

**openGauss**  
**2.1.0**

# Release Notes

<b>Issue</b>	01
<b>Date</b>	2021-09-30



**Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## **Huawei Technologies Co., Ltd.**

Address: Huawei Industrial Base  
Bantian, Longgang  
Shenzhen 518129  
People's Republic of China

Website: <https://www.huawei.com>

Email: [support@huawei.com](mailto:support@huawei.com)

---

# Contents

---

1 User Notice..... 1

2 Versions..... 2

3 Features..... 5

4 Important Notes..... 9

5 Known Issues..... 10

6 Common Vulnerabilities and Exposures (CVEs)..... 11

7 Source Code..... 12

8 Contribution..... 14

9 Acknowledgment..... 15

# 1 User Notice

---

openGauss is an open-source, secure, and reliable relational OLTP database with ultimate performance. It is released with the Mulan PSL v2 protocol, allowing users to copy, use, modify, and distribute the source code.

The version number of openGauss is named in *X.Y.Z* format. *X* indicates the version for architecture changes, *Y* indicates the version released every year, and *Z* indicates the patch version. Generally, a *Y* version is released every year. A new *X* version is released for major architecture or feature changes. The preliminary openGauss lifecycle plan is three years.

# 2 Versions

---

openGauss 2.1.0 is the preview version released by openGauss. The lifecycle of this version is half a year. 2.1.0 is compatible with the earlier versions. Main functions are as follows:

## Inherited Functions

- SQL standard syntax, UPSERT, data type, XML type, table, temporary table, global temporary table, foreign table, view, materialized view, index, foreign key, Generalized Inverted Index (GIN), sequence, function, trigger, ROWNUM, and MEDIAN aggregate function.
- Stored procedure, commit/rollback in stored procedure, omission of parameter parentheses () from the stored procedure or function calling, stored procedure debugging, and autonomous transaction.
- Security features such as authentication, permission management, network communication security, database audit, and encrypted database.
- Primary/standby nodes, cascaded standby node, logical replication, ultimate recovery time objective (RTO), and high availability (HA) of the standby node scale-out.
- Range partitioning, global partitioned indexes, LIST partitioning, HASH partitioning, and automatically extended partition based on range partitions.
- Full physical backup, logical backup, standby node backup, incremental backup and restoration, and point-in-time recovery (PITR).
- Memory-optimized table (MOT), NUMA-aware high-performance capability, and parallel query.
- Container-based deployment, IPv6, and PostGIS plug-in.
- AI capabilities: parameter self-tuning, slow SQL discovery, AI query time forecasting, database indicator collection, forecasting, and exception monitoring, and DeepSQL AI algorithms in the library.
- Delayed standby node replay, logical replication of standby node, standby Xlog archiving, standby node I/O write amplification optimization, gray upgrade, scale-out tool optimization, database running indicators added to a workload diagnosis report (WDR), and enhanced intelligent index recommendation.

## New Features

- The stored procedure compatibility is enhanced.
- The SQL engine capability is enhanced.
- The Ustore storage engine is supported.
- Segment-page storage is supported.
- High availability is based on the Paxos distributed consistency protocol.
- AI4DB and DB4AI competitiveness is continuously built.
- The log framework and error codes are modified.
- JDBC client load is balanced and read and write are isolated.
- The CMake script compilation is supported.
- The column-store table supports the primary key constraint and unique key constraint.
- The jsonb data type is supported.
- Automatic elimination of unique SQL statements is supported.
- The UCE fault detection is supported.
- The GB18030 character set is supported.
- The standby server catch is optimized.
- The client tool gsql supports automatic supplement of the readline command.
- The dynamic data masking is supported.
- The State Cryptography Administration (SCA) algorithms are supported.
- The tamper-proof ledger database is supported.
- The built-in role and permission management mechanism is supported.
- The transparent encryption is supported.
- The fully-encrypted database is enhanced.
- The dblink is supported.
- The Ubuntu system is supported.
- The hash index is supported.
- UPSERT supports subqueries.
- The MIN/MAX function supports the IP address type.
- The array\_remove, array\_replace, first, and last functions are added.
- The Data Studio client tool adapts the kernel features.

#### Modified Defects

- **I435UP** An error is reported when the EXPLAIN statement is executed.
- **I44QS6** When the **select get\_local\_active\_session() limit 1 ;** function is executed, the database breaks down.
- **I4566H** After UPDATE GLOBAL INDEX is performed on a partition of a partitioned table, the query result is inconsistent with the master version.
- **I45822** An error occurs when the GPC global plan cache information is queried in the global temporary table.
- **I442TY** Failed to recover to the timestamp specified by PITR.
- **I45T7A** Remote backup is abnormal when the database is installed in environment variable separation mode.

