Intel® Virtual RAID on CPU (Intel® VROC) for Linux*

Release Notes for Intel® VROC 9.0

Revision 005

June 2024

Contents

Tables

Revision History

Revision	Description	Revision Date
001	Initial public release	July 2023
002	Updates for Intel® VROC 8.2 Release	August 2023
003	Updates for Intel® VROC 8.6 Release	March 2024
004	Updates for tSATA support in Ubuntu*	May 2024
005	Updates for Intel® VROC 9.0 Release	June 2024

§§

1 Introduction

1.1 Overview

The Intel® Virtual RAID on CPU (Intel® VROC) Linux* 9.0 release is a family of products that provides enterprise RAID solutions targeted for the Intel® Xeon® 6 Processor Family based platforms (code name: *Sierra Forest and Granite Rapids*) that support the Intel® Volume Management Device (Intel® VMD) technology. The product family includes the following products:

1. Intel® VROC (VMD NVMe RAID) Linux* – This product provides an enterprise RAID solution on platforms that support the Intel® VMD technology. This functionality is developed for and incorporated within Linux* MDRAID.
2. Intel® VROC (SATA RAID) Linux* – This product provides an enterprise RAID solution for SATA devices connected to the SATA controllers on the Intel® Platform Control Hub (Intel® PCH) configured for RAID mode. This functionality is developed for and incorporated within Linux* MDRAID.

These products can be used independently. For example (unless otherwise restricted by the platform itself), Intel® VMD is not required to be enabled to use Intel® VROC (SATA RAID) Linux*. Likewise, Intel® VROC (VMD NVMe RAID) Linux* can be enabled and used without Intel® VROC (SATA RAID) Linux*.

1.2 Terminology

Table 1-1. Terminology

Term	Description
API	Application Programming Interface
BIOS	Basic Input/Output System
Array	This term is representative of a *mdadm* container required for Intel® metadata-based volumes using the IMSM option during volume creation.
Container	A container is a type of array used with Intel® metadata or other non-native metadata.
GB	Gigabyte
GiB	Gibibyte (1024 x 1024 x 1024 bytes)
GA	General Access – Operating System release package fully validated by Red Hat Linux*.
HII	Human Interface Infrastructure
Hot-Plug	The unannounced removal and insertion of a drive while the system is powered on.
I/O	Input/Output

Term	Description
Initramfs	Initial RAM File System
IMSM	Intel® Matrix Storage Manager
KB	Kilobyte
KiB	Kibibyte (1024 bytes)
Left-Symmetric	Default layout scheme for RAID 5 configurations. Not supported with IMSM metadata
Left-Asymmetric	Parity bit layout scheme used in RAID 5 configurations.
Matrix RAID	Two different RAID volumes within a single RAID array container.
MB	Megabyte
MiB	Mebibyte (1024 x 1024 bytes)
MD	Linux* kernel Multiple Device driver
Member	A SATA or NVMe drive used within a RAID array.
Mdadm	*mdadm* is a Linux* utility developed to manage software RAID devices on Linux*. It is available under the GPL license version 2 or later and supports SATA and NVMe SSDs.
NVMe	Non-Volatile Memory Express
OS	Operating System
OSV	Operating System Vendor (e.g., Red Hat, SUSE)
Pre-OS	A BIOS component to configure Intel® VROC RAID.
PV	Production Version
RAID	Redundant Array of Independent Disks. Allows data to be distributed across multiple drives to provide data redundancy or to enhance data storage performance.
RAID 0 (striping)	The data in the RAID volume is striped across the array's members. Striping divides data into units and distributes those units across the members without creating data redundancy but improving read/write performance.
RAID 1 (mirroring)	The data in the RAID volume is mirrored across the RAID array's members. Mirroring is the term used to describe the key feature of RAID 1, which writes duplicate data from one drive to another; therefore, creating data redundancy and increasing fault tolerance.
RAID 5 (striping with parity)	The data in the RAID volume and parity are striped across the array's members. Parity information is written with the data in a rotating sequence across the members of the array. This RAID level is a preferred configuration for efficiency, fault-tolerance, and performance.
RAID 10 (striping and mirroring)	The RAID level where information is striped across a two drive arrays for system performance. Each of the drive in the array has a mirror for fault tolerance. RAID 10 provides the performance benefits of RAID 0 and the redundancy of RAID 1. However, it requires four hard drives so it’s the least cost effective.
RAID Array	A logical grouping of physical drives.

Term	Description
RAID Volume	A fixed amount of space across a RAID array that appears as a single physical drive to the operating system. Each RAID volume is created with a specific RAID level to provide data redundancy or to enhance data storage performance.
Recovery Drive	The drive that is the designated target drive in a recovery volume.
Hot Spare Drive	Hot spare drive is a disk or group of disks used to automatically or manually, depending upon the hot spare policy, replace a failing or failed disk in a RAID configuration.
RHEL	Red Hat Enterprise Linux*
Intel® RSTe	Intel® Rapid Storage Technology enterprise.
RWH	It stands for RAID 5 Write Hole and can cause data integrity issue.
SLES	SUSE Linux* Enterprise Server
TB	Terabyte
TiB	Tebibyte (1024 x 1024 x 1024 x 1024 bytes)
UEFI Mode	Unified Extensible Firmware Interface. Refers to the system setting in the BIOS.
Volume	This term is representative of *mdadm* RAID within an Intel® metadata-based container.
Volume initialization	Immediately after a RAID volume has been created, initialization (or resync) commences if the RAID level is 1, 10, or 5 to guarantee volume data integrity.
Intel® VROC	Intel® Virtual RAID on CPU

1.3 Reference OEM Platform Documentation

Refer to your OEM for a full list of available feature sets. If any of the information in this document conflicts with the support information provided by the platform OEM, the platform documentation and configurations take precedence.

Customers should always contact the place of purchase or system/software manufacturer with support questions about their specific hardware or software configuration.

1.4 Supported Linux* OS Distributions

The Intel® VROC Linux* driver and tools are open sourced. Linux* OSVs are pulling in Intel® VROC features and patches from upstream Linux* community and integrate into their Linux* OS distributions. For most of Linux* OS distributions, the inbox kernel drivers and tools can fully support Intel® VROC functions well and there is no additional software download required. For some Linux* OS distributions, it requires an out-of-box Intel® VROC Linux* driver package.

The supported Linux* OS versions in each Intel® VROC 9.0 release follow the lifecycle of each Linux* distribution. Please follow the guidance from Linux* OSVs for their Linux* product's lifecycle.

1.4.1 Supported Linux* Distributions in Intel® VROC 9.0 Release

The release package for Intel® VROC Linux* 9.0 family of products was designed to work with, tested and validated on the following Linux* distributions:

- Red Hat Enterprise Linux* Server:
  - 8.8
  - 8.10
  - 9.2
  - 9.4
- SUSE Linux* Enterprise Server:
  - 15 SP5
  - 15 SP6[1]
- Ubuntu* Server:
  - 24.04 LTS[1]

Note: [1]There are no additional out-of-box driver packages from Intel. The OS inbox has already been validated and full functional for Intel® VROC. Always reference the latest Intel® Virtual RAID on CPU (Intel® VROC) Supported Configurations for details.

1.4.2 Supported Linux* Distributions in Intel® VROC 8.6 Release

The release package for Intel® VROC Linux* 8.6 family of products was designed to work with, tested and validated on the following Linux* distributions:

- Red Hat Enterprise Linux* Server:
  - RHEL* 8.9[1]
  - RHEL* 9.3[1]
- Ubuntu* Server:
  - 22.04.3 LTS[1]

Note: [1]There are no additional out-of-box driver packages from Intel. The OS inbox has already been validated and full functional for Intel® VROC. Always reference the latest Intel® Virtual RAID on CPU (Intel® VROC) Supported Configurations for details.

