instance might be registered with OS Management Hub but with a stopped `osmh-agent`. In this scenario, do one of the following:

      - If you want the instance to be upgraded with OS Management Hub, enable the appropriate agent.
      - If you don't want to upgrade the instance with OS Management Hub, then unregister the instance. For more information, see https://docs.oracle.com/en-us/iaas/osmh/doc/home.htm.
7. If the instance is managed by OS Management Hub, ensure that **Oracle Linux 8 Application Stream x86_64** or **Oracle Linux 8 Application Stream aarch64** Software Source is attached to the instance. For more information about managing data sources using OS Management Hub, see https://docs.oracle.com/en-us/iaas/osmh/doc/software-sources.htm

8. Perform a backup.

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

   > **Note**
   >
   > For an Oracle Linux 8 instance in Oracle Cloud Infrastructure, perform a boot volume backup. For instructions, see https://docs.oracle.com/iaas/Content/Block/Concepts/bootvolumebackups.htm.

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

10. Disable `AllowZoneDrifting` in the firewall configuration file to prevent the Leapp upgrade from being blocked. Type:

    ```
    sudo sed -i "s/^AllowZoneDrifting=.*/AllowZoneDrifting=no/" /etc/firewalld/
    firewalld.conf
    ```

11. 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.

12. 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.

13. If you installed the `python3-dnf-plugin-versionlock` package, clear any packages with locked versions.

    ```
    sudo dnf versionlock clear
    ```

14. Verify that the instance isn't running on a debug kernel. The following command must show an Oracle Linux RHCK or UEK kernel without the word, "debug" in it:

    ```
    uname -r
    ```

    If the system is running on a debug kernel, reboot the system using a non debug kernel.

15. Run the following command to ensure that no debug versions of Oracle Linux RHCK or UEK kernels are installed:

    ```
    dnf erase "kernel-debug*" "kernel-uek-debug*"
    ```

16. Obtain the latest Oracle Linux 8 packages.

    ```
    sudo dnf update -y
    ```

17. If you're upgrading Oracle Linux 8 KVM hosts, stop all the virtual machines that might be running.

    The command lists the virtual machines. From the list, stop specific virtual machines that are running.

    a. List the available virtual machines.

       ```
       sudo virsh list --all
       ```

    b. From the list, stop individual virtual machines that are running.

```
sudo virsh shutdown vm-name
```

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

    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.

19. Reboot the system.

    ```
    sudo reboot
    ```

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

21. Install the Leapp utility using the following command:

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

3 Upgrading the System

This chapter discusses the 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
```

Unless specified otherwise, all the procedures for upgrading an Oracle Linux 8 system also apply to upgrading an Oracle Linux 8 instance on Oracle Cloud Infrastructure.

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, 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-ol9.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-ol9.repo
   ```

2. Run the `preupgrade` command.

   Use the appropriate command argument for a system or an Oracle Cloud Infrastructure instance.

   - On a system:

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

   - On an instance in Oracle Cloud Infrastructure:

     ```
     sudo leapp preupgrade --oci [--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:

GPG Key Issue

The report might warn about the `gpg-pubkey`.

```
Risk Factor: high
Title: Packages not signed by Oracle found on the system
Summary: The following packages have not been signed by Oracle and may be
removed during the upgrade process in case Oracle-signed packages to be removed
during the upgrade depend on them:
- gpg-pubkey
```

To resolve this issue, run the following command:

```
sudo rpm -qa | grep gpg-pubkey
```

If the command output lists only the Oracle Linux 8 public key `gpg-pubkey-ec551f03-53619141`, the issue can be ignored. Otherwise, any other unsigned packages or `gpg-pubkey` entries in the report must be manually analyzed, as they might be removed during the upgrade.

Btrfs File System Issue

On aarch64 systems, the Leapp report might report the following:

```
Title: UEKR6 has been found and BTRFS filesystem is in use.
Summary: Upgrade process was interrupted because btrfs is enabled
         and UEKR6 has been found.
```

The page size for aarch64 systems has changed from 64 KB to 4 KB in UEK R7. For more information about issues that involve this feature, see the list of Arm Features in Oracle Linux 9 in Oracle Linux 9: Release Notes for Oracle Linux 9.

If the aarch64 system that's running UEK R6 is configured with the Btrfs file system, then you can't use Leapp to upgrade to Oracle Linux 9. For more information about issues that involve upgrading aarch64 systems that use the Btrfs file system, see Unbreakable Enterprise Kernel Release 7: Release Notes (5.15.0-0.30).

