

Databases

Decision Guide

Issue	01
Date	2026-08-05



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2026. 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 Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property 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 Cloud 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 Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 Databases Decision Guide..... 1

1 Databases Decision Guide

Overview

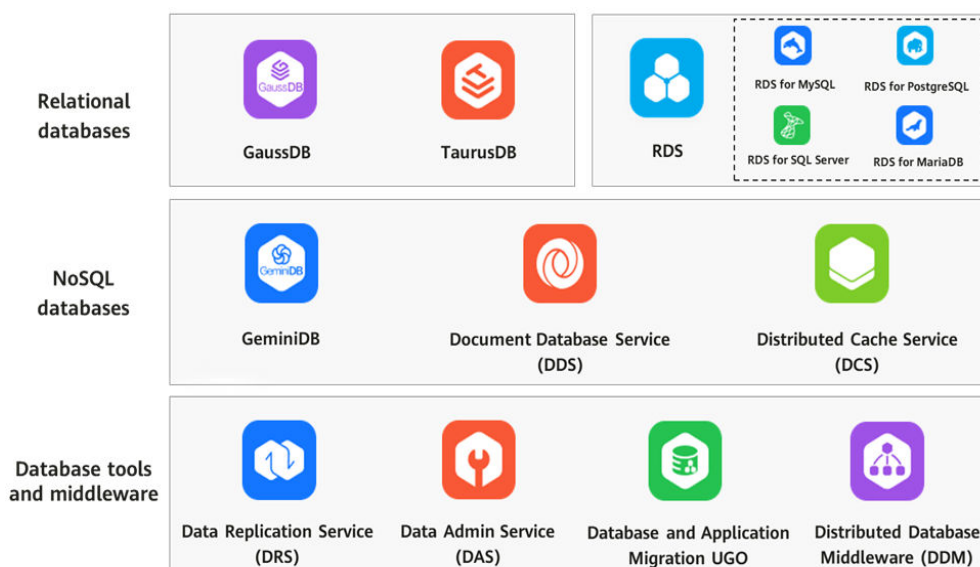
In the era of cloud-native and intelligent applications, databases have become a core component of the data infrastructure that supports enterprise digital transformation. Beyond storing, managing, and providing access to business data, they directly affect application throughput, responsiveness, scalability, security, and reliability. As enterprise workloads expand from traditional online transactions to real-time services, IoT connectivity, knowledge-driven applications, and intelligent operations, databases have evolved beyond standalone data storage engines. They now provide essential platform capabilities that support diverse data models, complex workloads, and continuous business innovation.

Huawei Cloud offers a broad portfolio of database services and supporting tools, including relational, key-value, document, in-memory, time-series, and wide-column databases. Core services such as GaussDB, Relational Database Service (RDS), GeminiDB, and Document Database Service (DDS) support development and testing, cloud adoption, and mission-critical production, helping enterprises build a modern, future-ready data foundation.

Selecting the right database service, or combination of services, requires evaluating the business scenario, data model, workload characteristics, performance targets, reliability requirements, and long-term architecture roadmap. This guide explains the key selection criteria, recommended scenarios for each service, and a clear path for database selection, architecture planning, and migration. It helps you select services for online transaction processing (OLTP) workloads and, when needed, extend their use to broader data processing and AI use cases.

The Landscape

The Huawei Cloud database service portfolio for OLTP workloads consists of three categories: relational databases, non-relational (NoSQL) databases, and database tools and middleware.

Figure 1-1 Database service portfolio

Relational Databases

Relational databases organize data into rows and columns and use relationships between tables to support data processing and analysis. The Huawei Cloud relational database portfolio includes GaussDB, TaurusDB, and RDS.