1.4.3 Supported Linux* Distributions in Intel® VROC 8.2 Release

The release package for Intel® VROC Linux* 8.2 family of products was designed to work with, tested and validated on the following Linux* distributions:

- Red Hat Enterprise Linux* Server:
  - RHEL* 8.7[1]
  - RHEL* 8.8[1]
  - RHEL* 9.1[1]
  - RHEL* 9.2[1]
- SUSE Linux* Enterprise Server:
  - SLES15 SP5[1]
- Ubuntu* Server:
  - 20.04.3 LTS

Note: [1]There are no additional out-of-box driver packages from Intel. The OS inbox has already been validated and full functional for Intel® VROC. Always reference the latest Intel® Virtual RAID on CPU (Intel® VROC) Supported Configurations for details.

1.4.4 Supported Linux* Distributions in Intel® VROC 8.0 Release

The release package for Intel® VROC Linux* 8.0 family of products was designed to work with, tested and validated on the following Linux* distributions:

- Red Hat Enterprise Linux* Server:
  - RHEL* 8.2
  - RHEL* 8.3
  - RHEL* 8.4
  - RHEL* 8.5
  - RHEL* 8.6[1]
  - RHEL* 9.0[1]
- SUSE Linux* Enterprise Server:
  - SLES15 SP2
  - SLES15 SP3
  - SLES15 SP4[1]

Note: [1]There are no additional out-of-box driver packages from Intel. The OS inbox has already been validated and full functional for Intel® VROC. Always reference the latest Intel® Virtual RAID on CPU (Intel® VROC) Supported Configurations for details.

1.5 Supported PCIe NVMe SSDs

Intel® VROC Linux* supports most shipping enterprise and datacenter NVMe SSDs. For the latest list of supported PCIe NVMe SSDs, refer to the Intel® Virtual RAID on CPU (Intel® VROC) Supported Configurations.

Platform providers are now allowed to self-validate their own list of NVMe SSDs for use with Intel® VROC (VMD NVMe RAID). For more details, contact your platform provider.

1.6 Intel® SSD Only VROC Upgrade Key

Intel SSD Only keys will no longer be supported on new platforms starting with 4th Gen Intel® Xeon® Scalable Processors based platforms. This feature has not been disabled due to legacy platform support. Any issues against 4th Gen Intel® Xeon® Scalable Processors based platforms using *Intel® SSD Only* keys will not be supported.

§§

2 New Features in Intel® VROC 9.0 Release

The Intel® VROC Linux* 9.0 release package introduces several new features to support the Intel® Xeon® 6 Processor Family as well as improve the user experience. The following sections list the key features introduced in Intel® VROC Linux* 9.0.

2.1 Support of Intel® VMD 4.0

Intel® VMD is a hardware logic inside Intel® Xeon® Processor Family. The Intel® Xeon® 6 Processor Family introduces Intel® VMD 4.0, which supports the following key features:

- Up to 128 MSI-X vectors.
- Support for inband PECI configuration access to PCIe SSDs.
- MCTP message routing to/from Intel® VMD-owned devices.
- Up to 6 VMD devices per socket.
- Intel® VROC Out-of-Band Management.

The Intel® VROC Linux* 9.0 release package includes all the Linux* drivers and tools to enable those features in Linux* environment. For details of Intel® VMD 4.0, reference the External Design Specification of the Intel® Xeon® 6 Processors.

§§

3 New Features in Intel® VROC 8.0 Release

The Intel® VROC Linux* 8.0 release package introduces several new features to support the 4th Gen Intel® Xeon® Scalable Processor Platforms as well as improve the user experience. The following sections list the key features introduced in Intel® VROC Linux* 8.0.

3.1 Support of Intel® VMD 3.0

Intel® VMD is a hardware logic inside Intel® Xeon® Scalable Processor. The 4th Gen Intel® Xeon® Scalable Processors introduce Intel® VMD 3.0, which supports the following key features:

- Up to 64 MSI-X vectors.
- Increase in the number of PCI Express lanes that can be controlled by Intel® VMD from 64 to 80.
- Increase in the number of Intel® VMD devices from 5 to 6.
- Intel® VMD Hot Plug support of NVMe devices attached to the Platform Controller Hub (PCH).
- Intel® VROC Out-of-Band Management.

The Intel® VROC Linux* 8.0 release package includes all the Linux* drivers and tools to enable those features in Linux* environment. For details of Intel® VMD 3.0, reference the External Design Specification of the 4th Gen Intel® Xeon® Scalable Processors.

3.2 Support of 3rd SATA Controller

The Intel® VROC Linux* 8.0 release introduces the support for the 3rd SATA controller introduced on Intel® C741 chipset on the 4th Gen Intel® Xeon® platforms. The Intel® C741 chipset has three integrated SATA host controllers that support up to twenty ports with data transfer rates of up to 6 Gb/s on all ports. The Intel® VROC Linux* 8.0 release can support all the three SATA controllers on the platform.

3.3 Support of Out-of-Band Management

The Intel® VROC Linux* 8.0 release introduces the support for Out-of-Band Management on platforms based on the Intel® Eagle Stream Customer Reference Board design. The Intel® VROC Linux* 8.0 release includes the driver packages that enable the Out-of-Band Management interface which supports NVMe drives and Intel® VROC (VMD NVMe RAID) volumes only. For details, reference the Intel® VROC Linux* Technical Product Specification.

3.4 Support of PCIe* Hot Plug on PCH

The Intel® VROC Linux* 8.0 release introduces the support of the hot plug of NVMe SSDs that are directly attached to the Intel® PCH PCIe lanes managed by Intel® VMD.

3.5 Support of Write-Intent Bitmap

The Intel® VROC Linux* 8.0 release introduces the support of the write-intent bitmap function. A write-intent bitmap can reduce the RAID rebuild/resync time especially for write-intensive workloads when a dirty shutdown or an unexpected system crash occurs. Write-intent bitmap support is only available for RAID levels with data redundancy. For details of how to use write-intent bitmap with Intel® VROC Linux* 8.0 release, refer the Intel® Virtual RAID on CPU (Intel® VROC) User Guide for Linux*.

§§

4 Intel® VROC Linux* Release Package

4.1 Release Package Contents in Intel® VROC 9.0 Release

The Intel® VROC Linux* 9.0 release package contains the Intel® VROC out-of-box drivers and tools for the supported Red Hat and SUSE Linux* distributions.

4.1.1 Release Packages for Red Hat Enterprise Linux*

As for the supported RHEL Linux* distributions, a **dd.iso** archive file, which contains rpm packages as well as source code packages, is provided for each supported OS version separately. The **dd.iso** can be loaded during the OS installation phase or installed in the Linux* environment. Refer to Chapter 5. Installation of Intel® VROC Linux* for detailed steps of installing the Intel® VROC Linux* out-of-box drivers.

The following is an example of the detailed contents of the **dd.iso** for RHEL* 9.2. After mounting the **dd.iso** to the designated directory in Linux*, the user will find the following three RPM packages in the **rpms/x86_64/** directory. The **kmod-iavmd-<version>.rpm** is the Intel® VMD kernel replacement driver for RHEL* 9.2, **ledmon-<version>.rpm** is an updated *ledmon* utility and **mdadm-<version>.rpm** is an updated *mdadm* utility for RHEL* 9.2. All of them can be installed through the Linux* command **rpm -Uvh**.

