openEuler 21.09

Technical White Paper

CONTENTS

01 /

Introduction

openEuler has evolved from a simple server operating system (OS) into a digital infrastructure OS that fits into any server, cloud computing, edge computing, and embedded deployment. It provides a secure, stable, and easy-to-use open source OS that is compatible with multiple computing architectures. openEuler suits operational technology (OT) applications and enables the convergence of OT and information and communications technology (ICT).

The openEuler open source community is a portal available to global developers, with the goal of building an open, diversified, and architecture-inclusive software ecosystem for all digital infrastructure scenarios. It has a rich history of helping enterprises develop their software, hardware, and applications.

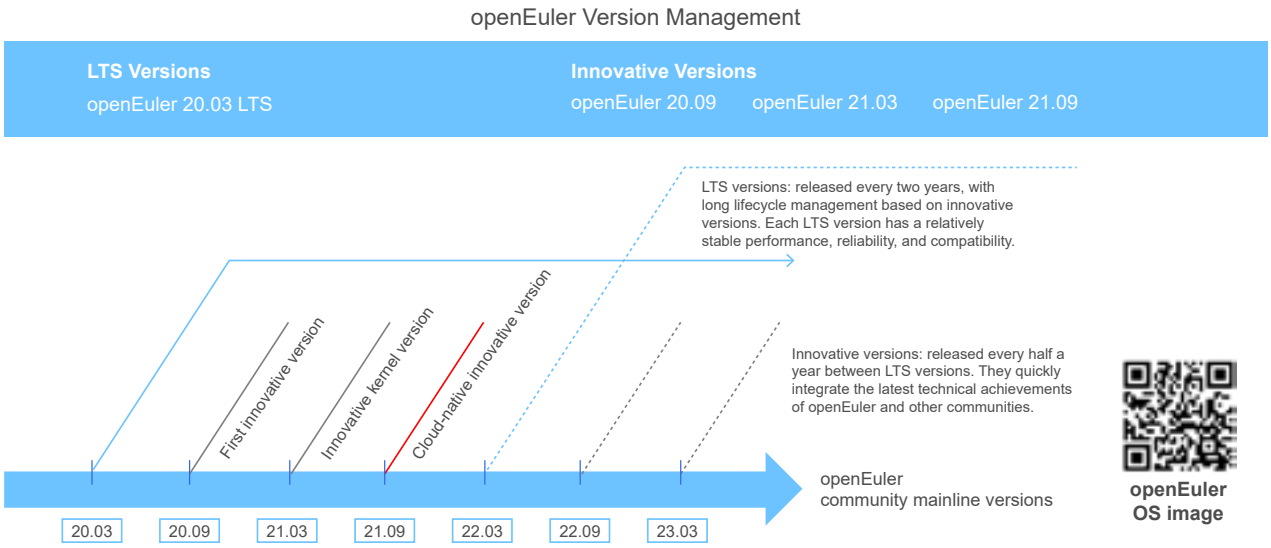
The openEuler open source community was officially established on December 31, 2019, with the original focus of innovating diversified computing architectures.

On March 30, 2020, the Long Term Support (LTS) version openEuler 20.03 was officially released, which was a new Linux distribution with independent technology evolution.

Later in 2020, on September 30, the innovative openEuler 20.09 version was released through the collaboration efforts of multiple companies, teams, and independent developers in the openEuler community. The release of openEuler 20.09 marked a milestone not only in the growth of the openEuler community, but also in the history of open sourced software in China.

On March 31, 2021, the innovative kernel version of openEuler 21.03 was released. This version is enhanced in line with Linux kernel 5.10 and also incorporates multiple new features, such as live kernel upgrade and tiered memory expansion. These features improve multi-core performance and deliver the computing power of one thousand cores.

Fast forward to September 30, 2021, openEuler 21.09 was released. This premium version is designed to supercharge all scenarios, including edge and embedded devices. It enhances server and cloud computing features, and incorporates key technologies including cloud-native CPU scheduling algorithms for hybrid service deployments and KubeOS for containers.

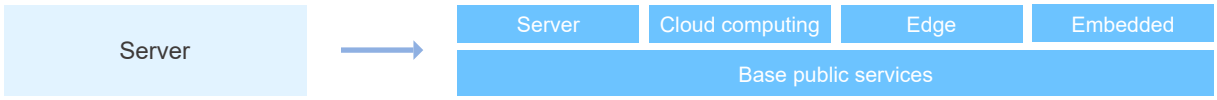


As an OS platform, openEuler releases an updated long-term service (LTS) version every two years. Each LTS version provides enhanced specifications and a secure, stable, and reliable OS for enterprise users.

openEuler is built on tried-and-tested technologies. A new openEuler version is released every 6 months to quickly integrate the latest technical achievements of openEuler and other communities. The innovative tech is first verified in the openEuler open source community as a single open source project, and then these features are added to each new release, enabling community developers to obtain the source code.

Technical capabilities are first tested in the open source community, and continuously incorporated into each openEuler release. In addition, each release is built on feedback given by community users to bridge the gap between innovation and the community, as well as improve existing technologies. openEuler is both a release platform and incubator of new technologies, working in a symbiotic relationship that drives the evolution of new versions.

Innovative Platform for All Scenarios

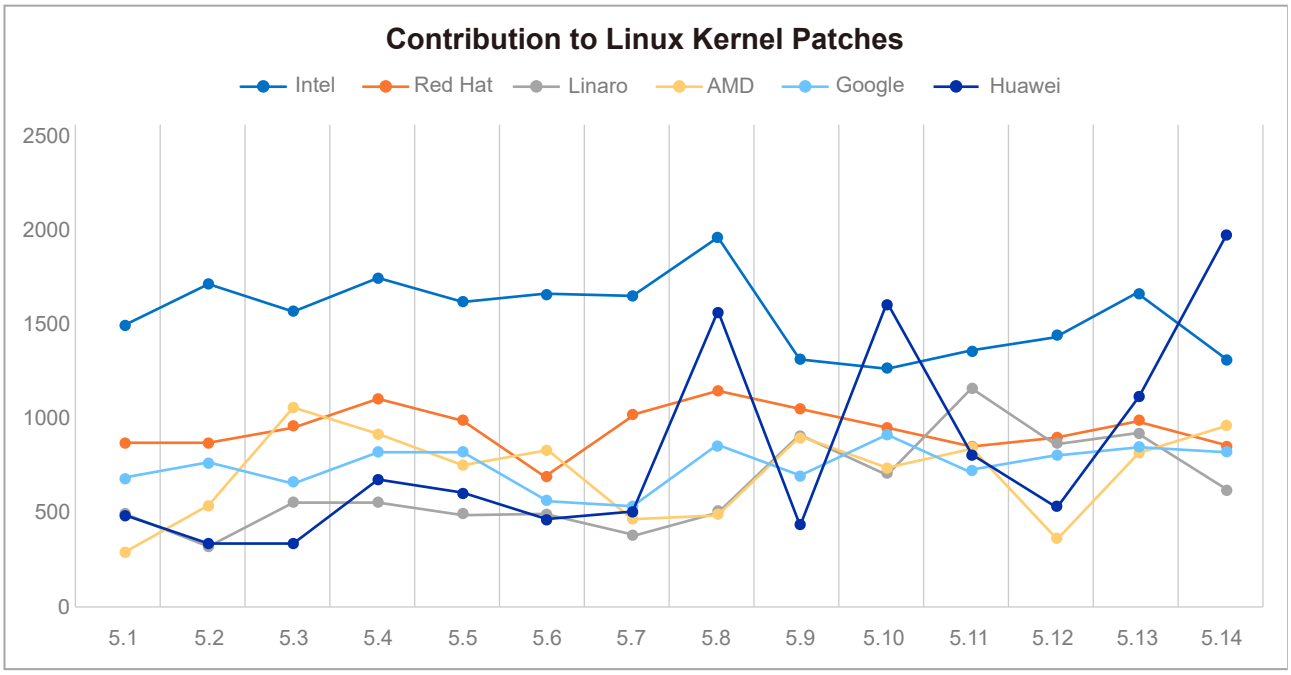


openEuler supports multiple processor architectures (x86, ARM, RISC-V) and will support other brands (PowerPC, SW64) in the future, as part of a focus to continuously improve the ecosystem of diversified computing power.

The openEuler community is home to an increasing number of special interest groups (SIGs), which are dedicated teams that help extend the OS features from server to cloud computing, edge computing, and embedded scenarios. The openEuler OS covers all scenarios, and comprises the 21.09 Edge and 21.09 Embedded editions designed for edge computing and embedded deployments, respectively.

The OS is a perfect choice for ecosystem partners, users, and developers who plan to enhance scenario-specific capabilities. By creating a unified OS that supports multiple devices, openEuler hopes to enable a single application development for all scenarios.

Continuous Contribution to the Linux Kernel



As a major contributor to the Linux kernel, the kernel development team is responsible for enhancing the processor architectures, Advanced Configuration and Power Interface (ACPI), memory management, file systems, media, kernel documents, bug fixes, and code rebuilds. Over the past decade, openEuler has contributed more than 17,000 patches to the Linux kernel.

In Linux kernels 5.10 and 5.14, openEuler's code contribution ranks No.1 in the world. openEuler is committed to kernel innovation and has been continuously contributing to upstream communities.