- **GaussDB** is Huawei's proprietary, enterprise-grade database that supports centralized and distributed deployments. It delivers high availability with a recovery point objective (RPO) of zero and a recovery time objective (RTO) of less than 10 seconds. A distributed deployment can scale to as many as 1,000 nodes. GaussDB is designed for mission-critical systems in financial services, telecommunications, government, and enterprise sectors. It is highly compatible with Oracle SQL syntax and the broader Oracle application ecosystem.
- **TaurusDB** is Huawei's proprietary, enterprise-grade, cloud-native database that delivers high performance and elastic scalability while providing full compatibility with MySQL. It is available in Enterprise Edition. TaurusDB Enterprise Edition uses Data Function Virtualization (DFV) storage and a decoupled storage and compute architecture. It supports up to 128 TB of storage and failover within seconds. This edition is suitable for performance-intensive workloads in financial services, gaming, and e-commerce.
- **RDS** is a cloud-based database service that is reliable, scalable, easy to manage, and immediately ready for use. It supports four database engines: MySQL, PostgreSQL, SQL Server, and MariaDB.
 - RDS for MySQL is fully compatible with native MySQL. It provides reliable operation, high performance, intelligent O&M, advanced security protection, rapid deployment, and elastic scaling.
 - RDS for PostgreSQL is fully compatible with native PostgreSQL and is secure, reliable, and easy to use and manage.
 - RDS for SQL Server is reliable, secure, scalable, cost-effective, and easy to manage. Its high-availability architecture supports recovery within seconds, data protection, and flexible backup options.

- RDS for MariaDB is highly compatible with MySQL. It is a powerful, high-performance, secure, and reliable database management system suitable for a wide range of applications.

NoSQL Databases

NoSQL databases use a range of data models to store, access, and manage data, including key-value, document, cache, in-memory, time-series, and wide-column models. By relaxing certain consistency constraints, some NoSQL databases can be optimized for applications that require massive data capacity, low latency, or flexible data models. The Huawei Cloud NoSQL database portfolio includes GeminiDB, Document Database Service (DDS), and Distributed Cache Service (DCS).

- **GeminiDB** is a distributed, multi-model NoSQL database service built on a decoupled storage and compute architecture. It combines high performance, availability, reliability, security, and elastic scalability with one-click deployment, backup and restoration, monitoring, and alarm reporting. GeminiDB supports mainstream NoSQL database engines, including Redis, DynamoDB, Cassandra, HBase, InfluxDB, and MongoDB. It delivers high I/O performance at low cost and is suitable for use cases in IoT, meteorology, internet services, and gaming.
- **Document Database Service (DDS)** is a distributed NoSQL database service compatible with the MongoDB protocol. DDS provides robust security, high availability, high reliability, elastic scaling, and ease of use. It supports one-click deployment, online scaling, disaster recovery, backup and restoration, monitoring, and alarm reporting. DDS can also integrate with third-party components that support the MongoDB protocol, including business intelligence (BI) tools and Spark.
- **Distributed Cache Service (DCS)** is a Redis-compatible managed service for cloud caching scenarios. DCS provides a high-speed in-memory data engine and a reliable, scalable, ready-to-use, and easy-to-manage caching service. It is designed for highly concurrent workloads that require rapid data access.

Database Tools & Middleware

Huawei Cloud also provides database tools and middleware, including Data Replication Service (DRS), Data Admin Service (DAS), Database and Application Migration UGO (UGO), and Distributed Database Middleware (DDM).

- **Data Replication Service (DRS)** is a cloud service used for online database migration and real-time synchronization. Designed for cloud environments, DRS applies least-privilege access controls and uses an architecture focused on reliability to reduce the complexity and cost of data migration between databases. It supports database migration and synchronization across a wide range of scenarios.
- **Data Admin Service (DAS)** is a web-based service for connecting to and managing databases on Huawei Cloud.
 - The **Intelligent O&M** module provides a one-stop management platform for cloud database development, O&M, and intelligent diagnosis.
 - The **Development Tool** module makes database management user-friendly, secure, and intelligent. It allows you to perform basic SQL

operations, advanced database management, and intelligent O&M through the web console.