- *ledmon-0.97-20231025.intel.13002832.el9.x86_64.rpm*
- *mdadm-4.3-20240220.intel.13598375.el9.x86_64.rpm*

The corresponding source code rpm packages are stored in the **src/** directory:

- *ledmon-0.97-20231025.intel.13002832.el9.src.rpm*
- *mdadm-4.3-20240220.intel.13598375.el9.src.rpm*

```
├── rhdd3
├── rpms
│   └── x86_64
│       ├── ledmon-0.97-20231025.intel.13002832.el9.x86_64.rpm
│       ├── mdadm-4.3-20240220.intel.13598375.el9.x86_64.rpm
│       └── repodata
│           ├── 3d8434ae8e66acac4c38db1cdb258f0df613eaf29d42cacffecf543233dbbb0c-
filelists.xml.gz
│           ├── 66e7f7ddff1048c24675837357089f94586296909b19f2443fa3c2dda88c070d-
other.xml.gz
│           ├── 7028ec63abd504a395d70fc98ebe01d7707c8850920908091a84338275f7cd27-
other.sqlite.bz2
│           ├── bc2d31c89a32778862a89d1a4e483c4899f4544610cc9c68fc427e986f5c5f0c-
primary.sqlite.bz2
│           ├── bec0b577d81959802d33e8191717c3eab0b0ee13ed183bc90a7ef095ed1aaad4-
primary.xml.gz
│           ├── f37d911e26da2f3d3b04a2cd38296dbbf4bff8fe424a5be5c50f42e8fe71f50c-
filelists.sqlite.bz2
```

```
│           └── repomd.xml
└── src
    ├── ledmon-0.97-20231025.intel.13002832.el9.src.rpm
    └── mdadm-4.3-20240220.intel.13598375.el9.src.rpm
```

4.1.2 Release Packages for SUSE Linux* Enterprise Server

As for the supported SUSE Linux* distributions, two RPM packages are provided for each supported OS version.

The following is an example of the detailed contents of Intel® VROC Linux* 9.0 out-of-box packages for SLES 15 SP5. **`ledmon-<version>.rpm`** and **`mdadm-<version>.rpm`** are two RPM files for the updated *ledmon* and *mdadm* utilities. Both of them can be installed through the Linux* command **`rpm -Uvh`**.

- *ledmon-0.97-20231220.intel.13307073.sle15SP5.x86_64.rpm*
- *ledmon-0.97-20231220.intel.13307073.sle15SP5.x86_64.rpm*

The corresponding source code rpm packages can be found in the **src/** directory:

- *ledmon-0.97-20231220.intel.13307073.sle15SP5.src.rpm*
- *mdadm-4.3-20240220.intel.13598374.sle15SP5.src.rpm*

4.2 Release Package Contents in Intel® VROC 8.6 Release

There is no out-of-box driver package specifically for Intel® VROC 8.6 release. All the supported Linux* distributions have fully inbox support of Intel® VROC.

4.3 Release Package Contents in Intel® VROC 8.2 Release

The Intel® VROC Linux* 8.2 release package contains the Intel® VROC out-of-box drivers and tools for Ubuntu* Server 20.04.3 LTS.

4.3.1 Release Packages for Ubuntu* Server 20.04.3 LTS

The Intel® VROC Linux* 8.2 release package for Ubuntu* Server 20.04.3 LTS contains the following two deb files and two source code tarballs in the **`src/`** directory.

```
├── mdadm_4.2-intel.1.11978297.ub20_amd64.deb
├── ledmon_0.95-intel.1.11992759.ub20_amd64.deb
└── src
    ├── mdadm_4.2.src.tar.gz
    ├── ledmon_0.95.src.tar.gz
```

4.4 Release Package Contents in Intel® VROC 8.0 Release

The Intel® VROC Linux* 8.0 release package contains the Intel® VROC out-of-box drivers and tools for the supported Red Hat and SUSE Linux* distributions.

4.4.1 Release Packages for Red Hat Enterprise Linux*

As for the supported RHEL Linux* distributions, a **dd.iso** archive file, which contains rpm packages as well as source code packages, is provided for each supported OS version separately. The **dd.iso** can be loaded during the OS installation phase or installed in the Linux* environment. Refer to Chapter 5. Installation of Intel® VROC Linux* for detailed steps of installing the Intel® VROC Linux* 8.0 out-of-box drivers.

The following is an example of the detailed contents of the **dd.iso** for RHEL* 8.2. After mounting the **dd.iso** to the designated directory in Linux*, the user will find the following three RPM packages in the **rpms/x86_64/** directory. The **kmod-iavmd-<version>.rpm** is the Intel® VMD kernel replacement driver for RHEL* 8.2, **ledmon-<version>.rpm** is an updated *ledmon* utility and **mdadm-<version>.rpm** is an updated *mdadm* utility for RHEL* 8.2. All of them can be installed through the Linux* command **rpm -Uvh**.

- *kmod-iavmd-1.0.0.1569-rhel_82.x86_64.rpm*
- *ledmon-0.95-1.intel.7468292.el8.x86_64.rpm*
- *mdadm-4.2-1.intel.9009306.el8.x86_64.rpm*

The corresponding source code rpm packages are stored in the **src/** directory:

- *iavmd-1.0.0.1569-rhel_82.src.rpm*
- *ledmon-0.95-1.intel.7468292.el8.src.rpm*
- *mdadm-4.2-1.intel.9009306.el8.src.rpm*

```
├── rhdd3
├── rpms
│   └── x86_64
│       ├── kmod-iavmd-1.0.0.1569-rhel_82.x86_64.rpm
│       ├── ledmon-0.95-1.intel.7468292.el8.x86_64.rpm
│       ├── mdadm-4.2-1.intel.9009306.el8.x86_64.rpm
│       └── repodata
│           ├── 22cf7d53426078db1d7cbf38aa1879cac3edc5fc7f0d6f4e8954a837a50d57e8-
primary.xml.gz
│           ├── 3e9b679754a7f20f44d2cbbebee9bc7af6bb9bb2f2cdf23de5f3926495d9da21-
other.sqlite.bz2
│           ├── 77c92a0cd607471c51ef5e1f6f87338a4bc83ec0a883063b124acf0cd385d5e0-
primary.sqlite.bz2
│           ├── bda60334ca49c9c1e2e4a82b5b28b8c6de275aac8ccd24111b3ca7c14045b0d8-
other.xml.gz
│           ├── d7cc3a4397a00a9dcb5917a3d1820c0f9822683db8b01ad18160fa470fd3a730-
filelists.sqlite.bz2
│           ├── e0903f2b57b0ff315861c2180e8d0b3e684e72f1b67e770126d9021399028c70-
filelists.xml.gz
│           └── repomd.xml
└── src
    ├── iavmd-1.0.0.1569-rhel_82.src.rpm
    ├── ledmon-0.95-1.intel.7468292.el8.src.rpm
```

```
└── mdadm-4.2-1.intel.9009306.el8.src.rpm
```

4.4.2 Release Packages for SUSE Linux* Enterprise Server

As for the supported SUSE Linux* distributions, a Driver Update Disk (DUD) ISO file and two RPM packages are provided for each supported OS version.

The following is an example of the detailed contents of Intel® VROC Linux* 8.0 out-of-box packages for SLES 15 SP2. The **iavmd-<version>-dud.iso** file is a DUD file of the Intel® VMD kernel replacement driver for SLES 15 SP2. This DUD driver can be loaded and installed during the OS installation phase. **ledmon-<version>.rpm** and **mdadm-<version>.rpm** are two RPM files for the updated *ledmon* and *mdadm* utilities. All of them can be installed through the Linux* command **rpm -Uvh**.

- *iavmd-1.0.0.1602-sles15_sp2-dud.iso*
- *ledmon-0.95-1.intel.7484205.sle15.x86_64.rpm*
- *mdadm-4.2-1.intel.9009857.sle15SP2.x86_64.rpm*

The corresponding source code rpm packages are stored in the **src/** directory:

- *ledmon-0.95-1.intel.7484205.sle15.src.rpm*
- *mdadm-4.2-1.intel.9009857.sle15SP2.src.rpm*

4.5 Release Packages for Out-of-Band Management in Linux*

4.5.1 Release Packages in Intel® VROC 8.2 Release

The Intel® VROC Linux* 8.2 release for Out-of-Band (OOB) Management contains the updated OOB driver packages for all the Intel® VROC supported Linux* distributions for the 4th Generation Intel® Xeon® Scalable Processor based platforms.

