

Unbreakable Enterprise Kernel

Unbreakable Enterprise Kernel Release 7 - Release Notes (5.15.0-0.30)



F51024-09
March 2023



Unbreakable Enterprise Kernel Unbreakable Enterprise Kernel Release 7 - Release Notes (5.15.0-0.30),
F51024-09

Copyright © 2022, 2023, Oracle and/or its affiliates.

Contents

Preface

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

1 About Unbreakable Enterprise Kernel Release 7

Certification of UEK R7 for Oracle Products	1-1
Compatibility	1-2
Notable changes in kernel headers	1-2

2 New Features and Changes

Summary of Notable Changes in UEK R7	2-1
Default Page Size on Arm Platform Changed to 4 KB	2-1
Core Kernel Features and Functionality	2-2
Changes to UEK Content Distribution and Packaging	2-3
File Systems	2-4
Btrfs	2-4
ext4	2-5
OCFS2	2-5
XFS	2-6
WireGuard Communication Protocol	2-6
RDMA	2-7
Security	2-7
New .machine Kernel Keyring Introduced	2-7
SGX Enabled on Intel Architecture	2-8
Zonefs for Zoned Block Devices	2-8
Deprecated and Removed Features	2-8

3 Related User Space Packages

4 Known Issues

Unusable or Unavailable Features for the Arm Platform	4-1
dracut-install: ERROR: installing 'virtio' may be displayed during UEK R7 installation	4-1
Upgrading from UEK R6 to UEK R7 on Arm platform may fail if RAID 5 default page size differs from default stripe size	4-2
Swap partitions created on Arm platform and booted into earlier UEK release do not work after upgrade to UEK R7	4-2
Cloud-init and systemd-udev fail to rename mlx5_core network interfaces during upgrade from UEK R6 to UEK R7	4-3
Mellanox NIC interface name subject to change after upgrading from UEK R6 to UEK R7	4-3
Random high CPU utilization issue encountered with database benchmark program	4-4

5 Installation and Availability

About Upgrading From a Previous Oracle Linux or UEK Release to UEK R7	5-1
Obtaining Packages for Installation	5-2
Enabling Access to Oracle Linux Yum Server Repositories	5-3
Subscribing to ULN Channels	5-3
Upgrading a System to UEK R7	5-4
Installing and Upgrading Oracle-Supported RDMA Packages on Oracle Linux	5-5
Installing Oracle-Supported RDMA Packages on Oracle Linux 8	5-5
Installing Oracle-Supported RDMA Packages on Oracle Linux 9	5-6
Upgrading Oracle-Supported RDMA Packages on Oracle Linux 8 and Oracle Linux 9	5-7

6 Summary of Upstream History

Preface

[Unbreakable Enterprise Kernel Release 7: Release Notes \(5.15.0-0.30\)](#) provides a summary of the new features, significant changes, as well as any known issues in Unbreakable Enterprise Kernel Release 7 (UEK R7).

Conventions

The following text conventions are used in this document:

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

Documentation Accessibility

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

For information about the accessibility of the Oracle Help Center, see the Oracle Accessibility Conformance Report at <https://www.oracle.com/corporate/accessibility/templates/t2-11535.html>.

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 Unbreakable Enterprise Kernel Release 7

This chapter provides an overview of Unbreakable Enterprise Kernel Release 7 (UEK R7) and contains important information about this major release.

UEK R7 is initially released with the 5.15.0-0.30.19 version and build of the kernel. The kernel's source code is available through a public git source code repository at <https://github.com/oracle/linux-uek>.

The following is a general description of the scope of support for UEK R7:

- Although UEK R7 uses the same versioning model as the mainline Linux kernel version, it is possible that some applications might not understand the 5.15.0 versioning scheme. Note, however, that regular Linux applications are usually neither aware of nor affected by Linux kernel version numbers.
- The kernel is developed, built, and tested on the 64-bit Arm (aarch64), Intel 64-bit x86_64, and AMD 64-bit x86_64 architectures and is based on the mainline Linux kernel version 5.15.0.
- UEK R7 is made available for installation on Oracle Linux 8.5 and later releases. Note that UEK R7 is shipped with Oracle Linux 9.
- UEK R7 maintains compatibility with the Red Hat Compatible Kernel (RHCK) and does not disable any features that are enabled in RHCK. Additional features are enabled to provide support for key functional requirements and patches are applied to improve performance and optimize the kernel for use on Oracle operating environments. Note that Oracle actively monitors upstream check-ins and applies critical bug and security fixes to UEK R7.

Certification of UEK R7 for Oracle Products

The following important information applies to the certification of Oracle products with UEK R7.

Note that certification of different Oracle products with UEK R7 might not be immediately available at the time of the UEK R7 release. Be sure to always check to ensure that the product you are using is certified for use with UEK R7 before upgrading or installing the kernel. You can check for certification information at <https://support.oracle.com/epmos/faces/CertifyHome>.

Oracle Automatic Storage Management Cluster File System (Oracle ACFS) certification for different kernel versions is described in Document ID 1369107.1, which is available at <https://support.oracle.com/epmos/faces/DocumentDisplay?id=1369107.1>.

Oracle Automatic Storage Management Filter Driver (Oracle ASMFD) certification for different kernel versions is described in Document ID 2034681.1, which is available at <https://support.oracle.com/epmos/faces/DocumentDisplay?id=2034681.1>.

Compatibility

Oracle Linux maintains full user space compatibility with Red Hat Enterprise Linux (RHEL), which is independent of the kernel version that is running underneath the operating system. Note that existing applications in user space continue to run unmodified with UEK R7; no recertifications are required for RHEL certified applications.

To minimize any impact on interoperability during releases, the Oracle Linux team works closely with third-party vendors that have hardware and software with dependencies on kernel modules. The kernel ABI for UEK R7 will remain unchanged in all subsequent updates to the initial release. In this release, there are changes to the kernel ABI that are relative to UEK R6 and require recompilation of third-party kernel modules on the system. Before installing UEK R7, verify its support status with your application vendor.

Notable changes in kernel headers

Upstream changes to kernel headers may mean that third party modules do not compile across different kernel versions without modification to source code. Notably, the `memcg_cache_params` structure has been moved from `include/linux/slab.h` to `mm/slab.h`, which means that code needs to be refactored to account for the change if you are compiling across kernel versions.

To solve this problem so that the code can compile for UEK R6 and UEK R7, change the header requirements in the source code. For example, change lines like those in the following example to what is shown in the second example:

```
#ifndef CONFIG_SLUB
#include <linux/slub_def.h>
#endif

#if ( LINUX_VERSION_CODE < KERNEL_VERSION(5,4,0) )

#ifdef CONFIG_SLUB
#include <linux/slub_def.h>
#endif

#endif
```


2

New Features and Changes

This chapter describes new features, enhancements, and other notable changes that are introduced in UEK R7.