- **Database and Application Migration UGO (UGO)** is a managed service for heterogeneous database schema migration. UGO automatically converts source-database DDL, DML, and DCL statements, as well as SQL in business applications, into syntax supported by GaussDB or RDS. With pre-migration evaluation, schema migration, and automated syntax conversion, UGO identifies adaptation work early, improves conversion rates, and reduces migration costs.

For heterogeneous migrations, UGO converts and migrates database schemas, while DRS migrates data online. Together, they provide an end-to-end migration solution. The combined solution automates migration from mainstream commercial databases to Huawei Cloud, simplifying cloud adoption and database platform migration.

- **Distributed Database Middleware (DDM)** is a MySQL-compatible, distributed middleware service designed for relational databases. Built on RDS for MySQL instances, DDM provides database and table sharding, read/write splitting, and elastic scaling to overcome the capacity and performance limits of standalone MySQL databases.

Identifying Use Cases

This section describes typical use cases for Huawei Cloud database services.

Internet, E-commerce, and SaaS platforms

- Core OLTP workloads:
 - TaurusDB is recommended as the primary database for internet applications that require high concurrency and elastic capacity.
 - RDS for MySQL is suitable for small- and medium-sized workloads and for migrating self-managed MySQL databases to the cloud with minimal application changes.
- Caching and high-speed key-value access:
 - Use DCS as the caching layer.
 - Use GeminiDB Redis API as the primary key-value database to avoid the complexity of writing data to both a cache and a database.
- Big data analysis and processing (MapReduce):
 - DDS can support document-oriented data analysis requirements. You can submit queries or scripts and distribute requests across DDS nodes for processing.
 - Large user bases and frequent campaigns can slow down core transactional databases and turn them into bottlenecks to business growth in e-commerce, financial services, online-to-offline (O2O) services, retail, and social networking. DDM provides linear horizontal scalability, expands processing capacity in real time, and supports highly concurrent transactions.

Financial services and mission-critical government and enterprise workloads

- Core transactions and accounting:

- GaussDB, in either a centralized or distributed deployment, is recommended for workloads that require financial-grade availability, strong consistency, and scalability.
 - For Oracle database migrations, use GaussDB with UGO for evaluation and schema conversion and DRS for online data migration.
- Peripheral systems and reporting:
For OLTP-oriented peripheral systems, consider RDS for PostgreSQL, RDS for MySQL, or a small-scale GaussDB deployment.

IoT, monitoring, and time-series platforms

- Monitoring and time-series metrics:
 - For InfluxDB-compatible workloads, use GeminiDB Influx API.
 - For large-scale time-series and spatiotemporal data, use GeminiDB Cassandra API.
- High performance and scalability: DDS supports high-QPS applications through a sharded architecture that enables rapid horizontal scaling as requirements change.

Gaming

DDS supports embedded documents, reducing the need for joins and simplifying development through flexible schemas that enable rapid iteration.

Choosing the Right Services for You

Start with the data model and workload requirements in the leftmost columns of the following table. Compare the characteristics across the remaining dimensions, and then use the final column to identify the recommended Huawei Cloud database service.

Data Model or Requirement	Core Workload Type	Performance and Latency	Scalability	Consistency	Compatibility and Ecosystem	Typical Use Case	Recommended Product
Relational	OLTP and hybrid transactional/analytical processing (HTAP)	Millisecond-level latency, extremely high TPS/QPS, and high concurrency	Online horizontal scaling with a distributed architecture	Strong ACID transactions	Standard SQL; partial Oracle syntax compatibility	Core financial transactions, mission-critical government and enterprise workloads, large ERP systems, and billing	GaussDB
Relational	OLTP	Millisecond-level latency and ultra-high throughput of millions of QPS	Decoupled storage and compute; storage autoscaling (up to 128 TB)	Strong ACID transactions	Fully compatible with MySQL	Highly concurrent internet applications, e-commerce, SaaS, and gaming	TaurusDB