The packages for Out-of-Band Management in Linux* are stored in a zip file named **OOB_8.2.0.1031.zip**. Two RPM files are provided for each supported Red Hat and SUSE Linux* distribution. One DEB file and one tarball file are provided for the supported Ubuntu* Server 20.04.3 LTS.

The following is an example of the two RPM files for Intel® VROC Out-of-Band Management in RHEL 8.x Linux* distributions. The **oob-0.1-OOB_<version>.x86_64.rpm** is the RPM for OOB daemon program running in the host Linux* environment. The **oob-0.1-OOB_<version>.src.rpm** is the source code of the OOB daemon program.

- *oob-0.1-OOB_8.2.0.1031.el8.x86_64.rpm*
- *oob-0.1-OOB_8.2.0.1031.el8.src.rpm*

Note: Intel® VROC Out-of-Band Management should be implemented and used together with OEM platforms' hardware/firmware design and implementation.

4.5.2 Release Packages in Intel® VROC 8.0 Release

The Intel® VROC Linux* 8.0 release introduces the support for Out-of-Band (OOB) Management on platforms based on the Intel® Eagle Stream Customer Reference Board design. The Intel® VROC Linux* 8.0 release includes the driver packages that enable the Out-of-Band Management interface which supports NVMe drives and Intel® VROC (VMD NVMe RAID) volumes in the Linux* environment.

The packages for Out-of-Band Management in Linux* are stored in a separated folder named *Out-of-Band (OOB)_Release_for_Linux*. Two RPM files are provided for each supported Linux* distribution.

The following is an example of the two RPM files for Intel® VROC Out-of-Band Management in RHEL* 8 Linux* distributions. The **`oob-0.1-OOB_<version>.x86_64.rpm`** is the RPM for OOB daemon program running in the host Linux* environment. The **`oob-0.1-OOB_<version>.src.rpm`** is the source code of the OOB daemon program.

- *oob-0.1-OOB_1.0.0.1298.el8.x86_64.rpm*
- *oob-0.1-OOB_1.0.0.1298.el8.src.rpm*

Note: Intel® VROC Out-of-Band Management should be implemented and used together with OEM platforms' hardware/firmware design and implementation.

§§

5 Installation of Intel® VROC Linux*

This chapter outlines installing of Intel® VROC Linux* out-of-box package as well as configuring the inbox Intel® VROC components in the Intel® VROC supported Linux* distributions.

5.1 Installation of Intel® VROC Linux* Update Packages

For some certain Linux* distributions, Intel may release out-of-box update packages to support a fully functional Intel® VROC product. This section illustrates how to install the Intel® VROC Linux* out-of-box packages in Red Hat Enterprise Linux* Distributions.

5.1.1 Installing Intel® VMD Replacement Driver During OS Installation

The Intel® VMD replacement driver is included in the Intel® VROC out-of-box package. This replacement driver is used to replace the OS kernel inbox VMD driver to enable or improve Intel® VROC functionalities on certain Intel® Xeon® platforms.

The Intel® VMD replacement driver can be installed during the OS installation phase or in the OS environment. The following steps illustrate how to install the Intel® VMD replacement driver during OS installation.

1. Prepare USB drives with the RHEL* installation ISO and the Intel® VROC out-of-box driver ISO package.

   Use the **dd** command to create a USB drive with the RHEL* ISO installation:

   ```
   # dd if=/path/to/<RHEL_OS>.iso of=/dev/sdX status=progress
   ```

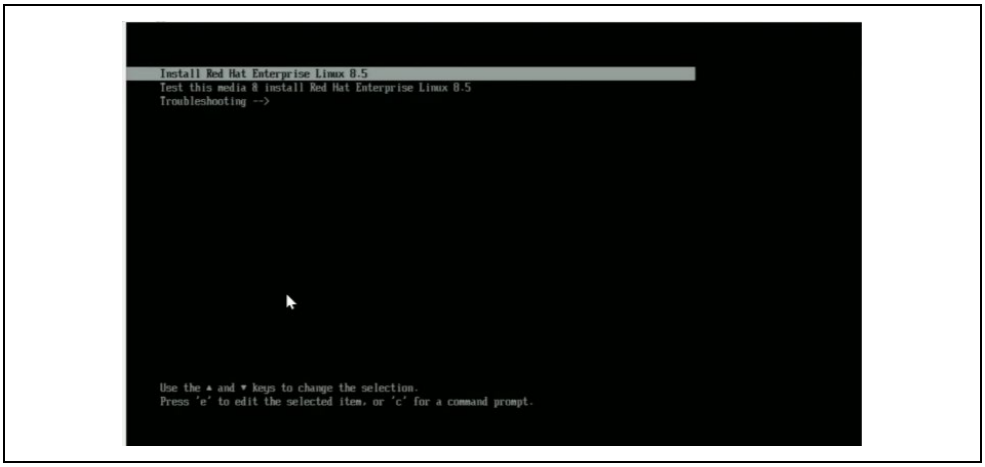
   Use the **dd** command to create a USB drive with the Intel® VROC out-of-box ISO package:

   ```
   # dd if=/path/to/<vroc_update_driver>.iso of=/dev/sdX status=progress
   ```

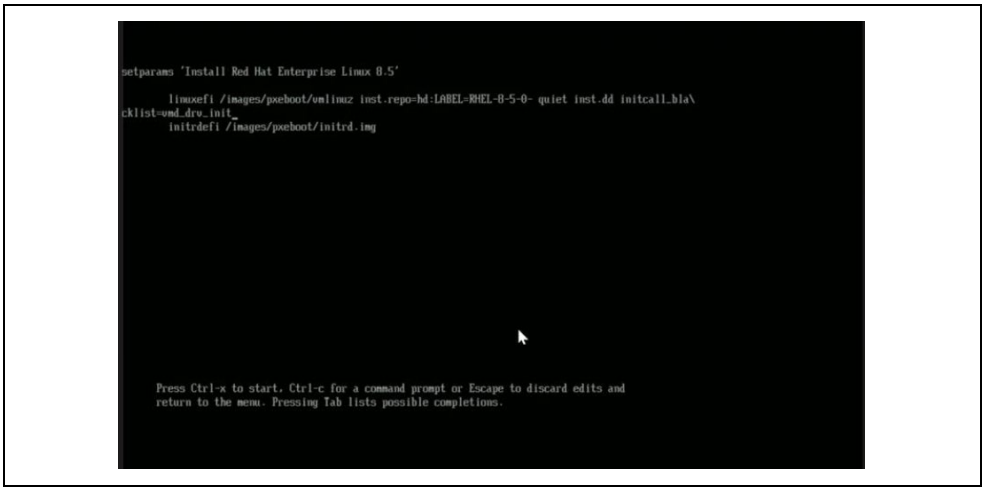
   After the USB drive with the Intel® VROC driver is prepared, make sure it has the correct label name *OEMDRV*:

   ```
   # lsblk -o name,label
       NAME             LABEL
       sdb              OEMDRV
   ```

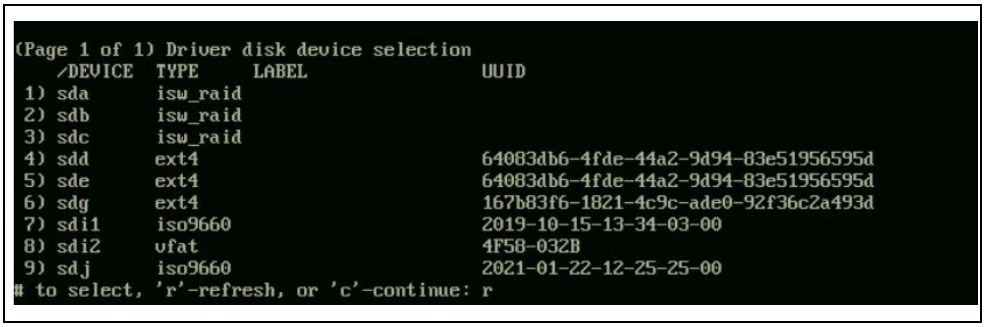
2. Boot your system with the bootable media created in the above step. When the system boots, we will get the below installation screen. Select the option **Install Red Hat Enterprise Linux 8.5** and press <E>.