Summary of Notable Changes in UEK R7

The following is a summary of the features, changes, and improvements that are introduced in UEK R7, relative to UEK R6:

- **Linux 5.15 stable kernel base**

The 5.15.0 mainline kernel release that is used as the base kernel for UEK R7 includes many upstream kernel features and improvements over previous UEK releases and over RHCK. For a listing of the major new features and enhancements that are introduced in this kernel, see [Core Kernel Features and Functionality](#).

- **64-bit Arm (aarch64) support**

This release provides improved support for the 64-bit Arm (aarch64) platform. One significant change for the Arm platform is that the default page size has changed to 4 KB, from the previous 64 KB default. The new 4 KB size pairs well with the workloads and memory amounts that exist on the majority of Arm-based systems. See [Default Page Size on Arm Platform Changed to 4 KB](#) for more detailed information about this notable change.

- **DTrace v2.0**

Dtrace v2.0 continues to be available in UEK R7 and leverages kernel tracing facilities like eBPF. Detailed information about DTrace releases and other notable changes are available at [Oracle Linux: DTrace Release Notes](#).

- **File systems support**

Support for the Btrfs and OCFS2 file systems is enabled in UEK R7. The XFS and NFS file systems have also been enhanced in this release. For more information about new file systems features that are introduced in UEK R7, see [File Systems](#).

Default Page Size on Arm Platform Changed to 4 KB

The default page size on the 64-bit Arm (aarch64) platform has changed to 4 KB, from the previous 64 KB default. The new 4 KB size pairs well with the workloads and memory amounts that exist on the majority of Arm-based systems.

This change has important implications if you intend to upgrade kernel from a previous release of UEK. See [About Upgrading From a Previous Oracle Linux or UEK Release to UEK R7](#) for more information.

For information about other known issues that are related to this important change, see [Known Issues](#).

Core Kernel Features and Functionality

Several major, core kernel features have been implemented in the upstream kernel, between the 5.4 release that was used as the base kernel version for UEK R6 and the 5.15.0 kernel release that is used as the base kernel version for UEK R7. Although some features have been backported into the UEK R6 kernel in update releases, the following are the significant new features that are available in UEK R7:

- **BPF improvements**

UEK R7 introduces numerous Berkeley Packet Filter (BPF) improvements, including the following:

- The introduction of Compile Once Run Everywhere in `libbpf` and in LLVM and BPF Type Format (BTF). This change enables the BPF verifier to use in-kernel BTF to type check BPF assembly code, which provides for safer and faster BPF tracing.
- BPF trampoline, which is a feature that enables kernel code to call into BPF programs with nearly zero overhead, is available for the `x86_64` architecture. Support for memory-mapping BPF array map and other improvements are also introduced in this release.
- BPF support for calling kernel functions directly. This enhancement enables BPF programs that call kernel functions initially to reuse Transmission Control Protocol (TCP) congestion control implementations.
- BPF programs are capable of sleeping during executing, simplifying the mechanism to bind a socket to a range of addresses or port numbers. The new `BPF_PROG_TYPE_SK_LOOKUP` program type runs when the kernel is searching for an open socket for an incoming connection. The mechanism can then decide which socket should receive the connection. This mechanism has been added as a way to bind a socket to a range of addresses or port numbers in a more simple way. Currently, this feature is limited to tracing and security-module programs.

- **Core scheduling capability included**

Core scheduling provides the ability to isolate groups of processes that are running on the same core, ensuring maximum protection against side-channel attacks. You can use core scheduling as a method for preventing Spectra-class vulnerability attacks, while keeping Simultaneous Multithreading (SMT) enabled and avoiding a performance penalty for disabling SMT.

- **New cgroup slab memory controller**

UEK R7 introduces a new control group (cgroup) slab memory controller that enables you to share slab memory between memory cgroups. This new implementation of the slab memory controller aims to reach much better slab utilization by sharing slab pages between multiple memory cgroups. Also, accounting is performed per-object rather than per-page. The new capability saves a significant amount of memory, which greatly reduces inefficiencies.

- **io_uring enhancements**

The `io_uring` system call, which is a Linux API for asynchronous I/O, is designed for higher performance than the previous Linux AIO API that is supported by QEMU. Several enhancements for `io_uring` are introduced in UEK R7, the majority of which are focused around networked I/O.

One `io_uring` performance improvement that is related to I/O is a new BIO recycling mechanism for removing some internal memory-management overhead, which reportedly provides a 10% increase in the number of I/O operations per second that `io_uring` is capable of sustaining.

Other notable changes for `io_uring` include Oracle ASMLib v3 + `oracaleasm-support`. With this release, Oracle ASMLib uses the `io_uring` system call in place of the legacy `oracleasm` driver interface, which has been removed in UEK R7.

- **Split lock detection**

In this release, the split lock detection CPU feature is enabled by default on `x86_64` systems that have the capability for this functionality, such as the Ice Lake processor. The `split_lock_detect` boot command enables you to warn or send `SIGBUS` (Bus error signals) to applications that make use of split locks. A split lock occurs when an atomic CPU instruction operates on data that spans two cache lines. This operation is much slower than an atomic operation within a cache line, and it disrupts performance on other cores.

Note that you can disable this feature in the kernel by setting `split_lock_detect=off`.

Changes to UEK Content Distribution and Packaging

How content is distributed and packaged in the UEK release has changed. Starting with UEK R7, the kernel is repackaged and streamlined into several separate RPMs to facilitate particular hardware requirements. A meta RPM package, named `kernel-uek`, continues to be the required package for all UEK installations and maintains backward compatibility with previous releases.

By default, the `kernel-uek` package and its dependencies are installed. Installing this package is equivalent to installing the full UEK kernel.

The following table provides additional details about how UEK R7 content is distributed and packaged and includes information about package dependencies, as well as any other notable requirements.

Package	Description
<code>kernel-uek</code>	This is a meta package that does not contain any files. The package has a dependency on <code>kernel-uek-core</code> and <code>kernel-uek-modules</code> . Installing this package is equivalent to installing the full UEK kernel. This is the required package for all UEK installations and maintains compatibility with previous releases.
<code>kernel-uek-core</code>	This package contains the UEK kernel and a minimal number of kernel modules and is installed along with the <code>kernel-uek-modules</code> package. Note that this package requires that the <code>linux-firmware-core</code> package also be installed.