If your aarch64 system is running UEK R6 but does **not** use the Btrfs file system, then, for the upgrade to proceed, confirm, and accept the page size change that takes effect because of the upgrade. You can confirm the page size change in the answer file or by running the following command:

```
leapp answer --section confirm_UEKR8_install_pagesize_4k.confirm=True
```

However, if the Oracle Linux 8 aarch64 system is already running UEK R7, then the upgrade to Oracle Linux 9 proceeds normally. No confirmation of the page change is required.

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 or the Oracle Cloud Infrastructure instance 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.
   - If you're working on an Oracle Cloud Infrastructure instance, connect to the instance through the console connection you previously created in Preparing for the Upgrade. For instructions, see Connecting to the Serial Console in https://docs.oracle.com/iaas/Content/Compute/References/serialconsole.htm#Instance_Console_Connections .

For example, on a local terminal window, the command that's provided to connect to the instance might resemble the following syntax:

```
ssh -o ProxyCommand='ssh additional-commands
```

If the command doesn't work at first use, you might need to specify the `-i path-to-key` option, for example:

```
ssh -i path-to-key -o ProxyCommand='ssh -i path-to-key additional-commands
```

Because OCI requests only rsa keys, on some systems, you might need to add the following in the `/etc/ssh/ssh_config` directory:

```
HostkeyAlgorithms +ssh-rsa
PubkeyAcceptedAlgorithms +ssh-rsa
```

2. On a separate terminal window of the system or instance to be upgraded, run the `upgrade` command with the appropriate command argument, depending on whether you're upgrading a system or an Oracle Cloud Infrastructure instance.
   - On a system:

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

   - On an instance in Oracle Cloud Infrastructure:

     ```
     sudo leapp upgrade --oci [--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:

```
Debug output written to /var/log/leapp/leapp-upgrade.log

============================================
             REPORT OVERVIEW
============================================
HIGH and MEDIUM severity reports:
    1. Packages not signed by Oracle found on the system
    2. Difference in Python versions and support in OL 8
    3. Default Boot Kernel
    4. Module pam_pkcs11 will be removed from PAM configuration
    5. Oracle Autonomous Linux managed instance upgrade requires user to
accept specific package update requirements for Oracle Autonomous Linux
    6. Oracle Autonomous Linux managed instance upgrade requires user to
accept certain software sources requirements.

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

Before continuing consult the full report:
    A report has been generated at /var/log/leapp/leapp-report.json
    A report has been generated at /var/log/leapp/leapp-report.txt
```

```
==========================================
           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.

After the completion of an instance upgrade, the instance retains its Oracle Linux 8 base image on the Instance Details page of the Oracle Cloud Infrastructure console, for example, `Oracle-Linux-8.6-2022.05-27-0`. You can apply a custom tag so you can track the upgrades that have been performed on the instance after its creation.

> **Important**
>
> See Oracle Linux 9 documentation for information about new features, changes, and deprecated items in Oracle Linux 9. 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 `el9` 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 have an instance managed by Oracle OS Management Hub, attach required software sources for Oracle Linux 9. For more information about managing software sources, see https://docs.oracle.com/en-us/iaas/osmh/doc/software-sources.htm.

   > **Note**
   >
   > The Oracle OS Management Hub can take some time to become aware of the changes to the managed instance Oracle Linux version.
   >
   > Also, before you perform this step, repositories that Leapp used to perform the upgrade remain enabled including Oracle Linux 8 and Oracle Linux 9 repositories. After performing this step, only those repositories you configure using OS Management Hub are enabled.

4. If you have an instance managed by Oracle Autonomous Linux, attach all mandatory software sources for Oracle Linux . For more information about managing software sources, see https://docs.oracle.com/en-us/iaas/autonomous-linux/doc/software-sources.htm.

   > **Note**
   >
   > The Oracle Autonomous Linux service can take some time to become aware of the changes to the managed instance Oracle Linux version.
   >
   > Also, before you perform this step, repositories that Leapp used to perform the upgrade remain enabled including Oracle Linux and Oracle Linux . After performing this step, only those repositories you configure using Oracle Autonomous Linux are enabled.

5. 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
   ```

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

7. If upgrading KVM hosts, restart the KVM virtual machines.

   ```
   sudo virsh start vm-name
   ```

8. 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"
   ```

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

10. 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. For example, if the system uses the Btrfs file system, then you can only use the UEK kernel. Therefore, consider removing the RHCK kernel and any earlier versions of the UEK kernel. Also, you can also rebuild the rescue kernel.
    - 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 el8
     rpm -qa | grep leapp
     ```

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

> ⚠ **Caution**
>
> Residual `el8` packages that remain on the system do not receive updates. Vulnerability scanners or other security audits might report warnings or failures about these packages.

11. Remove the `/root/tmp_leapp_py3` directory, which is no longer needed.
12. 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 8 system to Oracle Linux 9.

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 Linux9 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 ol9_codeready_builder` option to the appropriate Leapp command, for example:

  ```
  sudo leapp upgrade --oraclelinux --enablerepo ol9_codeready_builder
  ```

  Repositories that have been enabled during the Leapp upgrade remain enabled on the Oracle Linux 9 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 `el8` packages might not be upgraded**

  The same `rpm -qa` command syntax in the previous item that detects MySQL-related `*.el8` packages might also list more `*el8` 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 9 repositories that would serve the packages you need.
  2. After the upgrade is completed, manually install the packages from those Oracle Linux 9 repositories.
  3. After all the necessary packages have been installed, remove the residual `el8` 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

- **Errors reported that reference legacy directory /var/run**

  During the upgrade, messages similar to the following that reference the `/var/run` directory are reported:

  ```
    Installing       :
  leapp-upgrade-el8toel9-0.16.0-6.0.3.el8_6.20220810014405.8fa95c0.noarch
                                                                 6/6
    Running scriptlet:
  leapp-upgrade-el8toel9-0.16.0-6.0.3.el8_6.20220810014405.8fa95c0.noarch
                                                                 6/6
  [/usr/lib/tmpfiles.d/dnssec-trigger.conf:1] Line references path below legacy
  directory /var/run/, updating /var/run/dnssec-trigger → /run/dnssec-trigger;
  please update the tmpfiles.d/ drop-in file accordingly.
  .
  .
  .
  ```

  These messages can be ignored. The upgrade or package installation completes successfully.

  As an alternative workaround, you can update the configuration by following the instructions in the message and change the legacy `/var/run/*` directory path to `/run/*`.

  Bug ID 34491952

- **Unsigned package generates warning when upgrading from Oracle Linux 8 to Oracle Linux 9**

  This issue applies to Oracle Linux 8 systems that were upgraded from Oracle Linux 7 and are later upgraded to Oracle Linux 9.

When you use Leapp to upgrade an Oracle Linux 7 system to Oracle Linux 8, an unsigned package kernel-workaround-0.1-1.el8.src.rpm is installed as part of the upgrade process. The Leapp upgrade successfully completes, but doesn't notify you of the package.

If you later run Leapp to upgrade this same system to Oracle Linux 9, this time the upgrade process detects the existence of the package and generates a warning in the `/var/log/leapp/leapp-report.txt` as follows:

```
----------------------------------------
Risk Factor: high
Title: Packages not signed by Oracle found on the system
Summary: The following packages have not been signed by Oracle and may be
removed during the upgrade process in case Oracle-signed packages to be
removed during the upgrade depend on them:
- gpg-pubkey
- kernel-workaround
Key: f5a5d58476a97bf0a8904d00df5d1321189849ad
----------------------------------------
```

You can disregard the message. The package doesn't prevent the upgrade from completing.

(Bug ID 35343479)

System Management Issues

- **Ksplice Uptrack software displays error messages**

  During the upgrade, the Oracle Ksplice Uptrack software might report errors similar to the following:

  ```
  [ 256.033527] upgrade[390]: Upgrading : uptrack-1.2.80-0.el9.noarch 577/1453
  [ 256.037151] upgrade[390]: Running scriptlet: uptrack-1.2.80-0.el9.noarch 577/1453
  ...
  [ 256.045914] upgrade[390]: Traceback (most recent call last):
  [ 256.049230] upgrade[390]: File "/usr/lib/uptrack/access-key-from-uln", line 9, in
  <module>
  [ 256.051376] upgrade[390]: from up2date_client import up2dateAuth, rpcServer
  [ 256.056490] upgrade[390]: File "/usr/share/rhn/up2date_client/up2dateAuth.py",
  line 74
  [ 256.059251] upgrade[390]: os.chmod(path, 0600)
  [ 256.060997] upgrade[390]:
  [ 256.062842] upgrade[390]: SyntaxError: invalid token
  ```

  The report is a known but harmless issue, which you can ignore. After the upgrade is completed, Ksplice continues to operate normally.

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).

- **Upgrade blocked if `winbind` and `wins` Samba modules are used**

  If `winbind` and `wins` Samba modules are used in the `/etc/nsswitch.conf`, the upgrade is blocked. As a workaround, remove these modules from the file first, then perform the upgrade. After the upgrade is complete, restore these module entries to the file.

  For more information about configuring these modules, see Oracle Linux 9: Managing Shared File Systems.

- **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 9: Setting Up Networking.

- **After an upgrade, the firewall can close ports that had been opened**

  Because change of the firewall backend to `nftables`, some firewall ports can be closed after an upgrade.

  To resolve this problem, after the upgrade do the following:

  ```
  sudo firewall-cmd --permanent --direct --remove-passthrough ipv4 -A INPUT -
  j REJECT --reject-with icmp-host-prohibited
  sudo firewall-cmd --reload
  ```

Virtualization and Containers Issues

- **KVM virtual machine snapshots might not be listed after an upgrade**

  After an upgrade, the `libvirtd` service might report snapshot-related error messages similar to the following:

  ```
  libvirtd[53328]: internal error: Failed to parse snapshot XML from file
  '/var/lib/libvirt/qemu/snapshot/path-to-previous-snapshot-file'
  ```

  Furthermore, listing available snapshots from before the upgrade generates an empty list.

  ```
  sudo virsh snapshot-list previous-snapshot-file

   Name   Creation Time   State
  ------------------------------
  ```

  As a workaround, reboot the system. At the end of the boot process, the snapshots are listed and available again.

- **libvirtd service might fail to restart in nested virtualization configurations**

  In nested virtualization setups, the `libvrtd` service might not restart in the nested KVM host after the upgrade.

  As a workaround, reboot the nested KVM host.

- **Package installation failures reported when upgrading an Oracle Linux 8 KVM host**
  Installation of userspace packages fail when upgrading a KVM host. This error can also be detected during the preupgrade phase. The Leapp utility would report the issue with messages similar to the following example:

  ```
  Risk Factor: high
  Title: Unable to install RHEL 9 userspace packages
  Summary:
  ...
  Fatal glibc error: CPU does not support x86-64-v2
  ...
  ```

  The error occurs because of a missing `cpu mode` declaration in the virtual machine's XML file. To work around this issue, follow these steps:

  1. Run the following command:

     ```
     sudo virsh edit vm-name
     ```

  2. Add the following declaration to the virtual machine's XML file:

     ```
     <cpu mode='host-model' check='partial'/>
     ```

  3. Rerun the preupgrade process.

  > **Note**
  >
  > This issue is related to the issue `KVM virtual machines panic when started on Oracle Linux 9 hosts` that's documented in Oracle Linux 9: Release Notes for Oracle Linux 9.

Hardware Related Issues

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

  Support for certain hardware, such as the `e1000` driver, has been removed from RHCK beginning from RHCK 8. 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 8 systems with a huge number of packages to Oracle Linux 9 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
  ```

> **⚠ Caution**
>
> The new size that you set for this variable must not exceed 25 percent of the available space on the root partition.

Low Open Files Limit Issues

- **Upgrading might require increased open file limit**

  Upgrading Oracle Linux 8 systems with a low open file limit setting to Oracle Linux 9 might fail. You might see the following error messages (or similar messages) during the upgrade:

```
Traceback (most recent call last):
  File "/usr/lib/python3.6/site-packages/leapp/libraries/stdlib/__init__.py", line
185, in run
  File "/usr/lib/python3.6/site-packages/leapp/libraries/stdlib/call.py", line 199,
in _call
  File "/usr/lib/python3.6/site-packages/leapp/libraries/stdlib/call.py", line 73,
in _multiplex
  File "/usr/lib/python3.6/site-packages/leapp/libraries/stdlib/__init__.py", line
146, in _logfile_logging_handler
  File "/usr/lib64/python3.6/logging/__init__.py", line 1296, in debug
  File "/usr/lib64/python3.6/logging/__init__.py", line 1444, in _log
  File "/usr/lib64/python3.6/logging/__init__.py", line 1454, in handle
  File "/usr/lib64/python3.6/logging/__init__.py", line 1516, in callHandlers
  File "/usr/lib64/python3.6/logging/__init__.py", line 865, in handle
  File "/usr/lib/python3.6/site-packages/leapp/logger/__init__.py", line 40, in emit
  File "/usr/lib/python3.6/site-packages/leapp/logger/__init__.py", line 45, in
_do_emit
  File "/usr/lib/python3.6/site-packages/leapp/utils/audit/__init__.py", line 87, in
store
  File "/usr/lib/python3.6/site-packages/leapp/utils/audit/__init__.py", line 73, in
get_connection
  File "/usr/lib/python3.6/site-packages/leapp/cli/commands/upgrade/util.py", line
36, in wrapper
  File "/usr/lib/python3.6/site-packages/leapp/utils/audit/__init__.py", line 60, in
create_connection
  File "/usr/lib/python3.6/site-packages/leapp/utils/audit/__init__.py", line 27, in
_initialize_database
sqlite3.OperationalError: unable to open database file
```

  This is caused by an increase in the number of files in the Oracle Linux 9 ca-certificates package. In principle, similar issues can also occur if a package is updated that includes a substantially increased number of packaged files. The error appears when leapp reaches the `open files` limit during the Oracle Linux in-place upgrade to Oracle Linux 9.