3. Add the line **`inst.dd initcall_blacklist=vmd_drv_init`** in the end of the kernel boot cmdline. Press Ctrl + X to continue with the installation process.



4. Select the driver with label *OEMRDV* in this example. If you don't see the USB device which is named as *E*, input <r> to reflash again.



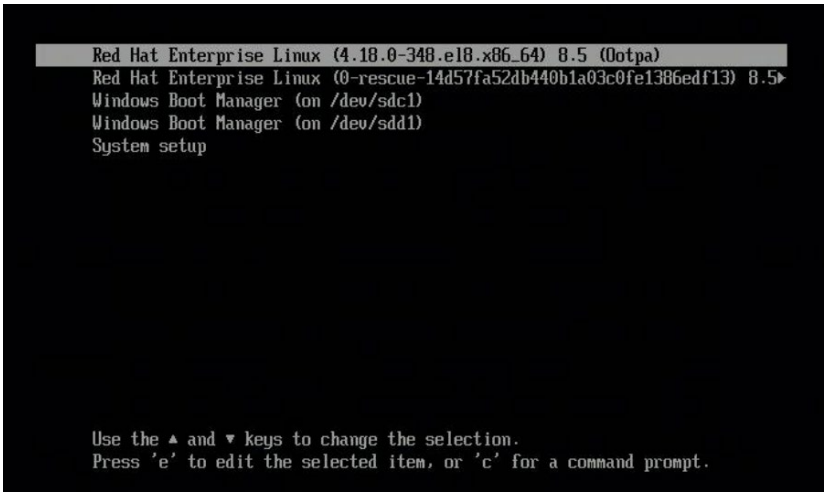
After refresh, select device with label *E*.

5. Toggle the **kmod-iavmd** package, then input <c> to continue to the main menu.

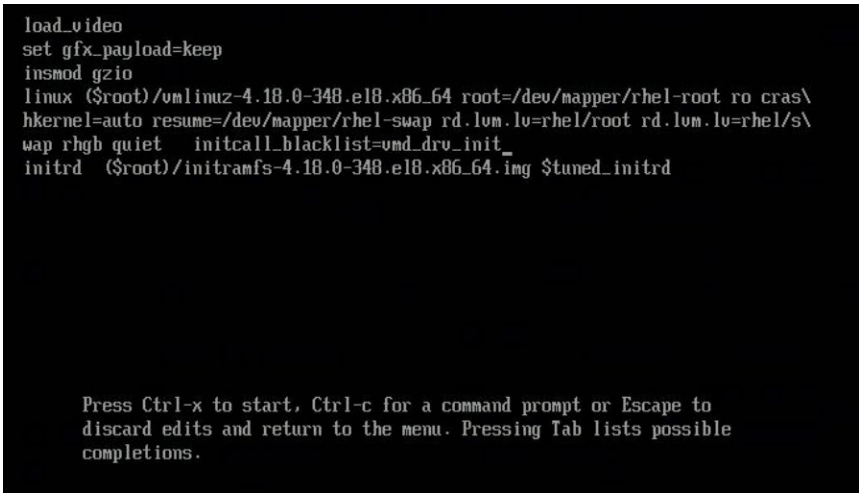
6. At the main menu, type <c> to continue with the installation.

7. Follow the normal RHEL* installer guidance to complete the OS installation.
8. A reboot is required after the installation. Add the line **initcall_blacklist=vmd_drv_init** in the boot cmdline before directly booting into the OS. This boot parameter is used to tell the OS to blacklist the inbox VMD driver and load the Intel® VMD replacement driver.

   To accomplish this, select the **Red Hat Enterprise Linux*** option and press <E>.



Then add the line **initcall_blacklist=vmd_drv_init** in the Linux* boot cmdline, and then press Crtl + X to start.



5.1.2 Installing Intel® VROC Out-of-Box Drivers in Red Hat Linux* OS

Section 4.1.1 illustrates the steps to install the Intel® VMD replacement driver during OS installation. This section will cover the steps to install the Intel® VMD replacement driver as well as other Intel® VROC update packages in Linux* OS.

1. Copy the Intel® VROC update package to the target Linux* OS and mount the iso to a certain directory. Enter that directory of **rpms/x86_64** and you will typically find three rpm packages as showed in the below example:

```
# ls -l rpms/x86_64/
total 828
-r--r--r--. 1 root root 328776 Apr  4 07:23 kmod-iavmd-1.0.0.1600-
rhel_85.x86_64.rpm
```

```
-r--r--r--. 1 root root  83860 Apr  4 07:23 ledmon-0.95-
1.Intel.7468292.el8.x86_64.rpm
-r--r--r--. 1 root root 431924 Apr  4 07:23 mdadm-4.2-
1.Intel.9009306.el8.x86_64.rpm
dr-xr-xr-x. 2 root root   2048 Apr  4 07:23 repodata
```

2. Use the command **`rpm -Uvh --force`** to install all those packages. See the examples below:

   This is an example to install the Intel® VMD replacement driver package prefixed by **`kmod-iavmd-`**:

```
# rpm -Uvh --force kmod-iavmd-1.0.0.1600-rhel_85.x86_64.rpm
warning: kmod-iavmd-1.0.0.1600-rhel_85.x86_64.rpm: Header V4 RSA/SHA256 Signature,
key ID c343c1b0: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:kmod-iavmd-1.0.0.1600-rhel_85    ################################# [100%]
```

   Check the rpm installation status by running the following command:

```
# rpm -qa | grep kmod-iavmd
kmod-iavmd-1.0.0.1600-rhel_85.x86_64
```

   This is an example to install Intel® VROC Linux* update package for *mdadm* utility:

```
# rpm -Uvh --force mdadm-4.2-1.Intel.9009306.el8.x86_64.rpm
warning: mdadm-4.2-1.Intel.9009306.el8.x86_64.rpm: Header V4 RSA/SHA256 Signature,
key ID b7accecb: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:mdadm-4.2-1.Intel.9009306.el8    ################################# [100%]
```

   Check the rpm installation status by running the following command:

```
# mdadm --version
mdadm – v4.2 – 2021-12-30 – Intel_Build: 1.Intel.9009306.el8
```

   This is an example to install Intel® VROC Linux* update package for *ledmon* utility:

```
# rpm -Uvh --force ledmon-0.95-1.Intel.7468292.el8.x86_64.rpm
warning: ledmon-0.95-1.Intel.7468292.el8.x86_64.rpm: Header V4 RSA/SHA256
Signature, key ID b7accecb: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:ledmon-0.95-1.Intel.7468292.el8  ################################# [100%]
```

   Check the rpm installation status by following command:

```
# ledmon --version
Intel(R) Enclosure LED Monitor Service 0.95
Copyright (C) 2009-2021 Intel Corporation.

This is free software; see the source for copying conditions. There is NO
warrantry;
not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

ledmon[41052] : exit status is STATUS_SUCCESS.
```

3. Update the GRUB configuration file to permanently add the following kernel boot parameter to allow the OS to automatically load the Intel® VMD replacement driver instead of the inbox one:

```
initcall_blacklist=vmd_drv_init
```

   To achieve that, edit the **`/etc/default/grub`** file, and add the line **`initcall_blacklist=vmd_drv_init`** to the end of the **`GRUB_CMDLINE_LINUX`** line. Take the following as an example:

```
# vim /etc/default/grub
```

```
GRUB_CMDLINE_LINUX=”crashkernel=auto resume=/dev/mapper/rhel-swap
rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet
initcall_blacklist=vmd_drv_init”
```

After editing the above **/etc/default/grub** file, regenerate the GRUB configuration file by running the following command:

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

5.1.3 Installing Intel® VROC Out-of-Box Drivers in Ubuntu* Server 20.04.3 LTS

dpkg is the package management tool in Ubuntu. Use the following command to install the Intel® VROC Linux* 8.2 release out-of-box drivers for Ubuntu* Server 20.04.3 LTS.