- **I464G5** Failed to use **gs\_ctl build** to rebuild a specified non-instance directory on a standby node. The error information is inconsistent.
- **I45TTB** The foreign table is successfully created for the file type that is not supported by file\_fdw, but no error is reported.
- **I491CN** When the subnet mask of the network address of the cidr type is 32, an error is reported when the MAX function is called.
- **I496VN** After a large number of Xlogs are stacked on the standby node, the archiving address is corrected. As a result, the archiving fails.
- **I49HRV** When the standby node archiving is enabled, the standby node archiving is slow. After the switchover, the new primary node is abnormal.
- **I492W4** When operations related to the mysql\_fdw and oracle\_fdw foreign tables are performed on the database installed using the OM, a core dump occurs in the database.
- **I498QT** In the maximum availability mode, when the synchronous standby parameter is ANY2 and the primary server is under continuous pressure, running the **kill-9** command to stop one synchronous standby server causes transaction congestion on the primary server for 2s.
- **I49L15** Two standby nodes are enabled for archiving. After one node is scaled in and out, the archiving of the other node is abnormal.
- **I43MTG** The developer guide does not contain information related to new functions.
- **I42YW8** The UPSERT subquery information is not supplemented.
- **I45WDH** file\_fdw does not support the fixed format. The related description needs to be deleted from the developer guide.
- **I484J0** The **gs\_initdb -T** parameter is not verified, and the value is incorrect after being set according to the guide.
- **I471CS** When **pgxc\_node\_name** contains hyphens (-), the database exits abnormally. If residual temporary tables are not cleared, automatic clearance and vacuum cannot be performed.
- **I40QM1** When gs\_basebackup is executed, an exception occurs on the standby node. As a result, the gs\_basebackup process is blocked and cannot exit.
- **I3RTQK** The standby node fails to be backed up using gs\_basebackup, and the message "could not fetch mot checkpoint info; status:7" is displayed.

openGauss is a standalone database. To use openGauss in formal commercial projects, you need to build complete toolchain capabilities such as database monitoring and primary/standby switchover.

# 3 Features

---

- **Standard SQL support**  
Supports SQL-92, SQL-99, SQL:2003, and SQL:2011 standards, GBK and UTF-8 character sets, SQL standard functions and analytic functions, and stored procedures.
- **Database storage management**  
Supports tablespaces where different tables can be stored in different locations.
- **Primary/standby deployment**  
Supports the ACID properties, single-node fault recovery, primary/standby data synchronization, and primary/standby switchover.
- **Application programming interface (API)**  
Supports standard JDBC 4.0 and ODBC 3.5 features.
- **Management tools**  
Provides the installation and deployment tool, instance start and stop tool, backup and restoration tool, scale-out and scale-in tool, and upgrade tool.
- **Security management**  
Supports SSL network connections, user permission management, password management, security auditing, and other functions, to ensure data security at the management, application, system, and network layers.
- **AI**  
Supports parameter self-tuning, slow SQL discovery, single query index recommendation, virtual index, workload index recommendation, database metric collection, forecasting, and exception monitoring; compatible with the MADlib ecosystem and supports high-performance AI algorithms in the library.

## New Features

Compared with 2.0.1, openGauss 2.1.0 Preview has the following new features:

- **The stored procedure compatibility is enhanced.**  
Stored procedures support packages, autonomous transactions, and PL/PGSQL single-step debugging.