Open and Transparent: The Open Source Software Supply Chain

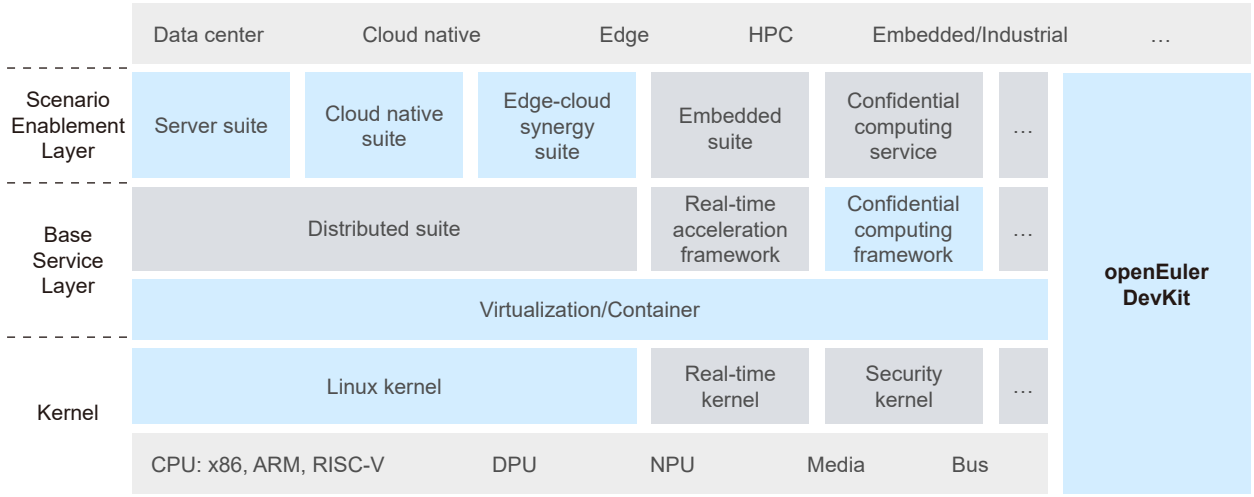
The process of building an open source OS relies on supply chain aggregation and optimization. To ensure reliable open source software or a large-scale commercial OS, openEuler comprises a complete lifecycle management that covers building, verification, and distribution. The brand regularly reviews its software dependencies based on user scenarios, organizes the upstream community addresses of all the software packages, and verifies its source code by comparing it to that of the upstream communities. The build, runtime dependencies, and upstream communities of the open source software form a closed loop, realizing a complete, transparent software supply chain management.

02/

Platform Architecture

System Framework

openEuler is an innovative open source OS platform built on kernel innovations and a solid cloud base to cover all scenarios. It is built on the latest trends of interconnect buses and storage media, and offers a distributed, real-time acceleration engine and base services. It provides competitive advantages in edge and embedded scenarios, and is the first step to building a full-scenario digital infrastructure OS.



Kernel innovations:

- **Enhanced cloud-native scheduling:** openEuler suits hybrid deployments of online and offline cloud services. Its innovative CPU scheduling algorithm ensures real-time CPU preemption and jitter suppression for online services. Additionally, its innovative memory reclamation algorithm against out of memory (OOM) allows online services to run reliably based on their higher service priorities.
- **EulerFS:** A new file system is designed for non-volatile dual in-line memory modules (NVDIMMs). It uses technologies such as soft updates and dual-view directories to accelerate file metadata synchronization and thus improve file read and write performance.
- **Tiered memory expansion etMem:** With the user-mode swap function, the discarded cold memory can be changed to the user-mode storage based on a preset policy. The user-mode swap delivers a higher performance than the kernel-mode swap and the whole swap process is transparent to users.

Cloud base:

- **KubeOS for containers:** In cloud native scenarios, the OS is deployed and maintained in containers, allowing the OS to be managed based on Kubernetes, just as service containers.
- **Secure container solution:** Compared with the traditional Docker+QEMU solution, the iSulad+shimv2+StratoVirt secure container solution reduces the memory overhead and boot time by 40%.
- **Dual-plane deployment tool eggo:** OSs can be installed with one click for ARM and x86 hybrid clusters, while deployment of a 100-node cluster is possible within just 15 minutes.

Scenario-specific innovations:

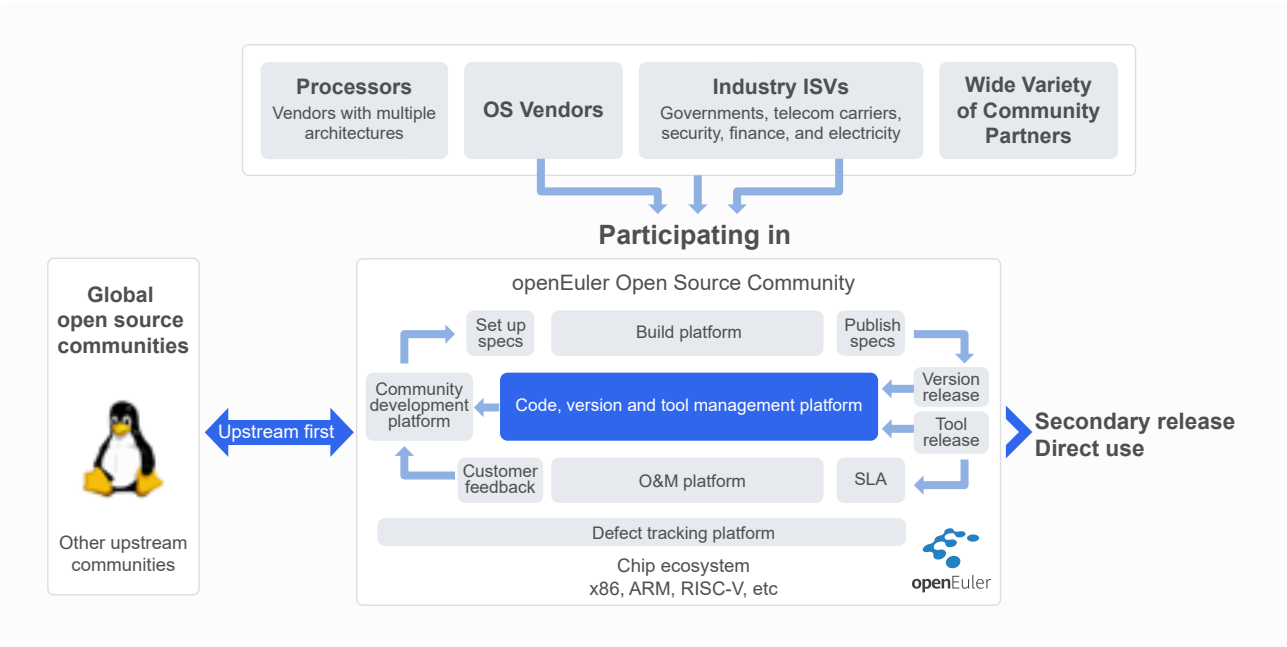
- **Edge computing:** openEuler 21.09 Edge is released for edge computing scenarios. It integrates the KubeEdge+ edge-cloud synergy framework to provide unified management, provisioning of edge and cloud applications, and other capabilities.
- **Embedded:** openEuler 21.09 Embedded is released for embedded scenarios, helping compress images under 5 MB and image loading within 5 seconds.

Flourishing community ecosystem:

- **Desktop environments:** UKUI, DDE, and Xfce.
- **openEuler DevKit:** Supports OS migration, compatibility evaluation, and various development tools such as secPaver, to simplify security configuration.

Platform Framework

The openEuler open source community partners with upstream and downstream communities to advance the evolution of openEuler versions.



03/

Operating Environments

Servers

To install openEuler on a physical machine, check that the physical machine meets the compatibility and hardware requirements. For a full list, visit https://openeuler.org/en/compatibility/.

Item	Configuration Requirement
Architecture	ARM64, x86_64
Memory	At least 4 GB
Drives	At least 20 GB

VMs

openEuler supports the following virtual machines (VMs):

- centos 7.9 qemu 1.5.3-175.el7 libvirt 5.0.0-1.el7
- centos 8 qemu 5.1.0-20.el8 libvirt 6.6.0-7.3.el8
- fedora 29 qemu 3.0.0-1.fc29 libvirt 4.7.0-1.fc29
- fedora 32 qemu 4.2.0-7.fc32 libvirt 6.1.0-2.fc3

Item	Configuration Requirement
Architecture	ARM64, x86_64
CPU	2 CPUs
Memory	At least 4 GB
Drives	At least 20 GB

Edge Devices

To install openEuler on an edge device, check that the edge device meets the compatibility and hardware requirements.

Item	Configuration Requirement
Architecture	ARM64, x86_64
Memory	At least 4 GB
Drives	At least 20 GB

Embedded Devices

To install openEuler on an embedded device, check that the embedded device meets the compatibility and minimum hardware requirements.

Item	Configuration Requirement
Architecture	ARM64, ARM32
Memory	At least 128 MB
Drives	At least 256 MB

04/ Kernel Innovations

New Features in the openEuler Kernel

openEuler 21.09 runs on Linux Kernel 5.10 and is built on 12 innovations that supercharge scheduling, memory, and networking. These are as follows:

01 Optimized process scheduling:
The optimized load balancing algorithm decreases the overheads caused by load balancing between processes and improves application performance.

02 Dynamic kernel preemption:
The new boot option **preempt=none/voluntary/full** allows the kernel to dynamically change the preemption mode.

03 Optimized mremap performance:
Entries at the page middle directory (PMD) and page upper directory (PUD) levels can be moved to map large memory blocks quickly.

04 Per memcg lru lock:
It alleviates lock contention between cloud native container instances to improve system performance.

05 Optimized huge page memory management:
The tail pages among HugeTLB pages are released in shared mapping mode to reduce the overheads incurred in managing the huge page memory.

06 Concurrent updates for translation lookaside buffer (TLB):
The local TLB and remote TLB can be refreshed at the same time to optimize the TLB shootdown process, for quicker TLB refreshes and improved service performance.

07 Optimized huge page vmalloc performance:
When calling vmalloc() to allocate space that is larger than the minimum size of huge pages, the huge page, not the base page, is used to map the memory, improving the TLB utilization and reducing TLB misses.

08 Memory reclamation against OOM:
In the event of OOM, memory reclamation is preferentially performed for process groups with low priorities to ensure the normal running of online services.

09 Pointer authentication codes (PACs):
The register value is authenticated before being used as a pointer to access the data or code, so as to defend against return-oriented programming (ROP) and jump-oriented programming (JOP) attacks.