1. Install the *mdadm* update package.

```
# dpkg -i mdadm_4.2-intel.1.11978297.ub20_amd64.deb
Selecting previously unselected package mdadm.
(Reading database ... 109196 files and directories currently installed.)
Preparing to unpack mdadm_4.2-intel.1.11978297.ub20_amd64.deb ...
Unpacking mdadm (4.2-intel.1.11978297.ub20) ...
Setting up mdadm (4.2-intel.1.11978297.ub20) ...
Generating mdadm.conf... done.
update-initramfs: deferring update (trigger activated)
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69464:
grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69464:
grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69519:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69519:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69534:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69534:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69549:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69549:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69564:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69564:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69636:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69636:
/usr/sbin/grub-probe
Found linux image: /boot/vmlinuz-5.4.0-156-generic
Found initrd image: /boot/initrd.img-5.4.0-156-generic
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69721:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69721:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69737:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69737:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69752:
/usr/sbin/grub-probe
```

```
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69752:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69767:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69767:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69939:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on vgs invocation. Parent PID 69939:
/usr/sbin/grub-probe
File descriptor 3 (pipe:[180652]) leaked on lvs invocation. Parent PID 70162:
/bin/sh
Adding boot menu entry for UEFI Firmware Settings
done
Created symlink /etc/systemd/system/mdmonitor.service.wants/mdcheck_continue.timer
→ /lib/systemd/system/mdcheck_continue.timer.
Created symlink /etc/systemd/system/mdmonitor.service.wants/mdcheck_start.timer →
/lib/systemd/system/mdcheck_start.timer.
Created symlink /etc/systemd/system/mdmonitor.service.wants/mdmonitor-
oneshot.timer → /lib/systemd/system/mdmonitor-oneshot.timer.
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.6) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-156-generic
W: Possible missing firmware /lib/firmware/ast_dp501_fw.bin for module ast
```

2. Install the *ledmon* update package.

```
# dpkg -i ledmon_0.95-intel.1.11992759.ub20_amd64.deb
Selecting previously unselected package ledmon.
(Reading database ... 109185 files and directories currently installed.)
Preparing to unpack ledmon_0.95-intel.1.11992759.ub20_amd64.deb ...
Unpacking ledmon (0.95-intel.1.11992759.ub20) ...
Setting up ledmon (0.95-intel.1.11992759.ub20) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ledmon.service →
/lib/systemd/system/ledmon.service.
Processing triggers for systemd (245.4-4ubuntu3.20) ...
Processing triggers for man-db (2.9.1-1) ...
```

5.2 Configuring Intel® VROC in Linux* Distributions with Inbox Support

5.2.1 Configuring Intel® VROC in Red Hat Enterprise Linux*

5.2.1.1 Enabling *ledmon* Service

By default, the *ledmon* service is disabled in Red Hat Enterprise Linux* distribution. The user should enable it and get it started by running the following command:

```
# systemctl enable --now ledmon.service
```

Check the *ledmon* service status by running the following command:

```
# systemctl status ledmon.service
```

5.2.2 Configuring Intel® VROC in SUSE Linux* Enterprise Server (SLES) 15

5.2.2.1 Installing SLES 15 SP2 onto Intel® VROC RAID

The SLES 15 SP2 installer by default is not able to assemble a RAID volume created in the PreOS/UEFI. When installing SLES 15 SP2 onto an Intel® VROC RAID volume, a boot command line **`autoassembly=1`** should be added to enable RAID assembly in the installer.

5.2.2.2 Installing *ledmon* in SLES Family

SLES 15 family Linux* distributions don't include the *ledmon* package in the base OS installation. The user should install *ledmon* manually after the OS installation. Use the following command to install *ledmon*:

```
# zypper install ledmon
```

5.2.2.3 Enabling *ledmon* Service

By default, the *ledmon* service is disabled in SLES Linux* distributions. The user should enable it and get it started by running the following command:

```
# systemctl enable --now ledmon.service
```

Check the *ledmon* service status by running the following command:

```
# systemctl status ledmon.service
```

§§

6 Intel® VROC Linux* Limitations

6.1 Missing of 3rd SATA Controller Device ID

The Intel® VROC Linux* 8.0 release introduces the support for the 3rd SATA controller introduced on the Intel® C741 chipset on the 4th Gen Intel® Xeon® Scalable Processor Platforms. In some of the Linux* distributions the Intel® VROC Linux* 8.x release supports, the device ID of the 3rd SATA controller is by default not included in the inbox Linux* AHCI driver. As a result, the 3rd SATA controller is not functional in those Linux* distributions. The following table lists the Linux* distributions with and without 3rd SATA controller support by default.

As a workaround, the following command can be used to manually add the device ID of the 3rd SATA controller to those Linux* distributions without the default support:

```
"echo "8086 282f" > /sys/bus/pci/drivers/ahci/new_id"
```

Table 5-1. 3rd SATA Controller Dev ID support

Linux* Distribution	3rd SATA Controller Dev ID in the Inbox Linux* AHCI Driver
RHEL* 8.2	No
RHEL* 8.3	No
RHEL* 8.4	No
RHEL* 8.5	No
RHEL* 8.6	Yes
RHEL* 9.0	Yes
SLES 15 SP2	No
SLES 15 SP3	Yes
SLES 15 SP4	Yes
Ubuntu* 20.04.3	No
Ubuntu* 22.04.3	No

6.2 Notice of Using Intel® VROC in RHEL* 9.0

The Red Hat Enterprise Linux* 9.0 GA release has the inbox support of Intel® VROC Linux* 8.0. Some known issues were reported in the RHEL* 9.0 GA release and resolved in the latest installation image and Red Hat errata. In order to successfully use Intel® VROC in RHEL* 9.0, the following actions should be taken:

1. Download the latest RHEL* 9.0 installation image from the Red Hat Customer Portal. The latest RHEL* 9.0 ISO image files were last modified on January 30th, 2023.
2. Apply an update for *kernel* from RHSA-2023:4137 - Security Advisory.
3. Apply an update for *systemd* from RHBA-2022:6604 - Bug Fix Advisory.
4. Reboot the system.

The following table lists the known issues found in RHEL* 9.0 and resolved in the latest ISO installation image and Red Hat product errata.

Table 5-2. Known Issues Resolved in RHEL* 9.0 with Errata

Title	Internal/External Reference ID	Resolution
RHEL* 9.0 doesn't boot when VMD and interrupt remapping are enabled	22015103377 / 2084146	Resolved in the updated RHEL* 9.0 ISO installation image
RHEL* 9.0 doesn't support 3rd SATA Controller	22015103390 / 2099740	Resolved in the updated RHEL* 9.0 ISO installation image
Reboot hangs after RHEL* 9.0 installing on Intel® VROC RAID	22015093219 / 2047682	Resolved in RHBA-2022:6604 errata
Auto rebuild issue during fio write	22015156993 / 2092914	Resolved in RHSA-2023:4137 errata

6.3 Linux* OS May Not Properly Boot/Install from A Degraded RAID Volume

This is a known limitation in current Intel® VROC Linux* design. If a RAID member drive is failed/removed during Linux* OS booting phase, the RAID volume is not able to be assembled properly. As a result, Linux* booting or installation onto that RAID volume may fail. Rebooting the system will resolve this issue.