- The SQL engine capability is enhanced.
  - The JDBC driver supports the getNstring API.
  - GUC parameters are initialized by resource group.
  - The Group Planner is reconstructed.  
exec\_type is pre-processed, is pathkeys rewritten, and query\_planner is pathized.
  - The list and hash partitioned tables support data exchange.
  - Partitioned table execution is optimized.  
The operator initialization of the partitioned table obtains the pg\_partition part for cache optimization, reducing the query delay of a single partitioned table and improving the performance.
  - Planhint is enhanced.  
The session-level optimizer parameters can be set by using planhint. The specified subquery can be not expanded. GPC can be disabled for a single query.
- The Ustore storage engine is supported.  
The Ustore storage engine, also called the in-place update storage engine, is a new storage mode added to the openGauss Kernel. The row storage engine used by the earlier openGauss Kernel versions is in append update mode. The append update has good performance in addition, deletion, and HOT (Heap Only Tuple) update (that is, update on the same page) in the service. However, in a non-HOT UPDATE scenario across data pages, garbage collection is not efficient. The Ustore storage engine can solve this problem.
- Segment-page storage is supported.  
The segment-page storage manages database file storage by segment and stores data of multiple tables in the same file, thereby resolving a problem that a large quantity of files cause relatively high pressure on a disk and a file system.
- High availability is based on the Paxos distributed consistency protocol.  
Log replication and primary selection framework are based on the Paxos distributed consistency protocol. Nodes can be added and deleted online, the leader capability can be transferred online, the primary node can be selected based on the priority, and the majority policy is supported. Various node roles are supported with an efficient flow control algorithm.
- AI4DB and DB4AI competitiveness is continuously built.
  - Enhanced openGauss AI capability: provides the root cause diagnosis capability for a single slow SQL statement and enhances capabilities such as intelligent index recommendation and time series forecasting.
  - DB4AI capability: provides the fenced UDF and native DB4AI algorithm capabilities, including the execution plan, operators, and SQL syntax in the database.
- The log framework and error codes are modified.  
The kernel log specification framework is optimized and the existing logs are reconstructed and optimized.
- JDBC client load is balanced and read and write are isolated.

- The IP addresses and port numbers of multiple nodes on the client are configured to adapt to HA switchover between multiple AZs and remote DR switchover.
- The connection-level read/write isolation configuration is supported.
- Always connecting to the read-only node (not the current primary node) and adapting to primary/standby switchover are supported.
- Preferentially connecting to read-only nodes is supported.
- Multiple read-only nodes are evenly distributed.
- Users are allowed to query the connection string configuration of each application on the database server to check for non-compliant connection configurations.
- The CMake script compilation is supported.
- The column-store table supports the primary key constraint and unique key constraint.
- The jsonb data type is supported.
- Automatic elimination of unique SQL statements is supported.  
When the number of unique SQL statements reaches the maximum, previous unique SQL statements are automatically deleted so that new unique SQL information can be recorded.
- The UCE fault detection is supported.  
The UCE errors are detected and the corresponding logs are displayed.
- The GB18030 character set is supported.
- The standby server catch is optimized.  
The GUC parameter **catchup2normal\_wait\_time** is added to control the maximum duration for blocking the primary server during data chasing on the synchronous standby server after the maximum availability mode is enabled.
- The client tool gsql supports automatic supplement of the readline command.
- The dynamic data masking is supported.  
Users are allowed to customize data masking rules by loading UDFs.
- The State Cryptography Administration (SCA) algorithms are supported.  
The SM3 encryption algorithm is supported by ODBC and JDBC, and APIs for SM4 algorithm encryption and decryption are provided.
- The tamper-proof ledger database is supported.  
The tamper-proof ledger database is implemented, the tamper-proof user history table and global block table are added, and operations on tables in the specified anti-tamper schema are audited.
- The built-in role and permission management mechanism is supported.  
High-privilege operations supported by the system are managed using built-in roles. Roles can be created for users as required.
- The transparent encryption is supported.  
The encryption is transparent to the application layer. Compared with non-encryption, the performance deterioration does not exceed 10%.
- The fully-encrypted database is enhanced.  
The encrypted equality query supports JDBC, stored procedures, and functions.

- The dblink is supported.  
The dblink plug-in is supported to implement interaction between tables in different databases.
- The Ubuntu system is supported.  
The database can be installed in the Ubuntu system through source code build or OM installation, and the database can be connected through the client driver.
- The hash index is supported.  
The performance of long-index column equality query is better than that of B-tree. Lock management is optimized to provide higher concurrency. Xlog protection is provided to prevent data loss.
- UPSERT supports subqueries.  
The UPSERT statement supports subqueries.
- The MIN/MAX function supports the IP address type.  
The MIN/MAX function can compare cidr and inet network address types.
- The array\_remove, array\_replace, first, and last functions are added.  
The array\_remove, array\_replace, first, and last functions are added.
- Data Studio features  
Data Studio supports multiple features of the openGauss kernel. The details are as follows:
  - Partition tables of the HASH, LIST, and INTERVAL types can be created.
  - Partition data can be viewed by partition column in a partitioned table.
  - User roles can be revoked.
  - The PRIMARY KEY and UNIQUE constraints on column-store tables are supported.

# 4 Important Notes

---

- For details about technical specifications, see section "Technical Specifications" in *Technical White Paper*.
- Currently, openGauss supports a maximum of eight standby nodes. If one primary node and multiple standby nodes are used and the primary node is faulty, promote a standby node with more logs to primary, preventing other standby nodes from being rebuilt.
- You are advised to deploy one primary node and two standby nodes to ensure database reliability and availability.