10 Branch Target Identifiers (BTIs):
A BTI instruction is used to guard against the execution of instructions which are not the intended target of a branch. BTIs and PACs combine to reduce control flow attacks.

11 eXpress Data Path (XDP):
An eBPF-based high performance, programmable network data path that processes data before network packets enter the network protocol stack to improve network performance. XDP works in various scenarios such as anti-DDoS, firewall, and network quality of service (QoS).

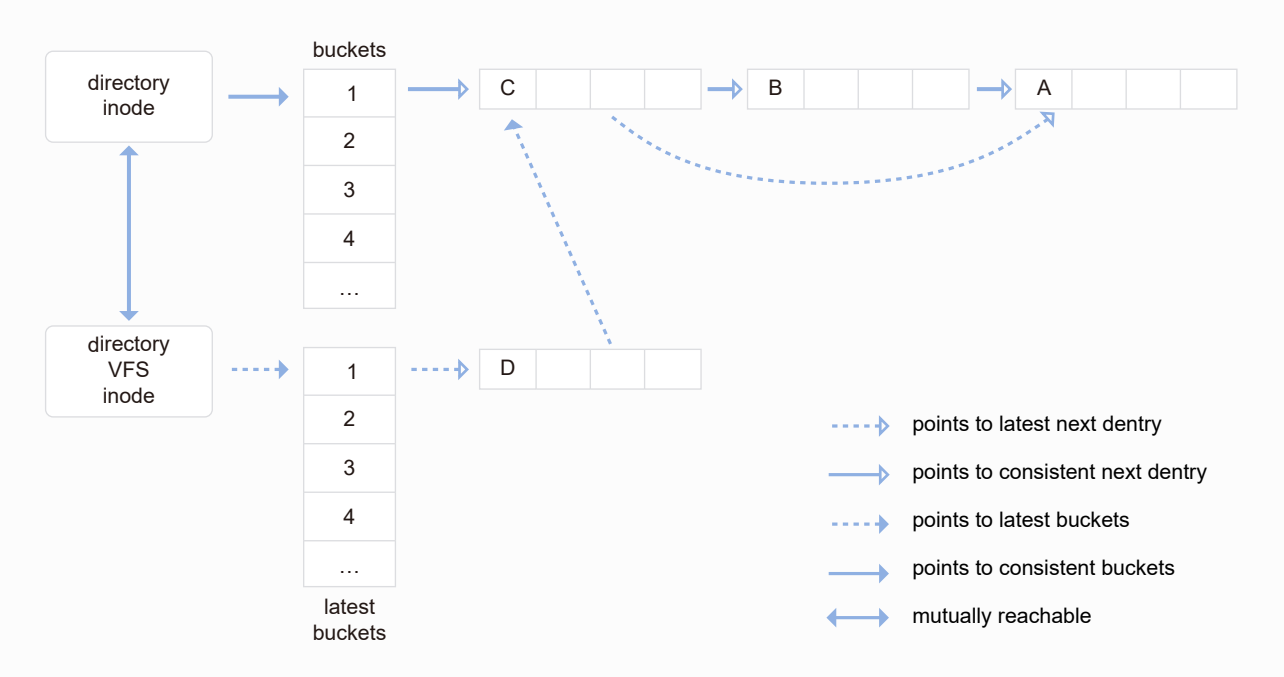
12 Shared Virtual Addressing (SVA):
It allows host processes and devices to use the same virtual address, avoiding the need for replicating resources between hosts and devices. SVA improves service communication performance across hosts and devices.

File System for New Media

NVDIMMs, such as Intel Optane, are new high-speed storage media that provide byte-level access. The popular kernel file system EXT4 can work with the Direct Access (DAX) feature to improve the data read and write performance of NVDIMMs. However, in terms of metadata management, EXT4 offers insufficient performance in the journal synchronization mechanism, which causes high metadata management overheads and write amplification. These common issues prevent NVDIMM advantages from being fully utilized.

The new file system, EulerFS, is equipped with metadata soft update technology. Its pointer-based directory and dual-view counting mechanism reduces metadata synchronization overheads to effectively improve the call performance of file system functions (create, unlink, mkdir, rmdir). EulerFS delivers much shorter latency and higher bandwidth over EXT4 and DAX alternatives.

Feature Description



- **Hash table directory:** Hash tables are used to manage directory entries, improving linear search efficiencies and reducing pseudo-sharing.
- **Unified allocator:** Data structures have a unified allocator, which breaks the barriers between different data structures and facilitates memory management.
- **Soft update:** This lightweight technology simplifies the function implementation and ensures file system consistency.
- **Pointer-based directory and dual-view counting:** This mechanism reduces metadata synchronization overheads and effectively improves the read and write performance of a file system.
- **Dependency tracing:** Directory entry operations (creation, deletion, etc) are not automatically persisted. After such an operation is complete, only the dependency persistence is traced in inodes. These operations become persistent in asynchronous mode to reduce the metadata operation latency.

Application Scenarios

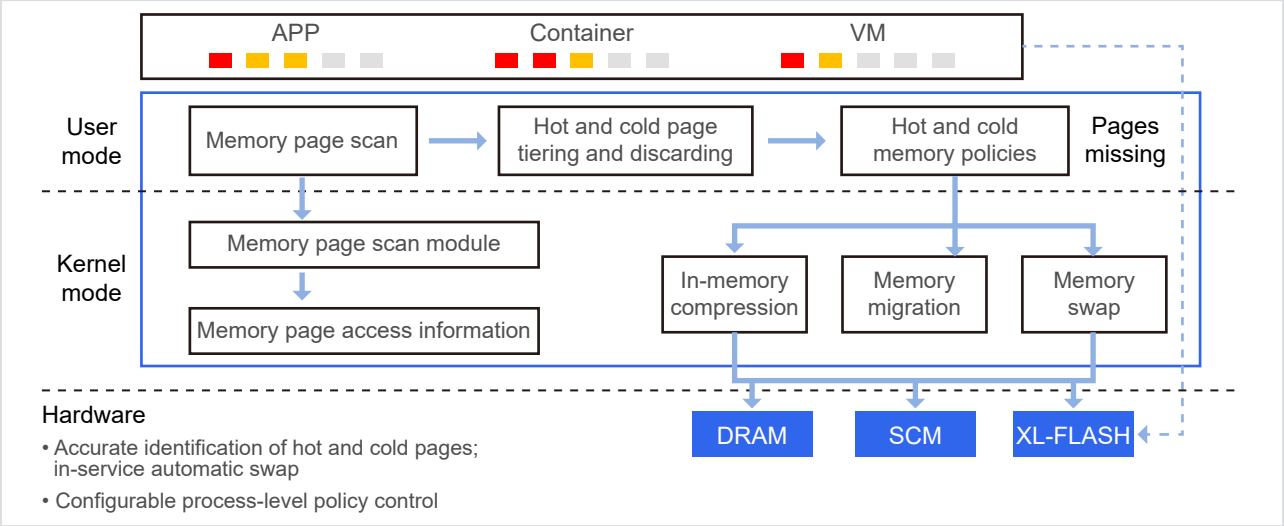
EulerFS is perfectly suited to compliment new media, such as NVDIMMs. It is an effective alternative to EXT4 and XFS in single-node applications and cloud-native distributed applications.

Tiered Memory Expansion

Today's memory manufacturing processes have plateaued in terms of development. While the mature ARM ecosystem helps slash CPU core costs, new technologies like VMs, big data, artificial intelligence (AI), and deep learning require higher computing power and memory capacities. A system that offers limited potential in either aspect is a huge challenge for future growth.

One solution is etMem for tiered memory expansion. DRAM and low-speed memory media, such as storage class memory (SCM), Apache Pass (AEP), and remote direct memory access (RDMA), form a multi-tier memory structure. Automatic memory scheduling redirects hot data to the DRAM high-speed memory area and cold data to the low-speed memory area. This tiered memory structure increases the memory capacity and ensures efficient and stable running of core services. etMem is ideal for applications that use a large amount of memory but do not access the memory frequently. Tests show that etMem delivers 40% higher performance for MySQL than counterparts. The user-mode memory swapping mechanism is added for the user-mode storage framework and user requirements.

Feature Description



openEuler 21.09 inherits the following features from its earlier version:

- **Process-level control:** The etMem configuration file can be used to expand the memory. Compared with the native LRU-based pageout kswap mechanism, etMem is more flexible and accurate.
- **Cold and hot tiering:** In user mode, a memory access scan can be performed for a designated process. You can configure a cold and hot tiering policy to classify memory access results into hot memory and cold memory.Soft update: This lightweight technology simplifies the function implementation and ensures file system consistency.
- **Discarding policy:** The cold memory is discarded when it meets the conditions specified in the etMem configuration file and the system environment configuration. The discarding process uses the native kernel capability, which is secure and reliable and does not affect user experience.

openEuler 21.09 now provides the following new feature:

User-mode swap. The discarded cold memory can be swapped to the user-mode storage based on a preset etMem policy. The user-mode swap delivers a higher performance than the kernel-mode swap and the whole swap process is transparent to users.

Application Scenarios

Tiered memory expansion is available for all service processes on a node. etMem excels in applications that use a large amount of memory but do not access the memory frequently, such as MySQL, Redis, and Nginx. All memory expansion operations are performed internally, that is within a node, and no cross-node operations are involved.

In a user-mode storage scenario, the userswap function can enable user-mode storage as a swap device.