Data Model or Requirement	Core Workload Type	Performance and Latency	Scalability	Consistency	Compatibility and Ecosystem	Typical Use Case	Recommended Product
Relational	OLTP	Millisecond-level latency and high reliability	Primarily vertical scaling, with horizontal read scaling through read replicas	Strong ACID transactions	Compatible with MySQL, PostgreSQL, and SQL Server	Small- and medium-sized enterprise websites, content management, mobile backends, development, and testing	RDS
Document	Content management, catalogs, and user profiles	Millisecond-level latency and high read/write throughput	Horizontal sharding	Configurable consistency	Fully compatible with MongoDB	Game equipment and player profiles, content management, mobile backends, and product catalogs	DDS

Data Model or Requirement	Core Workload Type	Performance and Latency	Scalability	Consistency	Compatibility and Ecosystem	Typical Use Case	Recommended Product
Key-value	In-memory caching	Millisecond-level latency	Vertical and horizontal scaling	Configurable consistency	Compatible with Redis	Application acceleration, session storage, leaderboards, counters, and distributed locks	DCS
	High-performance key-value caching and data structure storage	Ultra-low latency, with microsecond-level response times	Vertical and horizontal scaling	Configurable consistency	Compatible with Redis	Gaming, real-time advertising APIs, recommendation systems, flash sales, education, caching, and other highly concurrent, low-latency workloads	Gemini DB

Data Model or Requirement	Core Workload Type	Performance and Latency	Scalability	Consistency	Compatibility and Ecosystem	Typical Use Case	Recommended Product
	Key-value and document storage	Consistently low latency, with millisecond-level response times	Horizontal scaling	Configurable consistency	Compatible with DynamoDB	Highly concurrent web applications, IoT, e-commerce, retail, and elastic workloads	Gemini DB
Wide-column	Large-scale data ingestion and random reads	High throughput and millisecond-level latency	Horizontal scaling	Configurable consistency	Compatible with Cassandra	Large-scale data storage for industrial manufacturing, meteorology, and internet services	Gemini DB

Data Model or Requirement	Core Workload Type	Performance and Latency	Scalability	Consistency	Compatibility and Ecosystem	Typical Use Case	Recommended Product
Time-series	Time-series data storage	High-throughput writes and efficient aggregate queries	Horizontal scaling	Eventual consistency	Compatible with InfluxDB	Resource and business monitoring, IoT device monitoring, industrial monitoring, quality assessment, and fault tracing	Gemini DB

Data Model or Requirement	Core Workload Type	Performance and Latency	Scalability	Consistency	Compatibility and Ecosystem	Typical Use Case	Recommended Product
Database middle ware	Large-scale OLTP with large-scale data ingestion and random reads	Millisecond-level latency, high-throughput writes, and efficient aggregate queries	Vertical and horizontal scaling	Eventual consistency	Compatible with MySQL	Large-scale, highly concurrent workloads in e-commerce, financial services, O2O services, retail, social networking, IoT, government, large enterprises, and banking systems	DDM
Database tool	Online database migration and real-time synchronization	N/A	Vertical scaling	N/A	Transfers data across multiple source and destination database types	Database migration and synchronization during cloud adoption	DRS