Package	Description
kernel-uek-modules	This package contains remaining kernel modules that are required by the majority of server configurations. Note that this package requires that the <code>linux-firmware</code> package also be installed.
kernel-uek-modules-extra	This is an optional package that contains modules for hardware and subsystems that are uncommon for servers, as well as support for certain devices, such as Bluetooth, Wi-Fi, and video capture cards. If support for any of these components is required, install the package manually from the yum repository. You can list the modules explicitly provided by this package by running: <pre>rpm -q -l kernel-uek-modules-extra</pre>
linux-firmware-core	This package contains core firmware components and is a dependency for the <code>kernel-uek-core</code> package.
linux-firmware	This package contains firmware components that are not provided in the <code>linux-firmware-core</code> package and is a dependency for the <code>kernel-uek-modules</code> package. Note that this package requires that the <code>linux-firmware-core</code> package also be installed.

File Systems

The following file systems features and enhancements are introduced in UEK R7:

Btrfs

Note:

The default page size on the 64-bit Arm platform has changed to 4 KB, from the previous 64 KB default. If you are running Oracle Linux 8 on an Arm-based system with an earlier UEK release, this change impacts Btrfs file systems, for example, systems that are running the Raspberry Pi on an Oracle Linux 8 developer image. Prior to upgrading to UEK R7, you must migrate your data and prepare to reformat your file systems to prevent any data loss and ensure that the system does not become unbootable. See [Default Page Size on Arm Platform Changed to 4 KB](#).

This release introduces several Btrfs performance and data recovery improvements, as well as some RAID 1 enhancements, support for Linux read-write semaphores, and checksum support. Other notable changes for Btrfs include the following:

- **New rescue mount option added**

A new rescue mount option to group all existing mount options for recovery. `usebackuproot` is now an alias for `rescue=usebackuproot`; `nologreplay` is an alias for `rescue=nologreplay`.

- **Asynchronous SSD trimming available**

Asynchronous SSD trimming is available in Btrfs. For performance and wear-leveling reasons, solid-state storage drives benefit from receiving notification when a disk block goes unused. This operation is referred to as *discard* or *trim* and is performed automatically by Btrfs. When a file is deleted, Btrfs notifies the drive that the blocks belonging to the file are no longer being used. Previously, these notifications occurred synchronously, meaning the trim notifications were sent before ending the delete operation, which can harm performance. These notifications are now sent asynchronously.

- **fsync() performance improvements**

The following `fsync()` improvements are introduced:

- Improved `fsync()` performance (12% decrease on max latency reported by `dbench`).
- Substantial speed-up of parallel `fsync` by reducing the number of checksum tree look-ups and contention.
- Substantial speed up of parallel `fsync` for files with `reflinked/deduped` extents. For jobs 16 to 1024, on average, throughput is improved by roughly 50%; runtime is decreased by about 30%.

- **Pre-fetch chunk tree leaves at mount support**

Pre-fetch chunk tree leaves at mount, which improves mount speed in multi-TB file systems.

- **fs-verity and ID mapping support**

Support for `fs-verity` and ID mapping, as well as the DAMON to improve capability for monitoring memory access patterns of specific processes, is included in this release. The `fs-verity` generic layer, which is already available in the `ext4` and `F2FS` file systems, provides transparent integrity and authenticity protection of read-only files. The feature also includes capability for ID-mapped mount points, which provides the ability to map the user and group IDs of one mount to another mount.

ext4

The `ext4` file system continues to be supported in UEK releases.

OCFS2

OCFS2 continues to be supported in UEK releases. For Oracle Linux 9, user space packages are updated to enable OCFS2 support on this release.

XFS

 **Note:**

The default page size on the 64-bit Arm platform has changed to 4 KB, from the previous 64 KB default. As a result, if you are running Oracle Linux 8 on an Arm-based system with an earlier UEK release, and you previously manually changed an XFS file system to a block size that is greater than 4 KB, you must migrate your data and prepare to reformat file systems prior to upgrading to UEK R7. Systems with XFS file systems that are configured to use a 4 KB block size are unaffected. See [Default Page Size on Arm Platform Changed to 4 KB](#).

The following notable XFS file system changes are introduced in UEK R7:

- **Enhancements for XFS 2038+ and DAX operations and DAX metadata reduction.**

Capability for XFS 2038+, per-file and per-directory DAX operations, and DAX metadata reduction has been added in this release.

- **XFS file system includes new features**

The XFS file system supports two new options: `bigtime` and `inobtcount`. The `bigtime` option provides support for timestamps beyond the year 2038; the `inobtcount` option reduces mount time on large file systems. By default, these options are disabled. To enable these options while creating an XFS file system, use the `mkfs.xfs` command as follows:

```
sudo mkfs.xfs -m bigtime=1,inobtcount=1
```

 **Caution:**

Enabling these `mkfs.xfs` options creates a file system that is unmountable by older kernels, where these options are not supported.

WireGuard Communication Protocol

The WireGuard communication protocol is available in UEK R7 for both IPv4 and IPv6 networks. WireGuard uses encrypted virtual private networks (VPNs) by passing traffic over the User Datagram Protocol (UDP).

 **Note:**

WireGuard was previously enabled as a technology preview feature in UEK R6U1, with full support introduced in UEK R6U3.

WireGuard uses public key encryption for identification and encryption, while OpenVPN uses certificates for these tasks. With WireGuard, secure key generation and management is handled in the background. Note that although IPsec is still the standard for secure network communication, WireGuard is gaining in popularity because it is simpler to configure, as well as deploy.

For more information and step-by-step instructions, see [Oracle Linux: Configuring Virtual Private Networks](#).

RDMA

UEK R7 includes Remote Direct Memory Access (RDMA) features that are provided in the upstream kernel, with the addition of Ksplice and DTrace functionality. RDMA enables direct memory access between two systems that are connected by a network. RDMA facilitates high-throughput and low-latency networking in clusters.

Starting with Oracle Linux 9, the process of installing Oracle-supported RDMA packages has been simplified through the use of new, user space packages and a dedicated ULN channel and yum repository for RDMA-related packages. For more information, see [Installing and Upgrading Oracle-Supported RDMA Packages on Oracle Linux](#).

If you are running Oracle Linux 8, the process of installing Oracle-supported RDMA packages remains the same as in previous releases.

Security

The following security features are introduced in UEK R7:

New .machine Kernel Keyring Introduced

The `.machine` kernel keyring is introduced in UEK R7. You can use this keyring as a kernel level trust anchor for any operation that uses asymmetrical keys. The `.machine` keyring has the same level of trust within the kernel as the `.builtin_trusted_keys` and the `.secondary_trusted_keys` keyrings. Similar to the built-in and secondary keyrings, you can use keys in the `.machine` keyring to do the following:

- Perform kernel module signature validation.
- Add additional keys to the `.secondary_trusted_keys` keyring.
- Serve as a CA for IMA appraisal keys.