05/

Cloud Base

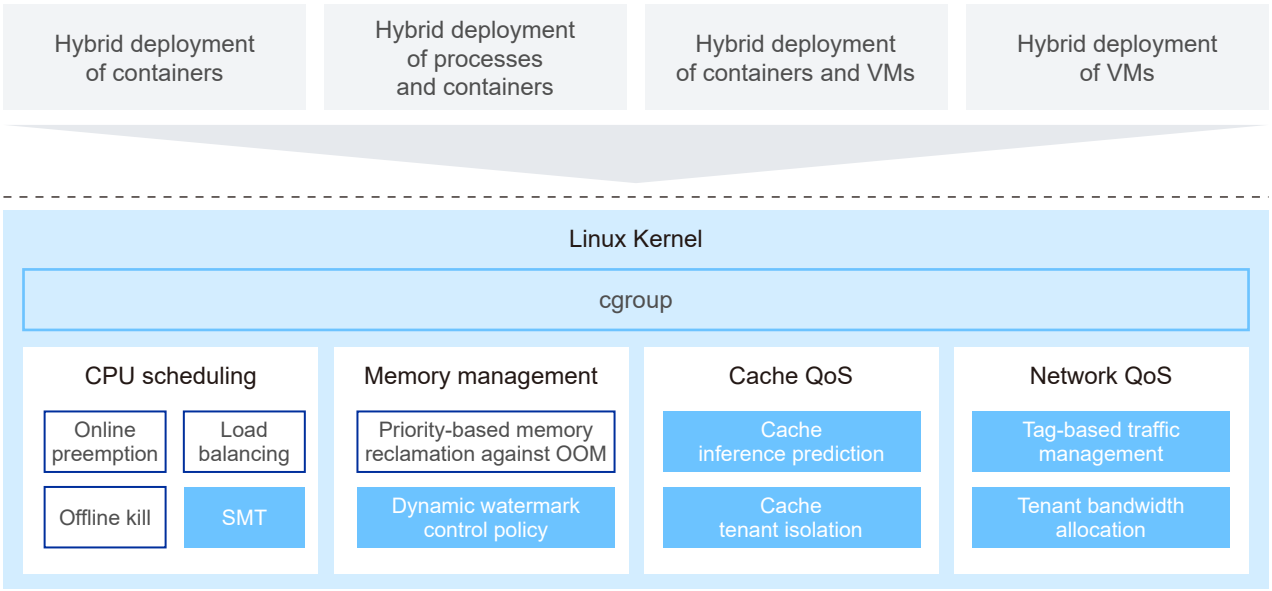
Enhanced Cloud-Native Scheduling

On the cloud, online interactive services are sensitive to latencies, and any impact can have a tidal effect. As a result, the average CPU usage is low (≈15%). Many enterprises adopt a hybrid deployment of online and offline services to improve resource utilization. However, the common kernel resource allocation and management mechanism is insufficient, and is the common cause of fluctuations in performance and QoS for online services. openEuler is equipped with its own CPU scheduling and memory reclamation algorithms that are designed for hybrid deployments of cloud native services. These innovative algorithms improve CPU utilization and ensure the QoS of online services.

One scheduling algorithm is quality aware scheduler (QAS), which is dedicated to cloud native services in hybrid deployments. QAS ensures that online tasks quickly preempt CPU resources, schedules tasks in a deterministic manner, and suppresses interference from offline tasks.

openEuler has optimized its memory reclamation algorithm against OOM. In the event of OOM, memory reclamation is preferentially performed for process groups with low priorities to ensure the normal running of online services.

Feature Description



- **Process property settings:** The cgroup interface can be called to set the task execution property, either to online or offline.
- **Preferential tasks:** When the CPU is processing online and offline tasks at the same time, the CPU preferentially processes online tasks, which can preempt CPU resources within microseconds over offline services.
- **Kill mechanism for offline tasks:** Offline tasks quickly respond to a received kill signal and exit in a short time.
- **Memory reclamation against OOM:** If OOM occurs, memory reclamation is preferentially performed on process groups that have low priorities. This ensures the normal running of online services.

Application Scenarios

Hybrid deployment of latency-sensitive interactive services (MySQL, Redis, and Nginx) and latency-insensitive services (offline AI training). Examples include hybrid deployments of containers and containers, containers and processes, containers and VMs, and VMs and VMs.

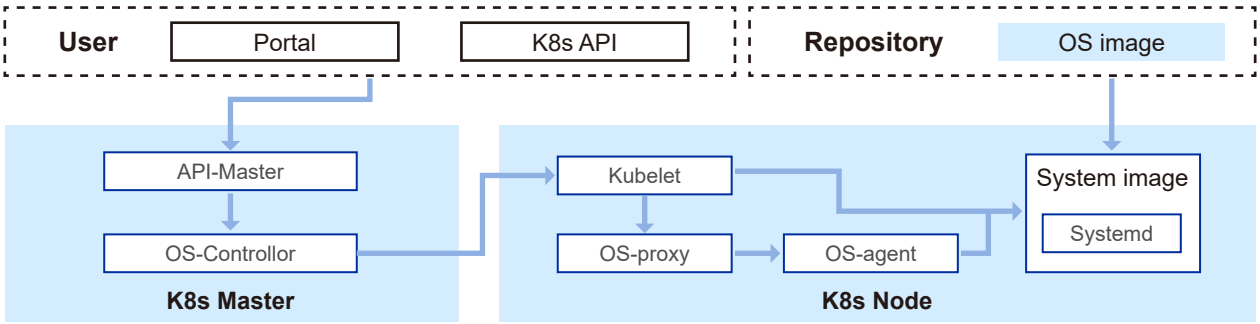
Container OS

Cloud native is the next step in cloud computing evolution. Kubernetes has become the foundation for most modern, cloud-native software infrastructure. Major OS vendors have launched their OSs for cloud native scenarios, including Red Hat Enterprise Linux CoreOS (RHCOS) and AWS Bottlerocket. These OSs are deployed and managed in containers, and offer an O&M experience similar to containers.

To adapt to this trend, openEuler has launched KubeOS, an OS that centrally manages cloud-native cluster OSs in containers. KubeOS has the following features:

- OS containerization and Kubernetes interconnection for atomized lifecycle management
- Lightweight OS cropping, which reduces unnecessary packages to quicken upgrades and replacements

Feature Description



- **OS-Controller:** The global manager monitors the OS instances and collects their information on all nodes. It implements global OS lifecycle management, including upgrades, restarts, and configurations, and evicting pods when the OSs are shut down.
- **OS-proxy:** The node proxy resides and runs on the worker node. It monitors the OS instance and collects its information on a single node, and then reports the information to the OS-agent.
- **OS-agent:** It receives commands from the OS-proxy to manage the container OS lifecycle, for example, installing and upgrading the container OS.

Application Scenarios

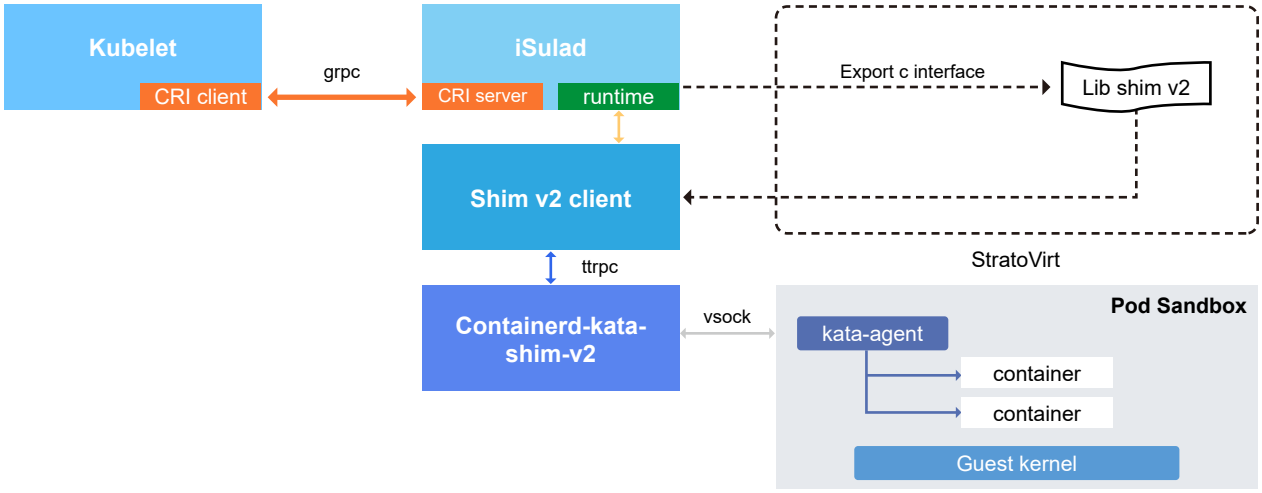
KubeOS is designed to manage the OSs of containerized cloud service hosts that run on Kubernetes, delivering a lifecycle management and O&M experience similar to container services.

Secure Container Solution

Containers are a mainstream cloud computing technology, and simplify encapsulation, accelerate deployments, and reduce environment dependencies. Containers can directly access host resources and share the host kernel. However, this can cause security vulnerabilities, such as container escapes, making common containers insufficient for most security isolation requirements in multi-tenant financial applications. Alibaba and Intel are leading players driving Kata open source projects. Both these companies use containers at the isolated virtualization layer to form a secure container solution. Likewise, Google launches the GVisor security sandbox for process-level virtualization, isolating risks from container applications.

By contrast, openEuler combines the virtualization platform StratoVirt and the container management engine iSulad to form a secure container solution. openEuler reduces the memory overhead and boot time by more than 40% over the Docker+QEMU solution. It provides a lightweight and secure execution environment for applications, isolating security risks in either container and host OS environments.

Feature Description



Virtualization platform StratoVirt offers the following features:

- **Robust security:** Offers language-level security based on Rust. Its modular design minimizes the attack surface and physically isolates each tenant.
- **Lightweight and low-overhead:** When using a simplified device model, it can start within 50 ms, and control the memory overhead within 4 MB.
- **Converged software and hardware:** Supports x86 and Kunpeng-V virtualization.
- **Lightning-fast scaling:** Achieves device scaling within milliseconds, providing flexible resource scaling capabilities for lightweight workloads.
- **Multi-scenario support:** Supports multiple application scenarios, including serverless, secure containers, and standard VMs, with just a single architecture.