§§

7 Known Issues in Intel® VROC Linux* 9.0

This chapter outlines the issues reported and internally found in the Intel® VROC Linux* 9.0 release.

Table 7-1. Fail to change LED state in the guest OS

Title	Fail to change LED state in the guest OS
Reference ID	18038276913
Version	RHEL* 9.2
Issue Description	When running RHEL* 9.2 in the virtual machine as the guest OS, with Intel VMD direct assign to VM, all the NVMe devices behind Intel VMD can be directly controlled in the guest OS. Trying to use ledctl locate the NVMe drive in the guest OS, the actual LED state is not changed accordingly.
Workaround	Not at this time. Request Intel VMD replacement driver to fix it.

Table 7-2. Fail to hot remove the NVMe device in the guest OS

Title	Fail to hot remove the NVMe device in the guest OS
Reference ID	18038277012
Version	RHEL* 9.2
Issue Description	When running RHEL* 9.2 in the virtual machine as the guest OS, with Intel VMD direct assign to VM, all the NVMe devices behind Intel VMD can be directly controlled in the guest OS. Trying to hot remove the NVMe device controlled by Intel VMD in the guest OS, it will fail that the block device still remains in the system.
Workaround	Not at this time. Request Intel VMD replacement driver to fix it.

8 Known Issues in Intel® VROC Linux* 8.6

This chapter outlines the issues reported and internally found in the Intel® VROC Linux* 8.6 release.

Table 8-1. Fail to Enable/Disable PPL Consistency Policy for RAID5 Volumes

Title	Enabling or disabling PPL consistency policy using `mdadm --grow` command fails in certain Linux* OSs
Reference ID	00847088/00866586/15015036181/15015575934
Version	RHEL* 8.8/8.9, RHEL* 9.2/9.3
Issue Description	When using `mdadm --grow` command to change ppl/resync consistency policy for RAID5 volume, following error message occurs and consistency policy is not changed successfully. "mdadm: Subarray 0 in /dev/mdxxx is active, cannot update ppl"
Workaround	Fixed in upstream: "manage: adjust checking subarray state in update_subarray". RHEL* 9.4 and 8.10 will include this fix. Workaround steps: • Stop RAID volume. # mdadm --stop /dev/md/<raid_name> • Change consistency policy to resync: # mdadm --update-subarray=0 --update=no-ppl /dev/md/<container_name> • Assemble RAID volume. # mdadm -As

Table 8-2. RAID Migration from RAID1/10 to RAID0 May Fail in Certain Linux* OS Versions

Title	RAID migration from RAID1/10 to RAID0 or grow RAID0 with more devices may fail in certain Linux* OS versions
Reference ID	15014729285
Version	RHEL* 8.8/8.9, RHEL* 9.2/9.3, SLES 15 SP5
Issue Description	When performing RAID migration from RAID1 or RAID10 to RAID0, the second step of growing RAID0 raid-devices may not work with no error message.
Workaround	Reboot the OS to make RAID reshape/migration continue.

Table 8-3. Reboot May Hang in Certain Linux* OS When Being Installed onto a RAID Volume

Title	Reboot may hang when OS is installed onto a VROC RAID volume
Reference ID	16021003289 / 00810313

Version	SLES 15 SP4/SP5
Issue Description	When rebooting or shutting down the OS by reboot/shutdown command, it may hang up if OS is installed onto a VROC RAID volume.
Workaround	Follow SUSE's guide to update the systemd package: Recommended update for systemd SUSE-RU-2023:3451-1 | SUSE Support | SUSE

§§

9 Known Issues in Intel® VROC Linux* 8.2

This chapter outlines the issues reported and internally found in the Intel® VROC Linux* 8.2 release.

Table 9-1. *mdadm – Examine* Command May not Report the Correct RAID Volume Size

Title	**`mdadm --examine`** command may not reports out the exact same RAID volume size which is created and configured in the UEFI
Reference ID	18031363804
Version	Intel® VROC Linux* 8.2
Issue Description	If a RAID volume is created in the UEFI with the designated size, the volume size reported by **`mdadm --examine`** command in the Linux* OS may not be exactly the same as the size displayed in UEFI.
Workaround	Not at this time.

Table 9-2. LVM Logical Volume Creation is Failed

Title	LVM logical volume creation is failed
Reference ID	18031362930
Version	Ubuntu* 20.04.3 LTS
Issue Description	When creating LVM logical volumes on an Intel® VROC RAID volume, it may report error when doing *vgcreate* operation.
Workaround	Known LVM bug. Fixed in LVM 2.03.11 version. Update LVM with following steps: 1. Make sure *focal-backports* repository is enabled: a. sudo apt install lvm2/focal-backports

Table 9-3. Online Capacity Expansion (OCE) May Not Start on the Second RAID Volume of Intel® VROC Matrix RAID

Title	Online Capacity Expansion (OCE) may not start on the second RAID volume of Intel® VROC Matrix RAID
Reference ID	18031364673
Version	Intel® VROC Linux* 8.2
Issue Description	Adding a new spare drive to the existing Intel® VROC Matrix RAID array and execute the **`mdadm --grow`** command to reshape/resize the matrix RAID array, the OCE may not start on the second RAID volume.
Workaround	Not at this time.

Table 9-4. Spare Drive Cannot be Chosen as the Linux* OS Installation Destination

Title	Spare drive cannot be chosen as the Linux* OS installation destination
Reference ID	18032519953
Version	Ubuntu* 20.04.3 LTS
Issue Description	When choosing the Intel® VROC spare drive as the installation destination when installing Ubuntu* 20.04.3 LTS, an error may occur, and the installation is aborted.
Workaround	Do not use the spare drive as the OS disk

Table 9-5. No Error Message When Trying to Change Consistency Policy to ppl on Intel® VROC RAID 1

Title	No error message when trying to change consistency policy to ppl on Intel® VROC RAID 1
Reference ID	18031364162
Version	Ubuntu* 20.04.3 LTS
Issue Description	When trying to change the consistency policy of Intel® VROC RAID 1 to ppl (which is not supported), the *mdadm* command returns 0 without printing any error message.
Workaround	Fixed in upstream: mdadm: block update=ppl for non raid456 levels

§§

10 Known Issues in Intel® VROC Linux* 8.0

This chapter outlines the issues reported and internally found in the Intel® VROC Linux* 8.0 release.

Table 10-1. Degraded RAID Volume is Not Visible During System Boot

Title	Degraded RAID Volume is not visible during system boot
Reference ID	16014892299 / 16014866490
Version	Intel® VROC Linux* 8.0
Issue Description	If a RAID member drive is failed/removed during the Linux* OS booting phase, the RAID volume is not able to be assembled properly. As a result, Linux* booting or installation onto that RAID volume may fail.
Workaround	Reboot the system.