Note that keys contained within the `.machine` keyring must be a root CA certificate. To qualify as a root CA certificate, the following two requirements must be met:

- The X.509 certificate shall be self-signed.
- The X.509 certificate shall contain X509v3 extensions with `basicConstraints=critical,CA:TRUE`

It is also highly advised the `keyUsage` field is set with `keyCertSign`.

Unlike previous UEK releases, keys contained within the `.platform` keyring in UEK R7 may only be used for `kexec`. Also, you may not use the `.platform` keys for any other purpose; whereas, in previous UEK releases, you could use the `.platform` keyring for kernel module signature validation, but they could not be used for any other kernel key operations.

The easiest way to add keys to the `.machine` keyring is by enrolling them with the `mokutil` utility. Note that this method requires that you first create an X.509 key pair.

SGX Enabled on Intel Architecture

Unbreakable Enterprise Kernel Release 7 enables Intel Software Guard Extensions (SGX) technology on the third-generation Intel Xeon Scalable processor (codename Ice Lake).

Applications can use this hardware functionality to populate protected regions of user code and data, called *enclaves*. When activated, the new hardware protects enclave code and data from outside access and modification. Enclaves provide a place to store secrets and then process data with those secrets, such as DRM software. SGX assists in providing protection against many known cybersecurity threats, thereby reducing the attack surface of servers through the use of secure enclaves, which protect information from processes running at a higher privilege.

Zonefs for Zoned Block Devices

The `zonefs` (zone file system) feature is a simple file system that exposes each zone of a zoned block device as a file. Unlike a regular POSIX-compliant file system with native zoned block device support, for example, `f2fs`, `zonefs` does not hide the sequential write constraint of zoned block devices to the user. Files that represent sequential write zones of the device must be written sequentially, starting from the end of the file (append only writes).

Deprecated and Removed Features

The following features are deprecated, removed, or no longer supported in UEK R7:

- **`/dev/raw` Device Removed**

The `/dev/raw` driver that was marked obsolete in the year 2005 has been removed in this release. In previous releases, the device nodes under `/dev/raw` provided an interface to direct I/O semantics for applications that were written prior to the introduction of the `O_DIRECT` file flag in Linux. This change also makes the `raw` command that is part of the `util-linux` package nonfunctional.

To obtain direct I/O file semantics, use the appropriate setting for the I/O file, as follows:

- For C, use the `O_DIRECT` flag as part of the `open()` system call.
- For Java, use the `RandomAccessFile` mode, "rwd", (Open for reading and writing).
- From the shell, you can use the `dd` flags, `iflag=direct` (for reading), and `oflag=direct` (for writing).

- **`resilient_rdmaip` Module Deprecated**

The `resilient_rdmaip` module is deprecated in UEK R7. This module may be removed in a subsequent UEK release.

- **Cisco `fnic 1.6` Driver Unsupported**

Cisco no longer supports the Cisco FCoE HBA Driver (`fnic 1.6`) that is sourced from the upstream kernel and which is available in most kernels, including UEK R5, UEK R6, and UEK R7. Cisco provides a fully supported UCS Linux driver (version 2.0.0.83, and later) that is tested on and compatible with Oracle Linux, with UEK R5 and later UEK releases, on the Cisco software download page. The driver package includes features that are not available in the currently included driver module such as NVMe support and multi-queue support.

Customers who are running Oracle Linux on Cisco servers must install the Cisco driver package to receive driver fixes, driver updates, new hardware support, and new feature support. Contact Cisco for more information about driver solutions on Oracle Linux.

- **oracleasm Kernel Module Removed**

The `oracleasm` kernel module is removed in UEK R7. Note that this module continues to be supported in the UEK R5 and UEK R6 releases.

Note that Oracle ASMLib continues to be supported through the use of `io_uring` interfaces. See [Core Kernel Features and Functionality](#) for more information.

- **DRBD Kernel Module Removed**

The DRBD (Distributed Replicated Block Device) kernel module and the `drbd-utils` package are removed in UEK R7. Note that the DRBD kernel module and its associated package, which was introduced as a technology preview in UEK R4 and deprecated in UEK R6U3, continues to be enabled in UEK R5 and UEK R6.

3

Related User Space Packages

To expose newly added functionality that is included in UEK R7, several user space binary packages are required. Some of these packages are separate from the packages that are included in the base distribution.

For more information about the ULN channels and Oracle Linux yum server repositories in which these packages are available, see [Installation and Availability](#).

The packages that are listed in the following table are specific to user space functionality. The versions listed are tested to take advantage of the features that are available in UEK R7. If you use any of the packages that are listed with UEK R7, you should update the package to the latest version for full compatibility with all of the features that are included in this release. Note that `btrfs-progs`, `ocfs2-tools` and the user space packages released in the UEK R7 repository are only supported with UEK R7 and should not be installed with the RHCK kernel.

Package Name	Oracle Linux 8 Version	Oracle Linux 9 Version
<code>btrfs-progs</code>	5.15.1 (x86_64, aarch64)	5.15.1 (x86_64, aarch64)
<code>ocfs2-tools</code>	1.8.6 (x86_64, aarch64)	1.8.6 (x86_64, aarch64)
<code>ndctl</code> , <code>ndctl-libs</code> , <code>ndctl-devel</code> , <code>daxctl</code> , <code>daxctl-libs</code> , <code>daxctl-devel</code>	73-1 (x86_64)	72-1 (x86_64)
<code>ipmctl</code> , <code>ipmctl-monitor</code> , <code>libipmctl</code> , <code>libipmctl-devel</code>	03.00.00.0427 (x86_64)	03.00.00.0427 (x86_64)
<code>mcelog</code>	179 (x86_64)	179 (x86_64)
<code>pciutils</code>	3.7.0 (x86_64)	3.7.0 (x86_64)
<code>dmidecode</code>	3.3 (x86_64)	3.3 (x86_64)
<code>wireguard-tools</code>	1.0.20210914 (x86_64, aarch64)	1.0.20210424 (x86_64) 1.0.20210914 (aarch64)
<code>adaptivemm</code>	2.0.1 (x86_64, aarch64)	2.0.1 (x86_64, aarch64)
<code>bcache-tools</code>	1.0.8 (x86_64, aarch64)	1.0.8 (x86_64, aarch64)
<code>libbpf</code> , <code>libbpf-static</code> , <code>libbpf-devel</code>	0.6.0 (x86_64, aarch64)	0.6.0 (x86_64, aarch64)
<code>snapper</code>	0.8.7 (x86_64)	0.8.7 (x86_64)
<code>dnf-plugins-extras</code>	4.0.8 (x86_64, aarch64)	?
<code>xfs-progs</code> , <code>xfsprogs-devel</code>	5.15.9 (x86_64, aarch64)	5.15.0 (x86_64, aarch64)
<code>e2fsprogs</code>	1.46.2 (x86_64, aarch64)	1.46.5 (x86_64, aarch64)
<code>crash</code>	8.0.0 (x86_64, aarch64)	8.0.0 (x86_64, aarch64)
<code>kexec-tools</code> , <code>kdump</code>	2.0.20 (x86_64)	2.0.23 (x86_64, aarch64)
<code>iproute</code>	5.15.0 (x86_64, aarch64)	5.15.0 (x86_64, aarch64)