Container engine iSulad comprises the following benefits:

- **Lightweight engine:** Uses the C/C++ programming language to reconstruct the lightweight container base to adapt to edge and cloud scenarios.
- **Hybrid scheduling:** Uses containerd-shim-kata-v2 to interwork with StratoVirt, and schedules and deploys StratoVirt and Kubernetes on a unified platform.

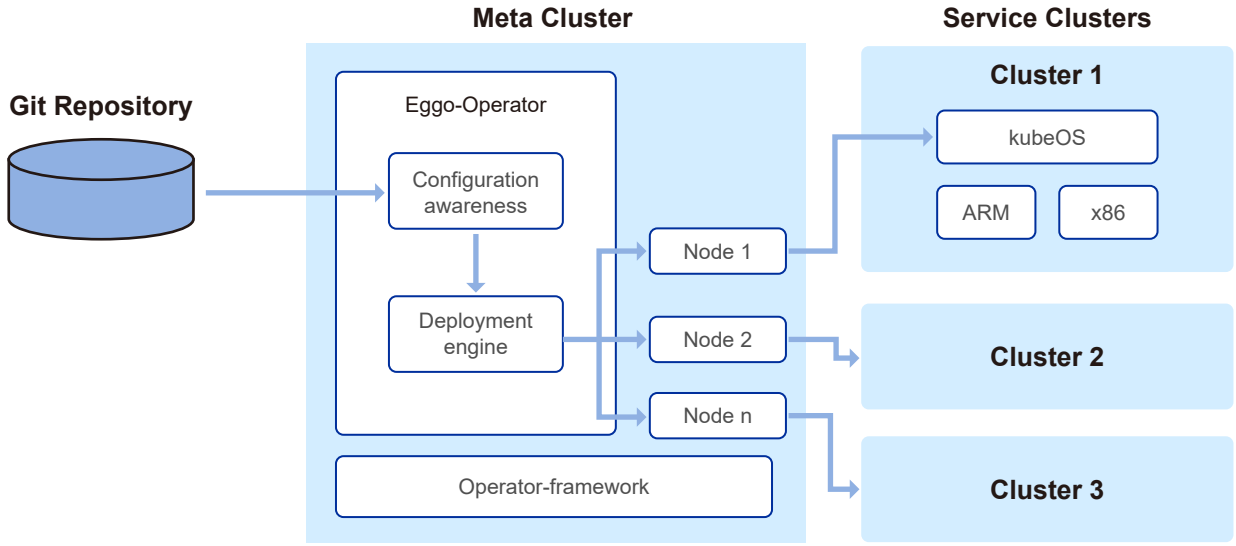
Application Scenarios

The secure container solution replaces Docker containers to ensure container service security and isolation. It is suited for multi-tenant applications and finance, telecom, and security fields.

Dual-Plane Deployment Tool

eggo is a Kubernetes cluster deployment and management project initiated by the openEuler SIG **sig-CloudNative**. It provides efficient and stable cluster deployment (online and offline) for a single cluster over multiple architectures. It interworks with GitOps to detect cluster configuration changes and enable unified and efficient deployment of cluster OSs.

Feature Description



- **Version-based cluster management:** Git repositories are used to store and track cluster configuration changes.
- **Configuration awareness:** GitOps detects cluster configuration changes in Git repositories, and then sends a cluster operation request to the deployment engine.
- **Deployment engine:** It delivers service cluster tasks. For example, it triggers tasks such as deploying the service cluster, destroying the service cluster, adding nodes, and deleting nodes.

Application Scenarios

eggo is designed to fit in x86 and ARM dual-plane cloud infrastructures. Because it runs on the Kubernetes cloud native framework, the dual-plane deployment tool implements unified cluster deployment, monitoring, and audit of OSs.

06/

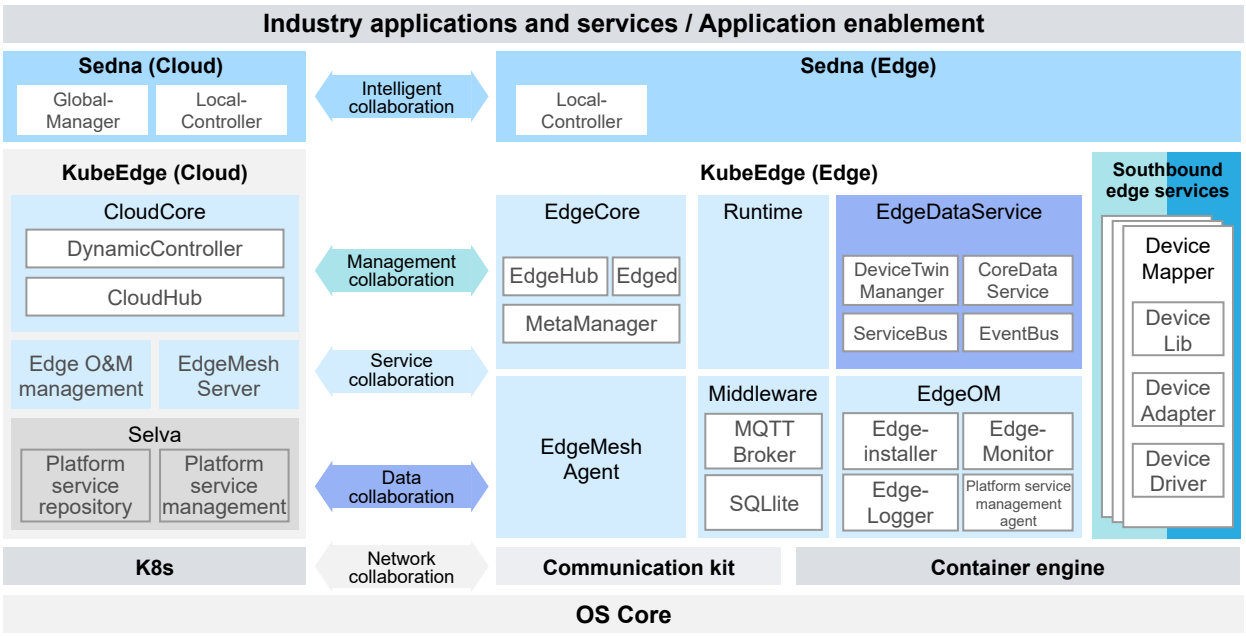
Scenario-Specific Innovations

Edge Computing

Edge computing is one of the 10 major technology trends, and new edge technologies are needed to handle the huge data volumes generated by smart city, autonomous driving, and industrial Internet applications. IDC forecasts that up to 48.6 ZB data will be generated in China by 2025, and centralized cloud computing will no longer be able to meet the necessary bandwidth load, network latency, and data management costs. In short, edge computing is vital to many industries.

openEuler 21.09 Edge integrates the KubeEdge+ framework to provide unified management and provisioning of edge applications. It enhances intelligent collaboration between edge and cloud to help improve AI service deployment, implementing edge-cloud discovery and traffic forwarding, as well as improving southbound capabilities.

Feature Description



openEuler 21.09 Edge comprises the following features:

Unified edge-cloud synergy framework KubeEdge+. This means it not only provides basic capabilities such as edge-cloud application management and deployment, but also premium communications between edge and cloud devices, as well as subsequent management of southbound peripherals.

While the 21.09 version is suitable to cover most deployment scenarios, future releases will include the following features:

- **Edge-cloud collaboration.** EdgeMesh Agent and EdgeMesh Server are deployed on the edge and the cloud, respectively, to implement smart service discovery and service routing.
- **Optimized southbound edge.** Device Mapper is used for southbound access and provides the peripheral profiler and parsing mechanisms. It manages and controls southbound peripherals and service streams. It is compatible with the EdgeX Foundry open source ecosystem.
- **Edge data services.** These services can implement on-demand data persistence in messages, media, or other data streams, and provide data analysis and data export capabilities.
- **Intelligent Sedna architecture.** The open source Sedna framework provides base edge-cloud collaborative inference, federated learning, and incremental learning capabilities. It facilitates easy model and dataset management, enabling personnel to quickly develop and deploy AI applications to the edge, and improving the training and deployment efficiency.

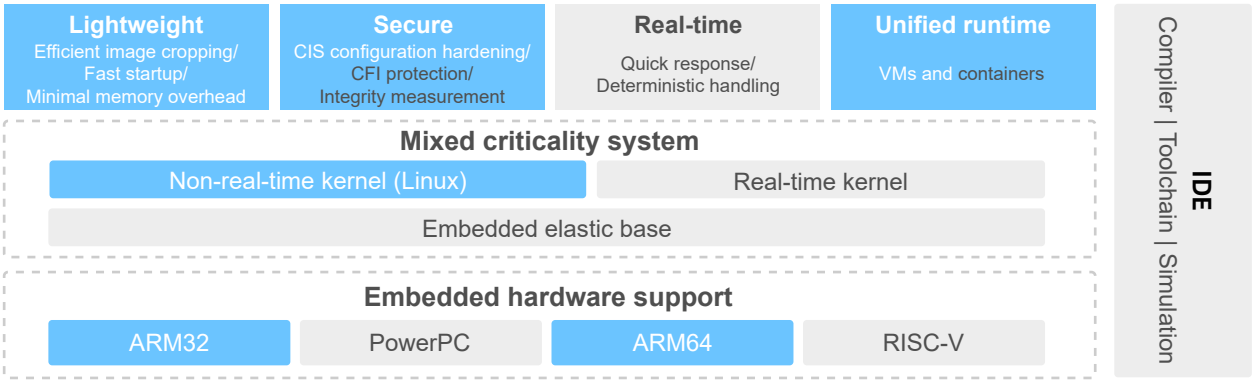
Application Scenarios

Edge computing can be used to shorten the gap between edge and cloud for a wide range of scenarios, such as smart manufacturing, urban transportation, tollway inspection, smart gas stations, medical image recognition, and smart campuses.