Table 10-2. tSATA Controller is Not Visible in Some Linux* Distributions

Title	tSATA controller is not visible in some Linux* Distributions
Reference ID	18021345328 / 18021071520 / 22014823700
Version	RHEL* 8.2/8.3/8.4/8.5, SLES 15 SP2
Issue Description	The tSATA device ID is missing in those Linux* distribution, as a result, the tSATA controller is not visible to the user.
Workaround	For each OS boot, manually add the tSATA device ID by running the following command: `# "echo "8086 282f" > /sys/bus/pci/drivers/ahci/new_id"`

Table 10-3. Online Capacity Expansion (OLCE) on OS RAID Volume May Not Work in SLES15 SP4

Title	Online Capacity Expansion (OLCE) on OS RAID volume may not work in SLES15 SP4
Reference ID	15011679625 / 22015576556
Version	SLES15 SP4
Issue Description	When the user tries to do OLCE on the Intel® VROC RAID volume where SLES15 SP4 is installed, the corresponding *mdadm* grow command will result in malfunction. This issue does not occur on data RAID volumes.
Workaround	Restart the *mdmon* service by running the following command before doing OLCE: `# systemctl restart mdmon@md*.service`

Table 10-4. It Takes Very Long Time to Create a Linux* Filesystem with TRIM Feature Enabled Onto Intel® VROC Linux* RAID 10 Volume

Title	It takes very long time to create a Linux* filesystem with TRIM feature enabled onto Intel® VROC Linux* RAID 10 volume
Reference ID	18013942972
Version	RHEL* 8.3/8.4
Issue Description	When creating a filesystem with TRIM feature enabled (e.g. mkfs.ext4) on an Intel® VROC RAID 10 volume, it will take a long time (hours) to complete.
Workaround	Disable the TRIM feature when using Intel® VROC RAID 10. The following is an example to create an EXT4 filesystem with *nodiscard* parameter: `# mkfs.ext4 -E nodiscard /dev/mdXXX`

Table 10-5. RAID Reshape Does not Continue After System Reboots

Title	RAID reshape doesn’t continue after system reboots
Reference ID	18014024616
Version	Intel® VROC Linux* 8.0
Issue Description	During the Intel® VROC RAID reshaping process, reboot the system. After system boots up, the RAID reshaping process doesn’t continue until the completion.
Workaround	Use the following command to manually start the reshape process: `# systemctl start mdadm-grow-continue@<container_md_name>.service`

Table 10-6. *FailSpare* Event is Not Logged in Syslog

Title	*FailSpare* event is not logged in syslog
Reference ID	18017126881
Version	Intel® VROC Linux* 8.0
Issue Description	When a hot spare drive is failed, the *FailSpare* event and critical information is not logged in syslog.
Workaround	Not at this time.

Table 10-7. *mdadm* is Unable to Remove the Hot-Removed Drive During RAID 10 Resyncing Sporadically

Title	*mdadm* is unable to remove the hot-removed drive during RAID 10 resyncing sporadically
Reference ID	18017145961
Version	Intel® VROC Linux* 8.0
Issue Description	While the RAID 10 resyncing is in progress, try to fail incrementally one RAID 10 member drive by using the command `mdadm If /dev/XXX`. The failed drive is not removed until resyncing is completed.

Workaround	Not needed.

Table 10-8. LVM Creation Failed on Intel® VROC RAID

Title	LVM creation failed on Intel® VROC RAID
Reference ID	18017201628 / 18014380819
Version	RHEL* 8.2 and RHEL* 8.4
Issue Description	When creating LVM on top of an Intel® VROC RAID, there will be an error because of the bug in LVM in RHEL* 8.2. For RHEL* 8.4, the issue occurs only with 4k logical block size drives. RHEL* 8.3 will not be affected.
Workaround	Not at this time.

Table 10-9. *mdadm* May Sometimes Fail to Stop RAID Volume Completely

Title	*mdadm* may sometimes fail to stop RAID volume completely
Reference ID	18017242555
Version	Intel® VROC Linux* 8.0
Issue Description	When stopping a RAID volume by running the command **`mdadm --stop`**, it may sometimes fail to stop it completely.
Workaround	Try several more times running the command **`mdadm --stop`**.

Table 10-10. *iostat* Command May Report 0 Statistics For *md* Devices

Title	*iostat* command may report 0 statistics for *md* devices
Reference ID	18017250779
Version	RHEL* 8.2/8.3/8.4/8.5
Issue Description	The *iostat* utility reports 0 statistics for *md* devices. This issue is resolved in RHEL* 8.6 and later versions.
Workaround	Not at this time.

Table 10-11. Failed Array is Not Automatically Stopped and Unmounted

Title	Failed array is not automatically stopped and unmounted
Reference ID	18017251688
Version	Intel® VROC Linux* 8.0
Issue Description	When using Intel® VROC RAID with a Linux* filesystem mounted, if the RAID volume is failed, the mounted file system will not be auto unmounted, and the RAID volume is still active.
Workaround	Not needed.

Table 10-12. Spare Drive is Recognized During SLES Installation

Title	Spare drive is recognized during SLES installation
Reference ID	22011931103/1508711532/18017426901
Version	SLES 15 SP2/SP3
Issue Description	During SLES 15 SP2/SP3/SP4 OS installation, the drive marked as spare in Intel® VROC UEFI HII is displayed when selecting the destination disk for installation.
Workaround	Do not install the OS onto the spare drive.

Table 10-13. NVMe RAID Member Device Does not Disappear from RAID Volume After Hot Remove When Native NVMe Multipathing is Enabled

Title	NVMe RAID member device doesn't disappear from RAID volume after hot remove when native NVMe multipathing is enabled
Reference ID	18017512825
Version	RHEL* 8.4
Issue Description	When native NVMe multipathing is enabled, the NVMe RAID member device is still visible in the RAID volume when it is hot removed.
Workaround	Manually remove it from the RAID volume by running the following command: `# echo "remove" > /sys/block/nvmeXn1/uevent`

Table 10-14. Rename RAID volume With Special Characters Results in An Unexpected Device Name in the */dev/md* DIRECTORY

Title	Rename RAID volume with special characters results in an unexpected device name in the */dev/md* directory
Reference ID	18018582025
Version	Intel® VROC Linux* 8.0
Issue Description	When changing or creating a RAID volume name with special characters, the device name created in the */dev/md* directory may not be the same as defined.
Workaround	Do not use special characters to name RAID volumes.

Table 10-15. Actual RAID Volume Capacity After OLCE Expansion to Maximum is 4MiB Smaller than Expected

Title	Actual RAID volume capacity after OLCE expansion to maximum is 4MiB smaller than expected
Reference ID	18019910856
Version	Intel® VROC Linux* 8.0

Issue Description	Create a RAID volume with a designated size (smaller than the maximum supported) and then increase the capacity through OLCE to maximum. The actual RAID volume capacity is 4MiB smaller than expected.
Workaround	Not at this time.

Table 10-16. Unexpected Metadata Update When RAID Level Migration From 0 To 10 is Failed

Title	Unexpected metadata update when RAID level migration from 0 to 10 is failed
Reference ID	18020767771
Version	Intel® VROC Linux* 8.0
Issue Description	When migrating a 2-disk RAID level 0 to a 4-disk RAID level 10, the *mdadm* command returns failed but the RAID metadata is updated to RAID level 10.
Workaround	Not at this time.

Table 10-17. RAID Resync Progress After System Reboots Becomes Smaller than the Number Before System Reboots

Title	RAID resync progress after system reboots becomes smaller than the number before system reboots
Reference ID	18020971695
Version	Intel® VROC Linux* 8.0
Issue Description	During RAID resync process, reboot the system and check RAID resync status. The % is less than the number before reboot. No impact on data consistency and RAID resync functionality.
Workaround	Not at this time.

Table 10-18. Failed RAID Volume is Still Available in the Intel® Matrix RAID Array

Title	Failed RAID volume is still available in the Intel® Matrix RAID array
Reference ID	22016090730 / 15012220576
Version	Intel® VROC Linux* 8.0
Issue Description	When hot remove one drive from the Intel® Matrix RAID array with RAID 0 volume and any other different level volume (RAID 1, RAID 5, RAID 10), then the RAID 0 volume will become failed, but the failed RAID 0 volume will be assembled after system reboots. It has no real impact to the user because the failed RAID volume is not accessible.
Workaround	Not at this time.

§§