Package Name	Oracle Linux 8 Version	Oracle Linux 9 Version
nvme-cli	1.16 (x86_64, aarch64)	1.16 (x86_64, aarch64)
bpftool	5.15.0 (x86_64, aarch64)	5.14.0 (x86_64, aarch64)
bcc, bcc-tools, bcc-docs, libbpf-tools, python3-bcc	0.23 (x86_64, aarch64)	0.23 (x86_64, aarch64)

4

Known Issues

This chapter describes any known issues for Unbreakable Enterprise Kernel Release 7.

Unusable or Unavailable Features for the Arm Platform

The following are specific features that are known to not work, remain untested, or have issues that render the feature unusable.

- **InfiniBand**

InfiniBand hardware is currently not supported for the Arm architecture when using UEK R7.

- **FibreChannel**

FibreChannel hardware is currently not supported for the Arm architecture when using UEK R7.

- **RDMA**

RDMA is not supported on the Arm platform.

- **Secure Boot and Lockdown**

The Secure Boot and Kernel Lockdown features are currently not supported or available for the Arm platform.

dracut-install: ERROR: installing 'virtio' may be displayed during UEK R7 installation

In UEK R7, `virtio` is not built as a module, but is built directly into the kernel. As such, you do not have to specify `virtio` in the dracut configuration file to add it to `initramfs`. If you previously had dracut configuration that included this module, attempting to install UEK R7 displays the following dracut error:

```
dracut-install: ERROR: installing 'virtio'
dracut: FAILED: /usr/lib/dracut/dracut-install -D
/var/tmp/dracut.FOKWjy/initramfs --kerneldir
/lib/modules/5.15.0-0.21.1.el8uek.x86_64/ -m xen_netfront xen_blkfront
virtio_blk virtio_net virtio virtio_pci virtio_balloon hyperv_keyboard
hv_netvsc hid_hyperv hv_utils hv_storvsc hyperv_fb ahci libahci
dracut-install: ERROR: installing 'virtio'
dracut: FAILED: /usr/lib/dracut/dracut-install -D
/var/tmp/dracut.G2XSGh/initramfs --kerneldir
/lib/modules/5.15.0-0.21.1.el8uek.x86_64/ -m xen_netfront xen_blkfront
virtio_blk virtio_net virtio virtio_pci virtio_balloon hyperv_keyboard
hv_netvsc hid_hyperv hv_utils hv_storvsc hyperv_fb ahci libahci
```

This error is displayed, regardless of whether you use the `yum` or `rpm` command to install UEK R7.

To work around the issue, prior to installing UEK R7, remove the "virtio" text from the dracut configuration file. Make sure to remove *only* the "virtio" text, leaving all other "virtio_*" entries intact, for example:

```
cat /etc/dracut.conf.d/01-dracut-vm.conf

add_drivers+=" xen_netfront xen_blkfront "
add_drivers+=" virtio_blk virtio_net virtio virtio_pci virtio_balloon "
add_drivers+=" hyperv_keyboard hv_netvsc hid_hyperv hv_utils hv_storvsc
hyperv_fb "
add_drivers+=" ahci libahci "
```

Use the following command to verify that `virtio` is built into the kernel:

```
grep CONFIG_VIRTIO= /boot/config-5.15.0-0.30.4.el8uek.x86_64
```

If `virtio` is built into the kernel, the output should be as follows:

```
CONFIG_VIRTIO=y
```

(Bug ID 33834972)

Upgrading from UEK R6 to UEK R7 on Arm platform may fail if RAID 5 default page size differs from default stripe size

Starting with UEK R7, the default page size on the Arm platform has changed to 4 KB, from the previous 64 KB default. This change in page size might cause an upgrade from UEK R6 to UEK R7 to fail on systems that are configured for RAID 5 when the default page size differs from the default stripe size.

For this reason, prior to upgrading from UEK R6 to UEK R7, you should back up and reformat RAID 5 volumes. In cases where retaining the same RAID 5 configuration is preferred, it is recommended that you continue to run UEK R6.

See [Default Page Size on Arm Platform Changed to 4 KB](#) for additional information.

(Bug ID 33858264)

Swap partitions created on Arm platform and booted into earlier UEK release do not work after upgrade to UEK R7

This release includes a significant change for the Arm platform regarding the default page size, which has changed to 4 KB, from the previous 64 KB default. Consequently, any swap partitions that were created on the Arm platform and booted into an earlier UEK release, for example, UEK R6, do not work after upgrading to UEK R7.



Note:

This issue is encountered on the Arm platform, irrespective of file system type.

Upon the first boot into UEK R7 after an upgrade, the following `systemd` service failure is indicated:

```
systemctl list-units --failed
UNIT LOAD ACTIVE SUB DESCRIPTION

dev-mapper-ol_myhost\x2dswap.swap loaded failed failed
/dev/mapper/ol_myhost-swap
```

To work around this issue, you must reinitialize the swap device with the new page size after upgrading to UEK R7. Use the `swapon` command as follows and specify the swap location:

```
sudo swapon --fixpgsz /dev/mapper/ol_myhost-swap

swapon: /dev/mapper/ol_myhost-swap: swap format pagesize does not match.
swapon: /dev/mapper/ol_myhost-swap: reinitializing the swap.
mkswap: /dev/mapper/ol_myhost-swap: warning: wiping old swap signature.
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=d7ef0a33-403f-447b-863f-d52b7f66c803
```

In the previous command, `/dev/mapper/ol_myhost-swap` is an example of a typical swap location that you might specify.

For more information about the important change in default page size for the Arm platform in UEK R7, see [Default Page Size on Arm Platform Changed to 4 KB](#).

(Bug ID 34322552)

Cloud-init and systemd-udev fail to rename mlx5_core network interfaces during upgrade from UEK R6 to UEK R7

During an upgrade from UEK R6 to UEK R7 on an Oracle Infrastructure instance, `cloud-init` and `systemd-udev` revert to using the older UEK R6 device naming scheme (`ifcfg-ens300f0`) for the `mlx5_core` network interface, rather than correctly renaming the device with the new UEK R7 device naming scheme (`ens300f0np0`).