Embedded Systems

openEuler 21.09 Embedded is released for embedded systems. Its lightweight design runs secure and lightweight containers that support both ARM32 and ARM64 chip architectures. It is designed to build embedded OS solutions and smooth the communication between ecosystem partners, users, and developers across the openEuler community. In addition, it will improve compatibility of chip architectures (PowerPC and RISC-V) and related capabilities (deterministic latency, industrial middleware, and simulation systems).

Feature Description



openEuler 21.09 Embedded comprises the following features:

- **Lightweight:** The Yocto compilation framework allows you to customize system items into compressed, lightweight models. For example, OS images can be reduced to less than 5 MB, memory overhead to under 15 MB, and startup to under 5s.
- **Security hardening:** Resources such as account passwords and file permissions can be hardened to improve resistance to vulnerabilities. OS security is enabled by default.
- **Lightweight containers:** The lightweight container runtime for embedded scenarios supports standard container images.
- **Multi-architecture support:** The ARM32 and ARM64 chip architectures are supported.

While the 21.09 version is a convenient choice in most deployments, the following features are set to be added to later versions:

- **Deterministic latency:** This processing capability runs on the domain-specific multi-kernel architecture and meets the multi-level latency requirements in the industrial control field.
- **Industry security certifications:** Further development will ensure that later releases are certified with mainstream security certifications, such as IEC61508 and EC62443.

Application Scenarios

Embedded software systems can supercharge performance for multiple fields, such as aerospace, industrial control, telecommunications, automobiles, and healthcare. Enhanced 5G and AI technologies will ensure that embedded systems will be suitable for IoT and edge computing devices.

Distributed Memory (Coming Soon)

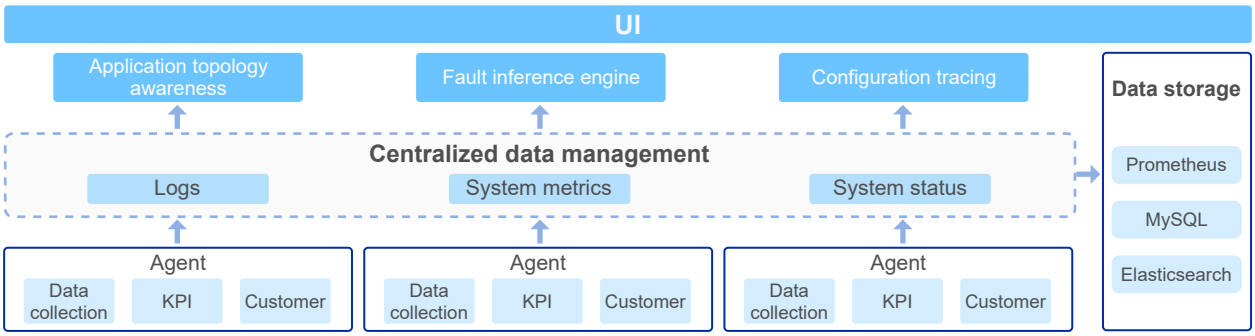
The timeliness requirement of massive data processing pushes applications to grow at scale. However, large-scale distributed cluster applications are complex, and often encounter performance issues such as bottlenecks in the existing computing architecture. The distributed memory suite is a distributed cluster acceleration platform designed for applications. It uses memory data processing and near-data computing technologies to multiply application performance for distributed application scenarios, such as big data, HPC, AI, distributed storage, databases, clouds, and virtualization. This suite can also accelerate applications requiring diversified computing power and for synergy across devices, edge, and cloud.

07 /

Enhanced Features

A-OPS: Intelligent O&M

Big data and machine learning technologies are maturing but this in turn is generating huge amounts of data increasing 2- to 3-fold every year. This shift is driving investments in new, efficient, and intelligent O&M systems that help enterprises reduce costs. Red Hat Insights uses data-driven and intelligent engines to automatically predict, diagnose, and locate faults, helping quickly resolve OS security and performance problems from hours to just minutes.



openEuler A-OPS provides a basic intelligent O&M framework that facilitates premium capabilities such as configuration tracing, application topology awareness, and fault locating, supporting quick troubleshooting and reducing O&M costs.

Feature Description

- **Application topology awareness:** This low-load probe framework based on eBPF provides automatic network topology awareness and detection capabilities at the application level. It helps visualize the network status using the WebUI, helping O&M personnel quickly detect network problems.
- **Configuration tracing:** Configuration changes are common and likely to cause problems during maintenance. A-OPS allows you to manage and configure cluster OS, including specifying the domain. This implements the configuration baseline and comparison functions that help you quickly rectify configuration problems.
- **Fault locating:** A-OPS provides an expert-mode engine to detect system faults in real time and rectify system faults, reducing system downtime and O&M costs.

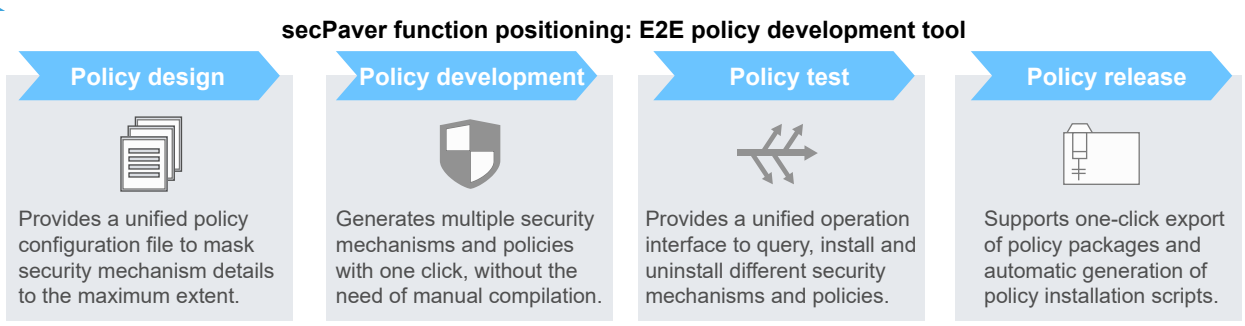
Application Scenarios

A-OPS is a perfect companion for any OS maintenance team. Using the existing A-OPS fault tree or adding fault trees can effectively improve the maintenance efficiency and reduce potential downtime.

secPaver: Security Policy Configuration Tool

secPaver is an SELinux policy development tool used to assist developers in creating application security policies. secPaver abstracts common policy configuration files and encapsulates unified policy operation interfaces to mask mechanism details for policy developers, simplifying policy development and improving efficiency. It can be used to develop and simplify SELinux policies, and can be extended to other security mechanisms, such as AppArmor.

Feature Description



secPaver has the following features dedicated to system policies:

- **Design:** The following command is used to create projects and modify the policy configuration files.
  pav project create *project_name/***opt/policy**
- **Generation:** The following command is used to generate the SELinux policy file based on the policy configuration file.
  pav project build --engine selinux -d/opt/policy*/project_name*
- **Testing:** The following commands can be used to install, query, and uninstall policies.
  pav policy list
  pav policy install/uninstall *policy_name*
- **Release:** The following command can be used to release the policy packages (containing policy files and installation and uninstallation script files).
  pav policy export *policy_name*

Application Scenarios

secPaver is designed to help developers and other configuration personnel to develop SELinux policy files for applications.

Kunpeng GCC

Kunpeng GCC is developed based on the open source GCC 10.3. It is a high-performance compiler for the Kunpeng 920 processor through software and hardware collaboration, memory optimization, SVE, and math library.

- The compiler fully utilizes the hardware features of Kunpeng processors to achieve higher operating efficiency. In the benchmark tests such as SPEC CPU 2017, Kunpeng GCC delivers much better performance than GCC 10.3 of the upstream community.
- What's more, it supports the mcmodel=medium, fp-model, quadruple-precision floating points, and vectorized math library.

Feature Description

- **mcmodel=medium addressing:** Allows symbols (> 4 GB) to be properly accessed, which resolves the error caused by buffer overflow in the ARM architecture.
- **Quadruple precision:** Effectively improves the precision of 128-bit floating point arithmetic over the ARM architecture.
- **Vectorized math library:** Automatically searches for the available vectorized math library in the vectorization phase.
- **SVE optimization:** Significantly improves program running performance for ARM-based machines that support SVE instructions.
- **SLP vectorization optimization:** Analyzes and vectorizes reduction chain groups to improve program performance.
- **Memory layout optimization:** Rearranges the positions of structure members so that frequently accessed structure members are placed in continuous memory space, increasing the cache hit ratio and improving program performance.
- **fp-model precision control:** Controls and refines the precision of floating-point calculations.

Application Scenarios

In the HPC test of Weather Research and Forecasting (WRF) and Microsystems Engineering and Materials Science (NEMO) applications, Kunpeng GCC delivers 10% higher performance than GCC 10.3 of the upstream community. Likewise, during the SPEC CPU 2017 benchmark test, Kunpeng GCC delivers nearly 15% higher performance than GCC 10.3 of the upstream community.

BiSheng JDK

BiSheng JDK is an enhanced JDK developed based on Open Java Development Kit (OpenJDK). It features high performance and high availability and can supercharge production environments in any field or industry scenarios, and can optimize the

performance especially for ARM-based scenarios. BiSheng JDK supports both OpenJDK 8 and 11 versions, and is compatible with Java Platform, Standard Edition (Java SE). BiSheng JDK offers the following advantages:

- **Stable and efficient:** In benchmark tests, such as SPECjbb 2015, BiSheng JDK delivers much better performance than OpenJDK.
- **Converged software and hardware:** BiSheng JDK fully utilizes the hardware features of Kunpeng servers for higher efficiency.
- **Premium security:** BiSheng JDK synchronizes release updates with OpenJDK community editions, performs strict analysis and control, and applies patches to CVE vulnerabilities as and when needed.
- **Open source:** BiSheng JDK provides releases and open source code, and continuously contributes to upstream communities.