# 5 Known Issues

---

- openGauss cannot monitor storage space, file permission, and slow disks. When the storage space is full or the file permission is abnormal, the database exits and the corresponding information is recorded in logs. On a slow disk, the response to database operations is slower than usual.
- openGauss has only database kernel capabilities and does not support primary/standby monitoring or switchovers. If the primary and standby nodes are disconnected and their logs do not match, the HA cluster needs to be rebuilt.
- By default, the OM starts the primary and standby databases based on the initial primary/standby relationship. If an application performs a primary/standby switchover on the databases, the application needs to notify the OM of the new primary/standby relationship to prevent the HA cluster from being rebuilt due to incorrect primary/standby relationship.
- The read-only mode of the standby node and cascaded standby node is incompatible with the ultimate RTO feature. If the ultimate RTO feature is enabled, disable the read-only mode of the standby node and cascaded standby node.
- Memory Optimized Tables (MOTs) are incompatible with the incremental checkpoint feature. If MOTs are used, disable the incremental checkpoint function.
- LLVM does not support the ARM architecture. When the MOT TPC-C is imported, an LLVM error is reported. To avoid this problem, disable the JIT function using the **enable\_mot\_codegen** parameter. You can configure **force\_mot\_pseudo\_codegen= true** to reduce the impact on TPC-C test performance when the JIT function is disabled.

# 6 Common Vulnerabilities and Exposures (CVEs)

---

For details about CVE involved in this version, see the following CVE list.

CVE Number
<a href="#">CVE-2021-3541</a>
<a href="#">CVE-2021-3517</a>
<a href="#">CVE-2021-3518</a>
<a href="#">CVE-2021-3537</a>
<a href="#">CVE-2021-3449</a>
<a href="#">CVE-2021-24032</a>
<a href="#">CVE-2020-8277</a>

# 7 Source Code

---

openGauss contains the database server code repository, open-source software code repository, driver code repositories (JDBC, ODBC, Go, and Python), database OM tool code repository, distributed solution code repository, code repository for migrating Oracle to openGauss, code repository for migrating MySQL to openGauss, and backup and restoration code repository. The code repository addresses are as follows:

- Database server code repository: <https://gitee.com/opengauss/openGauss-server>
- Open-source software code repository: [https://gitee.com/opengauss/openGauss-third\\_party](https://gitee.com/opengauss/openGauss-third_party)
- JDBC driver code repository: <https://gitee.com/opengauss/openGauss-connector-jdbc>
- ODBC driver code repository: <https://gitee.com/opengauss/openGauss-connector-odbc>
- Go driver code repository: <https://gitee.com/opengauss/openGauss-connector-go-pq>
- Python driver code repository: <https://gitee.com/opengauss/openGauss-connector-python-psycopg2>
- Database OM tool code repository: <https://gitee.com/opengauss/openGauss-OM>
- Distributed solution code repository: <https://gitee.com/opengauss/openGauss-distributed-solutions>
- Code repository for migrating Oracle to openGauss: <https://gitee.com/opengauss/openGauss-tools-onlineMigration>
- Code repository for migrating Oracle to openGauss: <https://gitee.com/opengauss/openGauss-tools-ora2og>
- Code repository for migrating MySQL to openGauss in real time: <https://gitee.com/opengauss/openGauss-tools-chameleon>
- Database consistency verification code repository: <https://gitee.com/opengauss/openGauss-tools-datachecker>
- Migration evaluation report code repository: <https://gitee.com/opengauss/openGauss-tools-migrationAssessmentReport>

- Backup and restoration code repository: <https://gitee.com/opengauss/openGauss-tools-backup>



# 8 Contribution

---

## Participating in Contribution

As an openGauss user, you can contribute to the openGauss community in multiple ways. For details about how to contribute to the community, see [Contribution](#). Here, some methods are listed for reference.

## Special Interest Groups (SIGs)

openGauss brings together people of common interest to form different 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 [Special Interest Group \(SIG\)](#).

## Mail List and Issues

You are welcome to actively help users solve problems raised in the [mail list](#) and issues (including [code repository issues](#)). In addition, you can submit an issue. All these will help the openGauss community 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 encountered when obtaining software or documents and difficulties encountered when using the system. You are welcome to pay attention to and help us improve the documentation module of the openGauss community.

## Internet Relay Chat (IRC)

openGauss has also opened an IRC channel as an additional channel to provide community support and interaction. For details, see [openGauss IRC](#).

# 9 Acknowledgment

---

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