To ensure that the `mlx5_core` network interface does not revert to using the former UEK R6 device naming scheme, do the following after the upgrade to UEK R7 has completed, prior to rebooting the system:

1. Remove the old network configuration file, for example:

```
sudo rm /etc/sysconfig/network-scripts/ifcfg-ens300f0
```

2. Remove any cached data saved by `cloud-init`:

```
sudo cloud-init clean
```

3. Reboot the instance for the changes to take effect.

(Bug ID 34146775)

Mellanox NIC interface name subject to change after upgrading from UEK R6 to UEK R7

During a kernel upgrade from UEK R6 to UEK R7, the `mlx5_core` device name is subject to change, from `ens2f0` (UEK R6) to `ens2f0np0` (UEK R7).

You might encounter this issue under the following circumstances:

- When upgrading an Oracle Linux 8 system that is running UEK R6 to UEK R7.
- When upgrading an Oracle Linux 8 system that is running UEK R6 to Oracle Linux 9 (which ships with UEK R7 by default).
- When upgrading an Oracle Linux 8 system that is already running UEK R7 to Oracle Linux 9.

 **Note:**

In the case where an Oracle Linux 8 system is already running UEK R7, if you previously configured the system to use backwards-compatible device names (`ens2f0`), you might need to apply the workaround that follows to your GRUB configuration after the upgrade to Oracle Linux 9 has completed.

Note that fresh installations of UEK R7 on Oracle Linux 8 and Oracle Linux 9 use the default naming convention for UEK R7 (`enp2s0f0np0`) by default.

To retain backwards-compatible (UEK R6) device names for the `mlx5_core` driver-based network interface card (NIC), perform the following workaround after upgrading to UEK R7, prior to rebooting your system. It is recommended that you back up your existing `grub.cfg` file before making this change.

1. Edit the `/etc/default/grub` file and append the end of the line in the `GRUB_CMDLINE_LINUX=` module as follows:


```
GRUB_CMDLINE_LINUX="console=xxxx mlx5_core.expose_pf_phys_port_name=0"
```
2. After editing the file, locate the `grub.cfg` file on your system, then run the command to update GRUB configuration, as appropriate:
 - On BIOS-based systems, the `grub.cfg` output/target file is usually located at `/boot/grub2/grub.cfg` and you would run the following command:


```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```
 - On UEFI-based systems, the `grub.cfg` output/target file could be located at `/etc/grub2-efi.cfg` or `/boot/efi/EFI/redhat/grub.cfg`. Depending on the location of the file, you would run one of the following commands:


```
sudo grub2-mkconfig -o /etc/grub2-efi.cfg
```

```
sudo grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```
3. Reboot the system for the changes to take effect.

(Bug IDs 34103369, 34145887)

Random high CPU utilization issue encountered with database benchmark program

A random high CPU utilization issue has been encountered with the database benchmark program running on a 192-CPU virtual machine in Azure. This issue was

initially discovered in Oracle Linux 8.4 and Ubuntu 20.04 (5.11.0-1022-azure); however, a complete fix for the issue is not yet available in the upstream kernels.

This issue typically manifests itself with a >90% CPU utilization spike occurring every 1 to 2 minutes and lasting approximately 5 to 20 seconds, which degrades the system's performance significantly. When the CPU utilization spike is occurring, *each* of the 192 CPUs' %sys increases up to 60+%, and the %si increases up to 30%. In certain cases, the >90% CPU utilization spike has been observed 100% of the time.

To avoid encountering this issue, set the `dm_mod.dm_mq_queue_depth=256` kernel parameter.

(Bug ID 33665982)

5

Installation and Availability

This chapter provides information about the availability of UEK R7 on Oracle Linux and includes installation and instructions on upgrading from a previous UEK release to UEK R7.

UEK R7 is supported on the Intel 64-bit x86_64, AMD 64-bit x86_64 and 64-bit Arm (aarch64) platforms.

About Upgrading From a Previous Oracle Linux or UEK Release to UEK R7

UEK R7 is made available for installation on Oracle Linux 8, starting with the Oracle Linux 8.5 release. By default, Oracle Linux 9 ships with UEK R7.

The suggested migration path for upgrading the system from an earlier UEK release to UEK R7 is as follows:

- If you are running Oracle Linux 7 with an earlier UEK release, upgrade the operating system to the latest Oracle Linux 8 release. For instructions on upgrading the Oracle Linux 7 system, see [Oracle Linux 8: Performing System Upgrades With Leapp](#).
- If you are running an Oracle Linux 8 release that's earlier than Oracle Linux 8.5 with UEK R6, first upgrade the system to the latest Oracle Linux 8 update release. From here, you can upgrade to UEK R7. If you are already running Oracle Linux 8.5 or later with UEK R6, you can directly upgrade the system to UEK R7.

For instructions on upgrading an Oracle Linux 8 system to Oracle Linux 9, see [Oracle Linux 9: Performing System Upgrades With Leapp](#).

! Important:

In UEK R7, the default page size for the 64-bit Arm (aarch64) architecture has changed to 4 KB default, from the previous 64 KB default. The new 4 KB default page size might have significant implications on Arm-based systems that are running Oracle Linux 8 with an earlier UEK release, with either a Btrfs or an XFS file system.

- If an Arm-based system uses a Btrfs or an XFS file system, and you are running Oracle Linux 8 with an earlier UEK release, you might not be able to upgrade to UEK R7 without first migrating data to an alternative file system. The default on-disk file system block size is set to be the equivalent of the page size for these file systems, which means that the change in page size can render the file system inaccessible and can cause data corruption.

Note, however, that Oracle has placed checks within the UEK R7 Arm RPM that prevent the installation of UEK R7 if a Btrfs file system is detected and the resulting change in block size could cause data to become inaccessible.

- For an XFS file system, the default block size is 4 KB. XFS enables you to manually set the block size at file system creation time. If you have XFS file systems with a block size greater than 4 KB, you are required to migrate data before upgrading to UEK R7.

Typically, a data migration plan might involve adding another storage device, formatting it with an unaffected file system or using XFS with the block size specified as 4 KB, and then moving your data onto the newly formatted device.

- Users of the Oracle Linux 8 developer image installed on Raspberry Pi systems are necessarily affected because the image uses a Btrfs file system, by default. If you are using this image, and you intend to upgrade to UEK R7, you must migrate data to an alternative unaffected file system before attempting to install UEK R7. For more information about using the Raspberry Pi hardware platform, see [Install Oracle Linux on a Raspberry Pi](#).
- also important to note that any existing swap partitions that were created on the Arm platform and booted into an earlier UEK release, such as UEK R6, don't work after upgrading to UEK R7. Consequently, the change to a 4 KB default page size on the aarch64 platform requires that any existing swap partitions on the system *must* be reinitialized with the new page size after booting the system with UEK R7. For further details, see [Swap partitions created on Arm platform and booted into earlier UEK release do not work after upgrade to UEK R7](#).