Decision examples

- Example 1: building an e-commerce platform
 - Requirement analysis: Core transaction data is relational and requires strong ACID transactions. Traffic increases significantly during major promotional events, requiring rapid elasticity and extremely high throughput. The technology stack is based on MySQL.
 - Selection path in the table: Relational -> OLTP -> Ultra-high throughput -> Decoupled storage and compute -> Strong ACID transactions -> Compatible with MySQL. This path leads to **TaurusDB**. Additionally, frequently accessed product data and user sessions also require caching. Based on the "Key-value -> Millisecond-level latency" path, select **DCS** as the supporting cache service.
- Example 2: developing an IoT platform that collects sensor data from millions of devices
 - Requirement analysis: The data consists of timestamped time-series records. The write volume is extremely large. High-concurrency writes and horizontal scalability are required. The system must support aggregate analysis of the collected data.
 - Selection path in the table: Time-series -> Time-series data storage -> High-throughput writes -> Horizontal scaling. This path leads to **GeminiDB Influx API**.
- Example 3: building a new core transaction system for a bank
 - Requirement analysis: The data is strictly relational. The workload combines OLTP and HTAP. Financial-grade strong ACID transactions and high availability are required. The system must support online horizontal scaling to accommodate future business growth.
 - Selection path in the table: Relational -> OLTP and HTAP -> Extremely high TPS -> Online horizontal scaling -> Strong ACID transactions. This path leads to Huawei Cloud's flagship database product, **GaussDB**.
- Example 4: building a game equipment management system
 - Requirement analysis:
 - The system must store player equipment data, including dynamic fields such as equipment IDs, attributes, and levels.
 - It must support more than 100,000 read/write requests per second during peak equipment trading periods.
 - It must support projected annual data growth of 50 TB through elastic scaling.
 - The read/write latency must not exceed 10 ms to ensure a smooth gaming experience.
 - The system must support cross-region data synchronization for globally distributed game servers.
 - Selection path in the table: Document -> Millisecond-level latency and high read/write throughput -> Horizontal sharding -> Configurable consistency -> Fully compatible with MongoDB. This path leads to **DDS**.

Getting Started and Best Practices

This section provides resources to help you evaluate, configure, and start using the selected database services. Resources include instance selection guidance, quick-start tutorials, and best practices.

RDS

- RDS for MySQL
 - [Getting Started: Buying an RDS for MySQL Instance and Connecting to It Through DAS](#)
This tutorial shows you how to purchase an RDS for MySQL instance from the management console and connect to it through DAS.
 - [Getting Started: Buying an RDS for MySQL Instance and Connecting to It Using a MySQL Client](#)
This tutorial shows you how to purchase an RDS for MySQL instance from the management console and connect to it through a MySQL client.
 - [Getting Started: Buying an RDS for MySQL Instance and Connecting to It Using MySQL-Front](#)
This tutorial shows you how to purchase an RDS for MySQL instance from the management console and connect to it through MySQL-Front.
 - [RDS for MySQL Instance Selection Suggestions](#)
Before purchasing an RDS for MySQL instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares RDS for MySQL instance types, billing modes, and storage types to help you choose the right configuration.
 - [RDS for MySQL Best Practices](#)
This section describes best practices for working with RDS for MySQL and provides operational guidelines that you can follow when using this service.
- RDS for PostgreSQL
 - [Getting Started: Buying an RDS for PostgreSQL Instance and Connecting to It Through DAS](#)
This tutorial shows you how to purchase an RDS for PostgreSQL instance from the management console and connect to it through DAS.
 - [Getting Started: Buying an RDS for PostgreSQL Instance and Connecting to It Using a PostgreSQL Client](#)
This tutorial shows you how to purchase an RDS for PostgreSQL instance from the management console and connect to it through a PostgreSQL client.
 - [RDS for PostgreSQL Instance Selection Suggestions](#)
Before purchasing an RDS for PostgreSQL instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares RDS for PostgreSQL instance types, billing modes, and storage types to help you choose the right configuration.

- **RDS for PostgreSQL Best Practices**
This section describes best practices for working with RDS for PostgreSQL and provides operational guidelines that you can follow when using this service.
- RDS for SQL Server
 - **Getting Started: Connecting to an RDS for SQL Server Instance Through DAS**
This tutorial shows you how to connect to an RDS for SQL Server instance through DAS.
 - **Getting Started: Connecting to an RDS for SQL Server Instance Using the SSMS Client**
This tutorial shows you how to connect to an RDS for SQL Server instance using SQL Server Management Studio (SSMS).
 - **RDS for SQL Server Best Practices**
This section describes best practices for working with RDS for SQL Server and provides operational guidelines that you can follow when using this service.
- RDS for MariaDB
 - **Getting Started: Buying an RDS for MariaDB Instance and Connecting to It Through DAS**
This tutorial shows you how to purchase an RDS for MariaDB instance from the management console and connect to it through DAS.
 - **Getting Started: Buying an RDS for MariaDB Instance and Connecting to It Using a MariaDB Client**
This tutorial shows you how to purchase an RDS for MariaDB instance from the management console and connect to it through a MariaDB client.

