

openEuler 20.03 LTS

Release Notes

Date

Contents

| Terms of Use | iii |
|--|-----|
| 1 User Notice | 4 |
| 2 Introduction | 5 |
| 3 Installing the OS | 6 |
| 4 Key Features | 9 |
| 5 Known Issues | 10 |
| 6 Resolved Issues | 11 |
| 7 Common Vulnerabilities and Exposures (CVE) | 12 |
| 8 Source Code | 13 |
| 9 Contribution | 14 |
| 10 Acknowledgement | 15 |

Terms of Use

Terms of Use

Copyright © Huawei Technologies Co., Ltd. 2020. All rights reserved.

Your replication, use, modification, and distribution of this document are governed by the Creative Commons License Attribution-ShareAlike 4.0 International Public License (CC BY-SA 4.0). You can visit https://creativecommons.org/licenses/by-sa/4.0/ to view a human-readable summary of (and not a substitute for) CC BY-SA 4.0. For the complete CC BY-SA 4.0, visit https://creativecommons.org/licenses/by-sa/4.0/legalcode.

Trademarks and Permissions

openEuler is a trademark or registered trademark of Huawei Technologies Co., Ltd. All other trademarks and registered trademarks mentioned in this document are the property of their respective holders.

Disclaimer

This document is used only as a guide. Unless otherwise specified by applicable laws or agreed by both parties in written form, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, including but not limited to non-infringement, timeliness, and specific purposes.

2020-04-03 iii

1 User Notice

- The version number counting rule of openEuler is changed from openEuler *x.x* to openEuler *year.month*. For example, openEuler 20.03 indicates that the version is released in March 2020.
- The Python core team has stopped maintaining Python 2 in January 2020. In 2020, openEuler 20.03 LTS fixes only the critical CVE of Python 2 and will reach the end of maintenance (EOM) on December 31, 2020. Please switch to Python 3 as soon as possible.

2 Introduction

openEuler is an open-source operating system. The current openEuler kernel is based on Linux and supports Kunpeng and other processors. It fully unleashes the potential of computing chips. As an efficient, stable, and secure open-source OS built by global open-source contributors, openEuler applies to database, big data, cloud computing, and artificial intelligence (AI) scenarios. In addition, openEuler community is an open-source community for global OSs. Through community cooperation, openEuler builds an innovative platform, builds a unified and open OS that supports multiple processor architectures, and promotes the prosperity of the software and hardware application ecosystem.

3 Installing the OS

Release Package

The openEuler release package contains ISO files, VM images, container images, and repo sources. Table 3-1 lists the ISO files in the release package. Table 3-3 lists the container images. Repo sources can be used online. Table 3-4 lists the repo source directories.

Table 3-1 ISO files

| ISO File | Description |
|---|--|
| openEuler-20.03-LTS-aarch64-d vd.iso | Basic installation ISO file of the AArch64 architecture, including the core components for running the minimum system. |
| openEuler-20.03-LTS-everything -aarch64-dvd.iso | Full installation ISO file of the AArch64 architecture, including all components for running the entire system. |
| openEuler-20.03-LTS-debuginfo- aarch64-dvd.iso | ISO file for openEuler debugging in the AArch64 architecture, including the symbol table information required for debugging. |
| openEuler-20.03-LTS-x86_64-dv d.iso | Basic installation ISO file of the x86_64 architecture, including the core components for running the minimum system. |
| openEuler-20.03-LTS-everything -x86_64-dvd.iso | Full installation ISO file of the x86_64 architecture, including all components for running the entire system. |
| openEuler-20.03-LTS-debuginfo-x86_64-dvd.iso | ISO file for openEuler debugging in the x86_64 architecture, including the symbol table information required for debugging. |
| openEuler-20.03-LTS-source-dv d.iso | ISO file of openEuler source code. |

Table 3-2 VM images

| Image File | Description |
|------------|-------------|
|------------|-------------|

| Image File | Description |
|--|---|
| openEuler-20.03-LTS.aarch64. qcow2.xz | openEuler VM image in the AArch64 architecture. |
| openEuler-20.03-LTS.x86_64.q cow2.xz | openEuler VM image in the x86_64 architecture. |

M NOTE

The default password of user **root** of the VM image is **openEuler12#\$**. Change the password upon the first login.