Feature Description

- **Java Flight Recorder (JFR):** JFR is an online tool used for collecting diagnostic and profiling data to minimize your performance overheads. For applications that run for a long time, the performance impact is less than 1%. To enable JFR in the production environment, run the **-XX:+FlightRecorder** command. A dump file will be generated for analysis. JFR can work with Java Mission Control (JMC) for better performance analysis visualization.
- **AppCDS (BiSheng JDK 8 only):** AppCDS is an extended version of Class-Data Sharing (CDS). It dumps application classes into JSA files to extend the class usage scope and improves the application startup and loading speeds.
- **Garbage-First garbage collector (G1 GC) (BiSheng JDK 8 only):** The Java Virtual Machine (JVM) can detect decreases in application loads and idle memory of the Java heap, and then automatically reduces the Java heap occupancy to return the idle memory resources to the OS. G1 GC helps reduce overheads in the container scenario where resources are paid for based on subsequent usage.
- **Z Garbage Collector (ZGC) (BiSheng JDK 11 only):** ZGC is a scalable low-latency garbage collector with a maximum GC pause time of 10 milliseconds, a fixed value regardless of heap size. It is compatible with ARM64.
- **NUMA-Aware for the G1 GC (BiSheng JDK 8 and BiSheng JDK 11):** The JVM can make full use of hardware features. During application running, the memory of the local node is preferentially used for object allocation. During garbage collection, memory replication is preferentially performed on the same node to ensure data affinity of applications after garbage collection.
- **New rapid serialization (BiSheng JDK 8 and BiSheng JDK 11):** This function reduces redundant new transmissions and improves the efficiency of serialization and deserialization.
- **KAE Provider (BiSheng JDK 8 only):** By using the provider mechanism, BiSheng JDK 8 supports KAE encryption and decryption of Kunpeng servers, improving the competitive edge of encryption and decryption services on Kunpeng AArch64 servers. The KAE Provider delivers 90% higher encryption and decryption performance than the native JDK.
- **jmap parallel scanning (BiSheng JDK 8 and BiSheng JDK 11):** By default, jmap of OpenJDK performs the Java dump operation using a single thread. However, jmap of BiSheng JDK 8 and BiSheng JDK 11 implements parallel and incremental scans, effectively improving the scan speed in large heap scenarios. Currently, this scan capability has been available in G1 GC, ParallelGC, and CMS.

Application Scenarios

Application scenario 1: Big data applications

BiSheng JDK optimizes memory allocation and reclamation for GC in big data applications, and eliminates redundant memory barriers in JIT code. In related benchmark tests, BiSheng JDK delivers 5% to 20% higher performance than OpenJDK.

Application scenario 2: Common Java applications

BiSheng JDK optimizes the weak memory model of Kunpeng servers to avoid invalid memory barriers. NUMA-Aware based on software and hardware improves the memory access efficiency of applications and fully unleashes application performance. BiSheng JDK enhances Java tools, such as JFR and jmap, to help developers quickly profile performance and locate faults.

08/

Community Governance

Vision

Build an innovative platform by cooperation within the openEuler community; build a unified and open community of openEuler OS to promote the multiple process architecture; and promote a prosperous software and hardware ecosystem.

Community Communication

openEuler contains many projects that are organized into community groups. Communication channels for these groups, including mailing lists, can be found in the relevant README files.

Mailing List

You can start a discussion on an open topic by sending an email to the relevant mailing list.
Visit https://openeuler.org/en/community/mailing-list/ to find a community mailing list. The following steps describe how to subscribe to a mailing list.
You can subscribe to a mailing list by visiting the web page or by email.

Web page

1. Click the list name in https://openeuler.org/en/community/mailing-list/ to go to the subscription page.
2. Enter the subscription email address and click **Subscribe**.
3. Log in to the mailbox and reply to the confirmation email sent from openeuler.org.

After this, you will receive a second email containing "Welcome" from @openeuler.org, indicating that you have successfully subscribed.

Note: If you have not received the "Welcome" email, it means subject form of the reply email is incorrect. In this case, reply to the original email again and include the original subject.

Email

1. Send an email with the title "subscribe" to the subscription address displayed next to the project name on the mailing list. The subscription address must consist of the list address and include the suffix "-join" (see example below).
2. Reply to the confirmation email sent from openeuler.org.
   Take Dev (dev@openeuler.org) as an example. The email is as follows:
   To: dev-join@openeuler.org
   Subject: Subscribe
   Body: NA

Currently, there are two types of mail subjects: announce and discussion. The method of sending a discussion email to a list is the same as that of sending emails to private addresses. It is a good practice, though not mandatory, to add a pair of square brackets and a subject as the prefix to the email subject. An announce email is used only to announce messages or precautions, therefore we cannot submit an issue based on this type.

Note: If you cannot receive any messages in your inbox, check your junk/spam mailbox.

To unsubscribe from a mailing list, perform the following steps:

1. Send an email whose subject is "unsubscribe" to the list of addresses you want to unsubscription from. Generally, you can add a suffix "-leave" after the list address (eg: dev-leave@openeuler.org).
2. Confirm the information in the confirmation email and reply.

After replying the confirmation email, the unsubscription is successful.

Obtaining Help

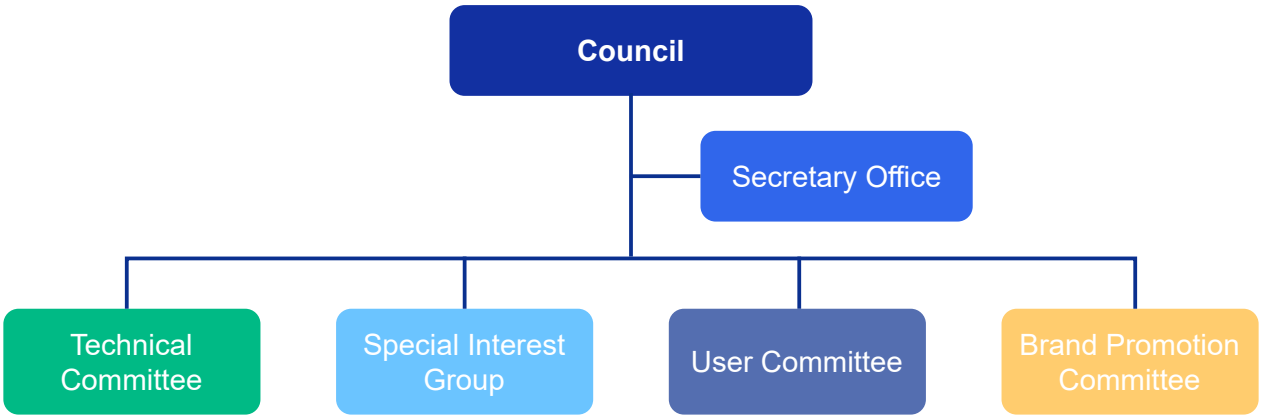
If you encounter any problems during the mailing process, contact the infrastructure support team:

Email: infra@openeuler.org

If you find any bugs related to the mailing list, please submit an issue to the infrastructure team. For details about submitting an issue, see the descriptions below.

Community Governance Organization Structure

The community governance organization guides the development direction of the openEuler community, and comprises the Council, Secretary Office, Technical Committee, User Committee, Brand Promotion Committee, and Special Interest Groups (SIG). Community Communication.



The Council formulates the long-term development plan and guides individual and collective policies for the community. It reviews the work of the user committee and brand promotion committee, and manages the work plan of each committee. The council is also responsible for promoting the openEuler community and related OS distributions to various industries worldwide for wide-scale usage and ecosystem construction.

Secretary Office

Daily work of the Secretary Office under the guidance of the openEuler Council is as follows:

1. Implements Council resolutions, organizes and holds Council meetings, and coordinates with community agencies.
2. Prepares quarterly and annual work reports of the community, and releases the reports after approval from the management board.

Technical Committee

The openEuler Technical Committee (TC) is the technical leader of the openEuler community.

The main responsibilities of the TC are as follows:

1. Makes the final decision on technical matters.
2. Finalizes the vision and direction of community technology development.
3. Establishes, coordinates, and performs other duties for the community SIGs. In addition, resolve the collaboration conflicts (if any) between SIGs, and coaches, reviews, and monitors the daily operation of the SIGs.
4. Implements the daily development work of the community and ensures the high-quality release of openEuler OS versions.
5. Leads innovation of system architecture, kernel, virtualization, cloud native, and security technologies to ensure continuous competitiveness.
6. Guides the community to establish original open-source projects and continuously build the technical influence of the community.

Conference Organization

Official conference (public): The TC holds a public online discussion at 10:00 a.m. (GMT+8) on alternating Wednesdays.

Details on how to join the conference will be released in the mailing list one day in advance.

This conference is open to developers who are interested in the openEuler community.

Special Interest Group

A Special Interest Group (SIG) is a team designated to a domain that holds regular tasks and activities to achieve the delivery objectives. SIGs have transparent rules, and must comply with the openEuler code of conduct. Anyone can join an SIG and contribute to the group projects. You can find all SIGs at: https://openeuler.org/en/sig/sig-list/.Technical Committee

User Committee

The user committee is the organization responsible for communicating with the end users of the openEuler community, and has the following responsibilities:

- Collects technical and product requirements for the openEuler community edition, releases openEuler OS versions, and organizes other open source projects of the openEuler community. In addition, provides feedback to the technical committee and the council, promotes the technical roadmap of the technical committee to meet the requirements of end users, and improves marketspace and long-term planning.
- Organizes best practices of the openEuler community edition, OS release version based on the openEuler community technology, and other open source projects of the openEuler community, and cooperates with the Brand Promotion Committee to publicize the cases with the authorization of end users.

Brand Promotion Committee

The Brand Promotion Committee is responsible for promoting the openEuler and related brands of the community. Its main responsibilities are as follows:

- Promotes the openEuler OS technology and community to improve the influence of the openEuler brand.
- Guides the wide use of the openEuler OS technology to build a global ecosystem.