TaurusDB

TaurusDB Enterprise Edition is Huawei's proprietary, enterprise-grade, cloud-native database for large-scale storage. It is fully compatible with MySQL. Built on DFV storage and a decoupled storage and compute architecture, it provides up to 128 TB of storage and zero data loss. It combines commercial-grade availability and performance with open-source cost efficiency.

- **Getting Started: Buying a DB Instance and Connecting to It Using the mysql Client**
This tutorial shows you how to purchase a TaurusDB instance from the management console and connect to it through a MySQL client.
- **Connecting to a DB Instance Through DAS**
This tutorial shows you how to purchase a TaurusDB instance from the management console and connect to it through DAS.
- **Suggestions on TaurusDB Instance Selection**
Before purchasing a TaurusDB instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares TaurusDB instance types, billing modes, and storage types to help you choose the right configuration.

- **TaurusDB Kernel Overview**
This tutorial introduces the architecture, key features, and capabilities of the proprietary TaurusDB kernel.
- **TaurusDB Best Practices**
This section describes best practices for working with TaurusDB and provides operational guidelines that you can follow when using this service.

GaussDB

- **Getting Started: Buying a GaussDB Instance and Connecting to It Through the DAS Console**
This tutorial shows you how to purchase a GaussDB instance from the management console and connect to it through DAS.
- **Getting Started: Buying a GaussDB Instance and Connecting to It Using gsql**
This tutorial shows you how to purchase a GaussDB instance from the management console and connect to it through a gsql client.
- **Suggestions on Instance Selection**
Before purchasing a GaussDB instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares GaussDB billing modes, edition types, instance types, deployment models, storage types, and instance specifications to help you choose the right configuration.
- **Best Practices**
This section describes best practices for working with GaussDB and provides operational guidelines that you can follow when using this service.

DCS

- **Getting Started with Redis and Reading and Writing Data Using Web CLI**
Using a single-node DCS Redis 5.0 instance as an example, this tutorial shows you how to create an instance on the DCS console, connect to it, and read and write data.
- **DCS Best Practices**
This section describes best practices for working with DCS and provides operational guidelines that you can follow when using this service.

GeminiDB

GeminiDB is a distributed, multi-model NoSQL database service built on a decoupled storage and compute architecture. It combines high performance, availability, reliability, security, and elastic scalability with one-click deployment, backup and restoration, monitoring, and alarm reporting.

- **GeminiDB Redis API**
 - **Getting Started**
This tutorial shows you how to purchase and connect to a GeminiDB Redis instance.

- **Suggestions on Instance Selection**

Before purchasing a GeminiDB Redis instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares product types, instance types, and billing modes of GeminiDB Redis API to help you choose the right configuration.
- **Best Practices**

This section describes best practices for working with GeminiDB Redis API and provides operational guidelines that you can follow when using this service.
- **GeminiDB Cassandra API**
 - **Getting Started**

This tutorial shows you how to purchase and connect to a GeminiDB Cassandra instance.
 - **Suggestions on Instance Selection**

Before purchasing a GeminiDB Cassandra instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares specifications, storage capacity, and billing modes of GeminiDB Cassandra API to help you choose the right configuration.
 - **Best Practices**

This section describes best practices for working with GeminiDB Cassandra API and provides operational guidelines that you can follow when using this service.
- **GeminiDB Influx API**
 - **Getting Started**