 Table 3-3 Container images

| Image File | Description |
|-------------------------------------|--|
| openEuler-docker.aarch64. tar.xz | openEuler container image in the AArch64 architecture. |
| openEuler-docker.x86_64. tar.xz | openEuler container image in the x86_64 architecture. |

Table 3-4 Repo sources

| Directory | Description | |
|---------------------|---|--|
| ISO | Stores ISO images. | |
| OS | Stores basic software package sources. | |
| debuginfo | Stores debugging package sources. | |
| docker_img | Stores container images. | |
| virtual_machine_img | Stores VM images. | |
| everything | Stores full software package sources. | |
| extras | Stores the extended software package sources. | |
| source | Stores source code software sources. | |
| update | Stores upgrade software package sources. | |
| EPOL | Stores the openEuler extended packages. | |

Minimal Hardware Specifications

Table 3-5 lists the minimum hardware specifications for installing openEuler 20.03 LTS.

2020-04-03 7

Table 3-5 Minimal hardware specifications

| Component | Minimal Hardware Specification |
|-----------|--|
| CPU | Kunpeng 920 (architecture: AArch64) x86-64 (Skylake or later) |
| Memory | ≥ 8 GB |
| Hard disk | ≥ 120 GB |

Hardware Compatibility

Table 3-6 lists the typical configurations of servers and components supported by openEuler. openEuler will gradually support other servers in the future. Partners and developers are welcome to participate in the contribution and validation.

Table 3-6 Supported servers and typical configurations

| Ven dor | Server | Server Model | Component | Typical Configuration |
|------------|----------------------|-------------------------|----------------------------|---|
| Hua | TaiShan 200 | 2280 balanced | CPU | HiSilicon Kunpeng 920 |
| wei | | model | Memory | 32G*4 2933MHz |
| | | | RAID controller card | LSI SAS3508 |
| | | | Network | TM210 |
| Hua wei | FusionServe r Pro | 2288H V5 rack server | CPU | Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz |
| | | | Memory | 32*4 2400MHz |
| | | | RAID controller card | LSI SAS3508 |
| | | | Network | X722 |

4 Key Features

- iSula lightweight container solution, unified IoT, and edge and cloud computing container solutions
 - Shortens a trace chain by three levels, and the memory usage of hundreds of containers is significantly lower than that of the Docker engine.
 - Supports standard open-source container runtime interface (CRI) and open container initiative (OCI) and flexibly interconnects with multiple OCI runtimes such as runC and Kata.
 - Secure container: combines the virtualization technology and container technology to ensure better isolation of secure containers.
 - System container: supports local file system startup to implement quick deployment, and supports systemd deployment to improve user namespace isolation.
- Kunpeng acceleration engine (KAE), supporting encryption and decryption acceleration
 - Digest algorithm SM3, which supports asynchronous models.
 - Symmetric encryption algorithm SM4, which supports asynchronous models and CTR, XTS, and CBC modes.
 - Symmetric encryption algorithm AES, which supports asynchronous models and ECB, CTR, XTS, and CBC modes.
 - Asymmetric algorithm RSA, which supports asynchronous models and key sizes 1024, 2048, 3072, and 4096.
 - Key negotiation algorithm DH, which supports asynchronous models and key sizes 768, 1024, 1536, 2048, 3072, and 4096.
- A-Tune intelligent system performance optimization engine, inferring service features and configuring optimal system parameters to ensure optimal service running
- Enhancing the performance of glibc, zlib, and gzip and fully using the NEON instruction set of AArch64 to improve the basic library performance
- Kernel feature enhancement
 - Supports ARM64 kernel hot patches.
 - Numa Aware Qspinlock: reduces cache/bus conflicts across NUMA nodes.
 - Optimizes the IOVA page table lookup and release algorithms to improve the performance of the IOMMU subsystem.
 - Optimizes the implementation of CRC32 and checksum based on ARM64 instructions and pipeline features, greatly improving data verification performance.
 - Supports ARM v8.4 Memory System Resource Partitioning and Monitoring (MPAM).