  As a workaround, increase the ulimit `open files` variable on the Oracle Linux system before starting the upgrade. The value known to cause issues is 1024. For example, the following command shows a system still using the default `open files` limit:

```
ulimit -a | grep open
open files                      (-n) 1024
```

  Change the open file limit to 10000. Do the following:

```
ulimit -Sn 10000
```

Verify that the change has occurred. For example, the following command shows an Oracle Linux system with an `open files` limit of 10000:

```
ulimit -a | grep open
open files                      (-n) 10000
```

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 8 and Oracle Linux 9. The table helps you to identify the corresponding repositories that the Leapp utility makes available after the host has completed the upgrade.

Oracle Linux 8 DNF Repositories	Oracle Linux 9 DNF Repositories	Notes
`ol8_baseos_latest`	`ol9_baseos_latest`	All Oracle Linux 9 upgrades require the BaseOS and AppStream repositories.
`ol8_appstream`	`ol9_appstream`	
`ol8_kvm_appstream`	`ol9_kvm_utils`	
`ol8_UEKR7` `ol8_UEKR6`	`ol9_UEKR8`	Oracle Linux 9 requires UEK R7 as a minimum UEK version.
`ol8_addons`	`ol9_addons`	
`ol8_ksplice`	`ol9_ksplice`	Available for Oracle Cloud Infrastructure instances only.
`ol8_oci_included`	`ol9_oci_included`	Available for Oracle Cloud Infrastructure instances only.
`ol8_codeready_builder`	`ol9_codeready_builder`	Suggested for developer systems only.

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

Oracle Linux 8 KVM Stack	Oracle Linux 8 Repository	ModuleStream	Oracle Linux 9 KVM Stack	Oracle Linux 9 Repository	Notes
Default KVM Stack	`ol8_appstream`	`virt:ol`	Default KVM Stack	`ol9_appstream`	The default KVM stacks are available in both RHCK and UEK.
Oracle Linux 8 KVM for UEK	`ol8_kvm_appstream`	`virt:kvm_utils#`	Oracle Linux 9 KVM for UEK	`ol9_kvm_utils`	The Oracle KVM Utils stack is available in UEK only.

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

Oracle Linux 9 KVM Stack	Oracle Linux 9 Repository	ModuleStream	Oracle Linux 10 KVM Stack	Oracle Linux 10 Repository	Notes
Default KVM Stack	`ol9_appstream`	`virt:ol`	Default KVM Stack	`ol10_appstream`	The default KVM stacks are available in both RHCK and UEK.
Oracle Linux 9 KVM for UEK	`ol9_kvm_appstream`	`virt:kvm_utils#`	Oracle Linux 10 KVM for UEK	`ol10_kvm_utils`	The Oracle KVM Utils stack is available in UEK 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 'ol9_addons' --enablerepo 'ol9_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:

- `ol9_baseos_latest`
- `ol9_appstream`
- `ol9_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 'ol9_addons' --enablerepo
'ol9_codeready_builder' ...
```

This command is equal to :

```
sudo leapp preupgrade --enablerepo 'ol9_baseos_latest' --enablerepo 'ol9_appstream' --
enablerepo 'ol9_UEKR8' --enablerepo 'ol9_addons' --enablerepo 'ol9_codeready_builder'
```

`--oci`
This argument is used on Oracle Cloud Infrastructure instance upgrades. The repositories covered by this argument are a superset of the `--oraclelinux` argument.
When you use this argument, the following repositories are automatically enabled:

- `ol9_baseos_latest`
- `ol9_appstream`
- `ol9_addons`
- `ol9_ksplice`
- `ol9_oci_included`
- `ol9_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 `--oci`. For example:

```
sudo leapp preupgrade --oci --enablerepo 'ol9_codeready_builder' ...
```

This command is equal to:

```
sudo leapp preupgrade --enablerepo 'ol9_baseos_latest' --enablerepo 'ol9_appstream' --
enablerepo 'ol9_ksplice' --enablerepo 'ol9_oci_included' --enablerepo 'ol9_UEKR8' --
enablerepo 'ol9_addons' --enablerepo 'ol9_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