For general information about working with file systems in Oracle Linux 8, see [Oracle Linux 8: Managing Local File Systems](#).

Obtaining Packages for Installation

If you have a subscription to Oracle Unbreakable Linux support, you can obtain the packages for UEK R7 by registering your system with the Unbreakable Linux Network

(ULN) and then subscribing it to additional channels. See [Subscribing to ULN Channels](#).

If your system is not registered with ULN, you can obtain most of the required packages from the Oracle Linux yum server. See [Enabling Access to Oracle Linux Yum Server Repositories](#).

When you have subscribed your system to the appropriate ULN channels or to the Oracle Linux yum server, you can proceed to upgrade your system to UEK R7. See [Upgrading a System to UEK R7](#).

Enabling Access to Oracle Linux Yum Server Repositories

Packages for UEK R7 and any associated user space applications are available on the Oracle Linux yum server at <https://yum.oracle.com/>.

For Oracle Linux 8, the kernel images and all of the associated user space packages for both the x86_64 and aarch64 platforms are made available by enabling the following repositories:

- `ol8_UEKR7`
- `ol8_baseos_latest`
- `ol8_addons`

For Oracle Linux 9, the kernel images and all of the associated user space packages for both the x86_64 and aarch64 platforms are made available by enabling the following repositories:

- `ol9_UEKR7`
- `ol9_baseos_latest`
- `ol9_addons`

To enable access to repositories on the Oracle Linux yum server, use the `dnf config-manager` command and specify the appropriate repositories for the release that you are running.

For example, you would enable access to the Oracle Linux 8 repositories as follows:

```
sudo dnf config-manager --enable ol8_baseos_latest ol8_addons ol8_UEKR7
```



Note:

You can only use the `dnf config-manager` to enable or disable repositories that already have a configuration file for the specified repository. Repository configurations are typically stored in the `/etc/yum.repos.d` file. The repository configurations that are required to install the UEK release on Oracle Linux 8 and Oracle Linux 9 are included in the `oraclelinux-release-el8` and `oraclelinux-release-el9` packages, respectively. Note that you might need to update the package to the latest version to obtain the correct yum repository configuration.

Subscribing to ULN Channels

For Oracle Linux 8, kernel image and user space packages are made available for the x86_64 platform in the following ULN channels:

- `ol8_x86_64_UEKR7`

- `ol8_x86_64_baseos_latest`

For Oracle Linux 8, kernel image and user space packages are made available for the aarch64 platform in the following ULN channels:

- `ol8_aarch64_UEKR7`
- `ol8_aarch64_baseos_latest`

For Oracle Linux 9, kernel image and user space packages are made available for the x86_64 platform in the following ULN channels:

- `ol9_x86_64_UEKR7`
- `ol9_x86_64_baseos_latest`

For Oracle Linux 9, kernel image and user space packages are made available for the aarch64 platform in the following ULN channels:

- `ol9_aarch64_UEKR7`
- `ol9_aarch64_baseos_latest`

The following instructions assume that you have previously registered your system with ULN.

To subscribe a system to a ULN channel:

1. Sign in to <https://linux.oracle.com> with a ULN username and password.
2. On the Systems tab, in the list of registered machines, click the link that corresponds to the name of the system.
3. On the System Details page, click **Manage Subscriptions**.
4. On the System Summary page, from the list of available channels, select each of the required channels, then click the right arrow to move the selected channel to the list of subscribed channels.
5. Click **Save Subscriptions**.

For more information about using ULN, see [Oracle Linux: Managing Software on Oracle Linux](#).

Upgrading a System to UEK R7

The following instructions describe how to upgrade a system to UEK R7. For more details about the suggested migration paths for upgrading to UEK R7, see [About Upgrading From a Previous Oracle Linux or UEK Release to UEK R7](#).

1. Enable access to the appropriate ULN channels or yum repositories, as described in [Subscribing to ULN Channels](#) and [Enabling Access to Oracle Linux Yum Server Repositories](#).

Tip:

Disable any other UEK channels or repositories that you might have previously configured as good practice.

2. After enabling access to the appropriate channels or repositories, upgrade the system to UEK R7 by running the following commands:

```
sudo dnf install -y kernel-uek
sudo dnf update -y
```

3. After the upgrade has completed, reboot the system.

Ensure to select the UEK R7 kernel (version 5.15.0) if it's not the default boot kernel.

For questions regarding installing software or updating a system, see [Oracle Linux: Managing Software on Oracle Linux](#).

Installing and Upgrading Oracle-Supported RDMA Packages on Oracle Linux

The following instructions describe how to install and upgrade Oracle-supported RDMA packages on Oracle Linux 8 and Oracle Linux 9.

Installing Oracle-Supported RDMA Packages on Oracle Linux 8



Note:

These instructions apply to the x86_64 platform.

The following instructions describe how to install RDMA release packages (`oracle-rdma-release`) on an Oracle Linux 8 system. These instructions include steps on how to remove other previously installed RDMA packages that could cause conflicts when installing the `oracle-rdma-release` packages.

If you are running Oracle Linux 9, see [Installing Oracle-Supported RDMA Packages on Oracle Linux 9](#) for instructions.

1. Subscribe your system to the appropriate RDMA ULN channel or yum repository.
 - If you are using the Oracle Linux yum server, enable the `ol8_UEKR7_RDMA` repository for Oracle Linux 8, for example:

```
sudo dnf config-manager --enable ol8_baseos_latest ol8_UEKR7 ol8_UEKR7_RDMA
```

- If you are using ULN, subscribe to `ol8_x86_64_UEKR7_RDMA` channel.

For additional instructions, see [Subscribing to ULN Channels](#) and [Enabling Access to Oracle Linux Yum Server Repositories](#).