5 Known Issues

- The FIPS boot mode of the kernel has not been fully authenticated. The FIPS boot may be abnormal. I17Z18
- When libvirt is used to start the GlusterFS VM, a 300-byte memory leak occurs each time. For details about the discussion, click https://github.com/gluster/glusterfs/issues/818. I185CH
- When the libvirt interface is used to continuously perform disk hot swap operations, there is a possibility that the hot remove interface returns a success message, but the disk is not removed and cannot be hot swapped again. You can stop the VM and then restart it. I1C72L
- There is a low probability that an unknown installation exception occurs when the x86_64 VM is used for installation. In this case, install the x86_64 VM again. I1C8HS
- CVE-2012-0039: When a local application calls the g_str_hash function, the application continuously consumes CPU resources, causing DoS attacks. This issue will not be resolved in the community.
- CVE-2015-9541: When Qt attempts to parse the abnormal SVG files which are
 constructed to launch exponential XML entity extension attacks, the memory may be
 insufficient. For details about the discussion, click
 https://codereview.qt-project.org/c/qt/qtbase/+/293909.
- Before compiling some open-source packages, you need to install basic software such as GDB, GCC, and make. Otherwise, the compilation fails due to lack of dependency.
- AArch64 and x86_64 have different definitions of the character type. As a result, an error is reported during the self-check using Coreutils, Augeas, and Diffutils. You can add the **--fsigned-char** compilation option to solve the problem.

6 Resolved Issues

For details about the complete issue list, click https://gitee.com/organizations/src-openeuler/issues.

For details about the complete kernel submission records, click https://gitee.com/openeuler/kernel/commits/openEuler-1.0-LTS.

Table 6-1 lists the resolved issues.

Table 6-1 Resolved issues

| Issue | Description |
|---------|---|
| I1BJTF | [Kernel bug] The Iscpu command on the ARM server cannot be used to display the CPU dominant frequency, and the CPU cache is incorrect. |
| I1BWPD | Failed to pull an image using the isula pull or curl pull command. |
| I1BV56 | Delete redundant gpg sig file for shadow-4.6. |
| I1BV38 | The unbuffer command is unavailable. |
| I1BA9B | The arping -w parameter is invalid. |
| I1AV3S | The oops error occurs when the latest LTP pty03 test case is executed. |
| I1AZ1I | 500 scheduled tasks are started. After 4 to 5 minutes, the tasks cannot be processed and the system stops responding. |
| I1AH2C | The warning information captured when the Kata container fails to be started is insufficient for fault locating. More errors need to be printed. |
| I1AGXO | In kata-runtime remote mode, the kata-runtime kill is not called when the isula rm -f command is executed. As a result, residual data exists. |
| I1AF39 | The soft lockup is found when the open function is triggered in the ext4 file system. |
| I1 ADUD | Isulad breaks down when a pod is created using kubectl. |

Common Vulnerabilities and Exposures (CVE)

For CVE involved in the version, see the CVE list.

8 Source Code

openEuler contains two code repositories:

- Code repository: https://gitee.com/openeuler
- Software package repository: https://gitee.com/src-openeuler

The openEuler release packages also provide the source ISO files. For details, see 3 Installing the OS.

9 Contribution

As an openEuler user, you can contribute to the openEuler community in multiple ways. For details about how to contribute to the community, see Contributions to the Community. Here, some methods are listed for reference.

Special Interest Groups (SIGs)

openEuler brings together people of common interest to form different special interest groups (SIGs). For details about existing SIGs, see the SIG list.

You are welcome to join an existing SIG or create a SIG For details about how to create a SIG see the SIG Management Procedure.

Mail List and Tasks

You are welcome to actively help users solve problems raised in the mail list and issues (including code repository issues and software package repository issues). In addition, you can submit an issue. All these will help the openEuler community to develop better.

Documents

You can contribute to the community by submitting code. We also welcome your feedback on problems and difficulties, or suggestions on improving the usability and integrity of documents. For example, problems in obtaining software or documents and difficulties in using the system. Welcome to pay attention to and improve the documentation module of the openEuler community.

IRC

openEuler has also opened a channel in IRC as an additional channel to provide community support and interaction. For details, see openEuler IRC.

10 Acknowledgement

We sincerely thank all the members who participated in and assisted in the openEuler project. It is your hard work to make the version released successfully and provide the possibility for the better development of openEuler.