This tutorial shows you how to purchase and connect to a GeminiDB Influx instance.
 - **Suggestions on Instance Selection**

Before purchasing a GeminiDB Influx instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares instance types and billing modes of GeminiDB Influx API to help you choose the right configuration.
 - **Best Practices**

This section describes best practices for working with GeminiDB Influx API and provides operational guidelines that you can follow when using this service.
- **GeminiDB DynamoDB-Compatible API**
 - **Getting Started**

This tutorial shows you how to purchase and connect to a GeminiDB DynamoDB-Compatible instance.
 - **Suggestions on Instance Selection**

Before purchasing a GeminiDB DynamoDB-Compatible instance, consider factors such as the price, performance, workload capacity, and workload scenario to choose the right one for your needs. This section compares

specifications, storage capacity, and billing modes of GeminiDB DynamoDB-Compatible API to help you choose the right configuration.

- **Best Practices**

This section describes best practices for working with GeminiDB DynamoDB-Compatible API and provides operational guidelines that you can follow when using this service.

DDS

DDS is a secure, highly available, reliable, and elastically scalable database service that is fully compatible with the MongoDB protocol. It also provides one-click deployment, online scaling, disaster recovery, backup and restoration, monitoring, and alarm reporting. DDS integrates with third-party components that support the MongoDB protocol, including BI tools and Spark.

- **Getting Started: Buying and Connecting to a Cluster Instance**

This tutorial shows you how to purchase a DDS cluster instance from the management console and connect to it through DAS.

- **Getting Started: Buying and Connecting to a Replica Set Instance**

This tutorial shows you how to purchase a DDS replica set instance from the management console and connect to it through DAS.

- **Best Practices**

This section describes best practices for working with DDS and provides operational guidelines that you can follow when using this service.

DAS

DAS is a web-based service for connecting to and managing databases on Huawei Cloud.

- **Getting Started: Logging In to a DB Instance**

This tutorial shows you how to connect to a DB instance from the management console and introduces the supported database engine types.

- **Getting Started: Developing or Maintaining Databases**

This tutorial shows you how to perform routine database administration and O&M tasks.

- **Best Practices**

This section describes best practices for working with DAS and provides operational guidelines that you can follow when using this service.

DRS

DRS is a cloud service used for online database migration and real-time synchronization.

- **Getting Started: Creating a Real-Time Migration Task**

This tutorial covers the preparation and execution of a real-time migration task in DRS, including account and database preparation, network configuration, task creation, progress monitoring, and object and data comparison.

- **Getting Started: Creating a Real-Time Synchronization Task**
This tutorial covers the preparation and execution of a real-time synchronization task in DRS, including account and database preparation, network configuration, task creation, progress monitoring, and object and data comparison.
- **Best Practices**
This section describes best practices for working with DRS and provides operational guidelines that you can follow when using this service.

UGO

UGO is a managed service for heterogeneous database schema migration.

- **Getting Started: Database Evaluation**
This tutorial shows you how to prepare for database evaluation, create an evaluation project, and view the evaluation results.
- **Getting Started: Database Schema Migration**
This tutorial shows you how to create a database migration project and complete database migration and verification.
- **Best Practices**
This section covers the complete process for migrating Oracle schemas to GaussDB, including solution design, resource planning, implementation, and each stage of object migration.

DDM

DDM is a MySQL-compatible, distributed middleware service designed for relational databases. It addresses distributed scaling challenges, overcomes database capacity and performance bottlenecks, and supports highly concurrent access to massive data volumes.

- **Getting Started: Buying a DDM Instance and Connecting to a Schema Using DAS**
This tutorial shows you how to purchase a DDM instance from the management console and connect to it through DAS.
- **Best Practices: Formulating Sharding Rules**
This tutorial shows you how to select an appropriate sharding key and sharding algorithm when creating a sharded table.
- **Best Practices: Determining the Number of Shards in a Schema**
This tutorial shows you how to select an appropriate number of shards in a schema when creating a sharded schema.