2. Remove any existing packages that are related to RDMA, for example:

```
sudo dnf remove 'ibacm*'
sudo dnf remove 'ibutils*'
sudo dnf remove 'infiniband-diags*'
sudo dnf remove 'libibacl*'
sudo dnf remove 'libibcm*'
sudo dnf remove 'libibmad*'
sudo dnf remove 'libibumad*'
sudo dnf remove 'libibverbs*'
sudo dnf remove 'librdmacm*'
```

```

sudo dnf remove 'mstflint*'
sudo dnf remove 'opensm*'
sudo dnf remove 'oracle-rdma-tools'
sudo dnf remove 'perftest*'
sudo dnf remove 'qperf*'
sudo dnf remove 'rdma*'
sudo dnf remove 'rds-tools*'
sudo dnf remove 'rdma-core*'

```

3. Clean the yum cached files from all of the enabled repositories:

```
sudo dnf clean all
```

4. Install the RDMA packages for UEK R7.

- If you are installing the packages on a bare-metal system, use the following command:

```
sudo dnf install oracle-rdma-release
```

- If you are installing the packages on a virtual platform (either a Xen hypervisor or a KVM guest), use the following command:

```
sudo dnf install oracle-rdma-release-guest
```

- (Optional) If you require the `libpcap` package, you must install this package separately:

```
sudo dnf install libpcap
```

Each UEK release requires a different set of RDMA packages. If you change the kernel on your system to a UEK release that is earlier than UEK R7, use the following command to remove the existing UEK-based RDMA packages before installing the correct packages for the new kernel:

```
sudo dnf remove --setopt=clean_requirements_on_remove=1 oracle-rdma-release
```

Note that the previous command might not work for all of the related packages. For example, in Oracle Linux 8, the `libpcap` package is a dependency for key system packages and therefore cannot be removed. Instead, you can use the `dnf history undo` command as follows to roll back and remove the dependencies for the `rdma-core` package:

```
sudo dnf history undo rdma-core
```

 **Caution:**

Downgrading UEK versions is not advised, except for testing purposes.

Installing Oracle-Supported RDMA Packages on Oracle Linux 9

 **Note:**

These instructions apply to the x86_64 platform.

The process of installing Oracle-supported RDMA packages on Oracle Linux 9 has been simplified through the use of new, user space packages, as well as a dedicated ULN channel and yum repository for RDMA-related packages.

If you are running Oracle Linux 8, the process of installing Oracle-supported RDMA packages remains the same as it was in previous releases. For instructions, see [Installing Oracle-Supported RDMA Packages on Oracle Linux 8](#).

The following instructions describe how to install RDMA release packages (`oracle-rdma-release`) on an Oracle Linux 9 system:

1. Ensure that you have subscribed to the ULN channel or have enabled the yum repository that contains the RDMA-related user space packages for Oracle Linux 9.
 - If you are installing packages from ULN, subscribe to the `o19_x86_64_RDMA` channel.
 - If you are installing packages from the Oracle Linux yum server, enable the `o19_RDMA` yum repository.

2. Clean the yum cached files from all of the enabled repositories by running the following command:

```
sudo dnf clean all
```

3. Install the RDMA packages for UEK R7:

- If you are installing the packages on a bare-metal system, run the following command:

```
sudo dnf install oracle-rdma-release
```

- If you are installing the packages on a virtualized platform (either on a Xen hypervisor or KVM guest), run the following command:

```
sudo dnf install oracle-rdma-release-guest
```

4. (Optional) If you require the `libpcap` package, you must install this package separately:

```
sudo dnf install libpcap
```

Upgrading Oracle-Supported RDMA Packages on Oracle Linux 8 and Oracle Linux 9

You can upgrade the Oracle-supported RDMA packages on Oracle Linux 8 and Oracle Linux 9 by using the `dnf update` command.

If you are upgrading a system that has the `oracle-rdma-release` or `oracle-rdma-release-guest` package installed, if the package version is lower than version 0.18.1-1 and you intend to upgrade to version 0.18.1-1, or later, you must first manually remove the `rdma-core-devel` package. You should remove this package by using the `rpm -e --nodeps` command, which removes the package outside of the standard yum or DNF package manager control and leaves any dependencies intact, for example:

```
sudo /bin/rpm -e --nodeps rdma-core-devel
sudo dnf update
```

6

Summary of Upstream History

This appendix provides information about the major features and enhancements in each subsequent upstream kernel release. The following are the major upstream kernel features that are included in UEK R7, by virtue of the selection of the upstream 5.15.0 kernel base. Note that some of the key features that are listed here may have also been backported into the most recent UEK R6 release. For more details about these features, see <https://kernelnewbies.org/LinuxVersions>:

5.5

- `io_uring` enhancements
- Btrfs enhancements (RAID1, checksumming)
- BPF improvements

5.6

- Wireguard VPN (Technology Preview in UEK R6U1)
- USB4 support
- Btrfs SSD trimming
- `vboxsf` (VirtualBox Shared Folder, for guests)
- MultiPath TCP
- Intel Data Streaming Accelerator

5.7

- Arm Kernel Pointer Authentication
- Split lock detection (in UEK R6U1)
- vDPA support

5.8

- XFS: Per file/per directory DAX ops
- vDPA improvements
- Paravirtualized memory hot(un)plug

5.9

- Slab memory controller for cgroups
- `io_uring` enhancements
- Btrfs enhancements
- NFS v4.2 User `xattr` support
- KVM: disable PV spinlocks
- NVMe Support for the Zoned Namespace (ZNS)
- vDPA for Mellanox Devices

5.10

- XFS: No more Y2038 problems
- Btrfs: Performance improvements
- AMD Secure Nested Paging IOMMU
- AMD SEV-ES Support
- Intel Sapphire Rapids
- Arm Memory Tagging Extension
- BPF improvements
- vDPA Mellanox improvements
- `io_uring` enhancements
- Multi-Path TCP enhancements
- `devlink` enhancements

5.11

- Intel SGX Support (base driver)
- Btrfs improvements (data recovery, performance)
- `io_uring` enhancements
- Networking: Preferred busy-polling
- Secure Encrypted Virtualization (SEV-ES) KVM guests

5.12

- Btrfs enhancements (zoned devices, subpage block size for read-only)
- Kernel thread-based NAPI polling
- CXL 2.0 Support
- `io_uring` enhancements
- vDPA management tool
- Core PMU support for Sapphire Rapids
- AMD SEV attestation command
- KVM: Upgrade PMU support to ARMv8.4 (next-gen Ampere)
- `mlx5` vDPA subfunctions
- `devlink` enhancements

5.13

- Intel Bus Lock Detection
- SGX for KVM guests and (basic) NUMA
- `io_uring` enhancements (multi-shot)
- AMD SEV Guest Live Migration
- `mlx5` TC sample offload (is this interesting for DB/RDMA?)
- BPF enhancements (call kernel functions)

- Btrfs enhancements (readahead for send, perf improvement)
- vDPA enhancements, virtio-vdpa

5.14

- XFS enhancements (cache flushes, log recovery)
- Multi-Path TCP enhancements
- Secret memory areas (memfd_secret)
- Core scheduling
- virtio-IOMMU
- Raspberry Pi 400

5.15

- AMD 5-level paging
- vDPA MultiQueue
- `io_uring` enhancements