Contributions

To contribute to the openEuler community, the first step is to select your desired project from the SIG/project list of openEuler. Once you are subscribed, you can attend SIG/project meetings and subscribe to the mailing list. An SIG or a project is usually composed of a series of help-wanted issues that you can work on.

Signing CLA

You must sign a Contributor License Agreement (CLA) before you can contribute to the community.

Community Code of Conduct

The openEuler community complies with the code of conduct stipulated in the *Contributor Convention V1.4. For details, see the V1.4 version.*

To report explicit or inappropriate behaviors, you can contact the openEuler Technical Committee at: tc@openeuler.org.

Commitment of Contributors

To maintain an open and professional environment, the openEuler community will not tolerate harassment, bullying, aggression or bad behavior of any kind. These include but are not limited to the basis of the following categories: race, color, ethic or national origin; age; nationality; sex, gender, or sexual orientation; bodily shape; physical or mental disabilities; or level of experience, education, or social status.

Our Principles

Actions that contribute to the creation of a positive environment include but are not limited to:

- Friendly and inclusive choice of words
- Respectful of diverse viewpoints and experiences
- Open to criticism and suggestions
- Prioritize actions in the interests of the community

The openEuler community prohibits harassment, explicit, and any other unacceptable behavior, online or otherwise, including but are not limited to:

- Pornographic, violent, sexual, or other explicit comments, images, and videos
- Comments that are deemed disruptive, inciting hate, rumors, insults, or derogatory statements, and attacks on personal and political beliefs
- Public or private harassment (sexual, bullying or otherwise) to colleagues or subordinates
- Release other private information, such as addresses and email addresses, without explicit prior consent
- Other misconducts that can be justified as a violation of professional ethics

Our Commitment

Maintainers of the community and related projects hold the right to interpret "proper conduct", and fully and impartially correct misconduct when needed. Maintainers hold the right to remove, edit, and reject any and all comments, code, or other issues that violate the code of conduct. Community project maintainers may temporarily or permanently ban any participant whom they consider to be improper, threatening, offensive, or harmful.

Application Scope

The code of conduct applies to everyone, everywhere, who participate in the openEuler community. The code of conduct applies to anyone who represents the community, as well as the public platforms where the person joins.

Representing the community includes but is not limited to: using the official email of the community, posting messages on official media accounts, and participating in online or offline activities as a designated representative.

The code of conduct on behalf of the community may further be defined and interpreted by the Secretary Office and submitted to the Council for approval and issuance.

Supervision and Investigation

To report abuse, harassment and misconduct found within each project team, send emails to the Secretary Office at: secretary@openeuler.io.

The Secretary Office will consider and investigate all complaints and respond accordingly. The project team is obligated to keep whistleblower information confidential.

Implementation

Send an email to tc@openeuler.org to report abuse, harassment, and misconduct to the project team.

The maintenance team will review and investigate all complaints and respond to them as necessary. The project team is obligated to keep whistleblower information confidential. Specific implementation guidelines may be issued separately later.

Anyone who fails to comply with or execute the code of conduct may be temporarily or permanently disqualified from participating in the project, until the matter is resolved with the project owner or other members.

Community Contributions

We work hard to ensure that the documents and software found within the community are best-in-class. However, we are aware that documents can be improved (such as the one you are reading), code needs to be reviewed, functions or variables can be reconstructed or commented out, and test cases can be supplemented and optimized. We will help you understand the organization of openEuler SIGs and walk you through your first contribution.

Introduction to the SIG

SIGs, short for Special Interest Groups, are organizations within the openEuler community that are designed to better manage and improve the work process.

- SIGs are open to anyone to join and contribute.
- An SIG is established for one or more specific technology projects. SIG members promote the output of deliverables and strive to make the deliverables a part of the openEuler community.

- Core members of the SIG govern the group. Core members are based on the accumulated experience and contributions.
- Each SIG has one or more projects on Gitee, with each project comprising one or more repositories. The SIG deliverables are stored in these repositories.
- As a member, you can submit issues in the repository of a specific SIG, participate in discussions, resolve issues, and participate in reviews.
  Members can communicate with other members in the SIG through the mailing list and during video conference.

Finding the most relevant SIG of your domain can help you obtain a quicker response of the issues submitted. There are two methods to find the SIG.

- **Method 1:** To find your desired SIG or project, you can view the list of all SIGs established in the openEuler community. The following information about the SIG is also provided:
  - Projects under the SIG and subsequent the repository addresses
  - Communication methods within the SIG, including mailing list and video conference
  - Contact information of maintainer
- **Method 2:** If you know the project name, you can perform fuzzy search in the repository list on the openEuler website to quickly locate the home page address of the target project. Generally, you can find the SIG information, communication methods, group member usernames, and contact information in the **README.md** file of the project home page address.

If you cannot locate the SIG you are interested in using either of the preceding methods, you can send an email to community@openeuler.org for help. You are advised to use [Development Process Question] as the subject of the email, and include details of the SIG or project that you are looking for in the body.

How to Contribute

Review an Issue

- **Find the issue list:** On the toolbar of the home page (Repository of the project on Gitee) of the project that you are interested in, click **Issues** to find the issue list of the SIG For example, the issue list address of the community team is https://gitee.com/openeuler/community/issues.
- **Review an issue:** If you want to handle one of the issues, you can assign it to yourself. In the comment box, enter /assign or /assign @[yourself] in the comment box. The community bot will assign the issue to your account, and your name will be displayed in the owner list.
- **Participate in discussion of an issue:** Each issue may have been discussed by participants. If you are interested in the issue, you can comment on it in the comment box.

Raise Questions or Suggestions

- **Raise a problem:** If you find a problem or bug, you can raise the problem to the admin by submitting an issue on the issue list repository of a specific project. When submitting an issue, comply with the issue submission rules.
- **Feedback:** You can share your opinions or suggestions in the SIG by submitting an issue, which is a public message that is open to everyone. To attract more attention, you can also attach the issue link to the email and send it to everyone through the mailing list.

Set Up the Development Environment

1. Install openEuler.
2. Prepare the development environment.
3. Download and configure software packages.

For details, see Appendix 1.

Participate in Coding

- Understand the SIG and development precautions in the project
  The coding language, environment, and coding conventions may differ between each SIG project. If you want to understand and participate in coding, we provide contributor guides tailored to each project. The guide is generally found in the **CONTRIBUTING.md** file on the SIG homepage, alternatively, you can directly find the file in **README.md** of the project. In addition to these documents, the SIG may also provide other guidance information located in the specific community directory of the SIG or its projects. If you do not find relevant information or have questions about related information, you can submit an issue in the SIG or send questions to the mailing list of the SIG to which the project belongs. If you do not receive a response, you can ask community@openeuler.org for help.
- Download code and pull a branch
  To contribute code, you need to learn how to download code from Gitee and incorporate code appealing for a pull request (PR). The method of using the hosting platform is similar to that of GitHub. If you have used GitHub before, you can roughly understand or even skip this chapter.
- Modification, construction, and local verification
  After completing the modification on the local branch, perform the construction and local verification.
- Submit a PR
  When you submit a PR, it means that you are already contributing code to the community.
- Add a Software Package
  After a software package is added to Gitee, openEuler automatically creates a repository with the same name on **openEuler:Factory** on OBS. In this way, when the code is submitted to the created Gitee repository, the code compilation is automatically checked.

Code Review

openEuler is an open community. We hope that all participants in the community can become active reviewers.

To make submissions easier to accept, all contributors must:

- Comply with the coding conventions of the SIG, if any.
- Prepare complete submission information.
- If you submit a large chunk of code at one time, you are advised to separate the large content into a series of logically small sets. Submitting the content separately will make it easier for reviewers to understand your ideas.
- Label the PR with the appropriate SIG and monitor labels. This ensures the community bot will help you better complete the entire PR process.
  For reviewers, it is strongly recommended that you follow the code of conduct and respect each other for better collaboration. The article The Gentle Art Of Patch Review is a good resource for key review points that code reviewers are expected to understand. Among which, they should encourage new contributors to actively participate in the program, instead of discouraging new contributors by giving overly critical comments. In particular, pay attention to the following points during review:
- Is the idea behind the contribution sound?
- Does the contribution feature correct formatting?
- Will the contribution need further modification?

Note: If your PR does not draw enough attention, you can seek help from the mailing list of the SIG or send email to dev@openeuler.org.

Select the Community Component Package

Creating an RPM package, also called packaging, refers to the task of compiling and binding software and metadata, such as the full name of the software, description, and dependency list required for normal running. This is to allow users to install, remove, or upgrade the corresponding software using the package management tool.

- Packaging guide: openEuler standardizes a variety of open source projects into a coherent system. Therefore, a packaging guide is drafted to standardize RPM development.
- openEuler complies with Linux Standard Base (LSB) to reduce the differences between distributions. openEuler also complies with the Linux Filesystem Hierarchy Standard (FHS). This standard is a reference on how to manage the Linux file system hierarchy.
- In addition to following these general rules that are followed by Linux distributions, this document standardizes the actual details of packaging for the openEuler community version.

Disclosure of Community Security Issues

- Security handling process
- Security disclosure information

For details about the security handling process and security disclosure information, see Appendix 2.

09/ Copyright Description

10/ Trademarks

11/ Appendixes

Appendix 1: Setting Up the Development Environment

Environment Preparation	Address
Downloading and Installing openEuler	https://openeuler.org/en/download/
Preparing the Development Environment	https://gitee.com/openeuler/community/blob/master/en/contributors/prepare-environment.md
Building a Software Package	https://gitee.com/openeuler/community/blob/master/en/contributors/package-install.md

Appendix 2: Security Handling Process and Security Disclosure Information

Disclosure of Community Security Issues	Address
Security Handling Process	https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-process.md
Security Disclosure Information	https://gitee.com/openeuler/community/blob/master/zh/security-committee/security-disclosure.md


OpenEuler