Document Database Service

Service Overview

Issue	01
Date	2025-01-17

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.

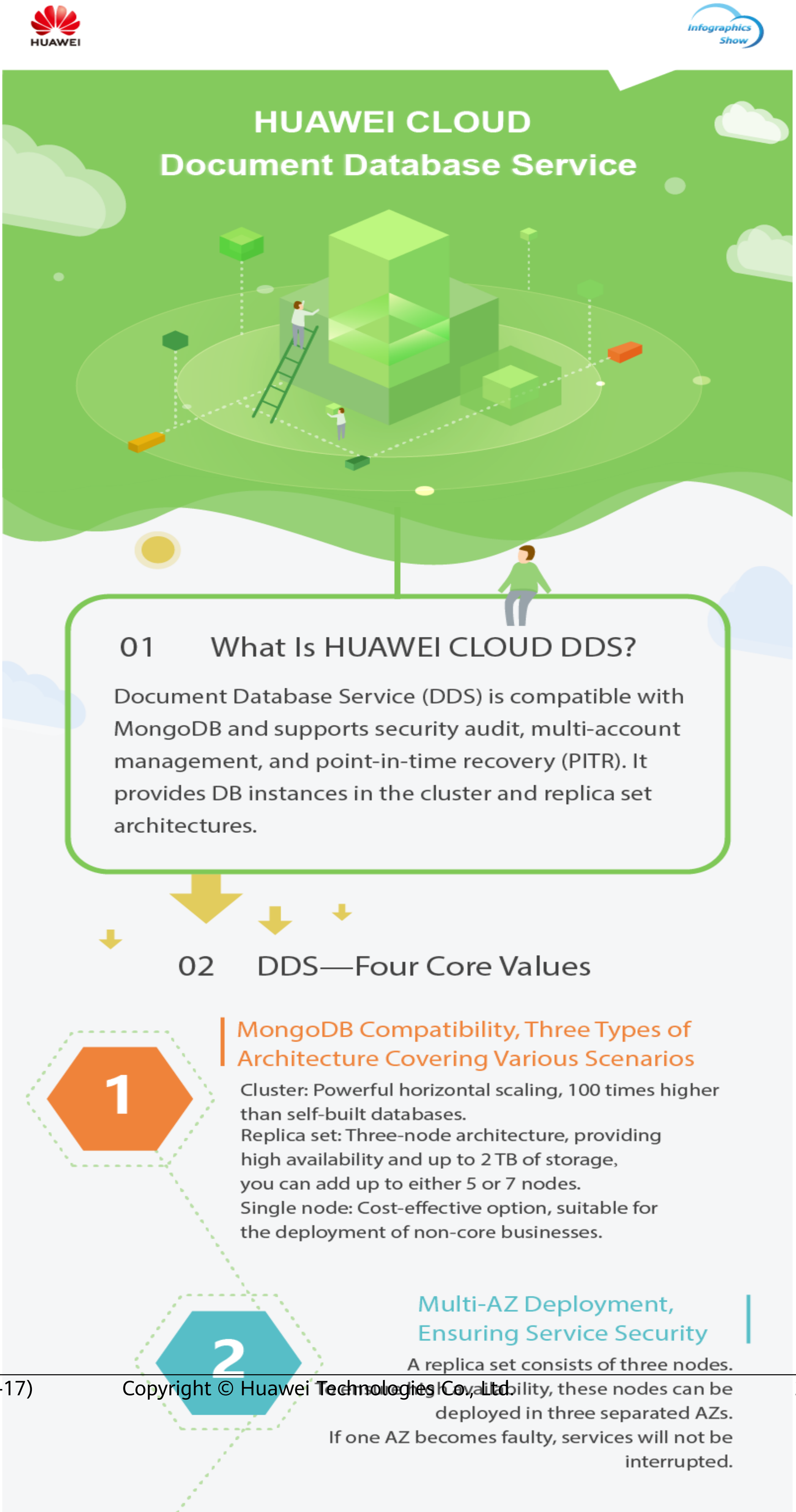
Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Infographics



HUAWEI CLOUD
Document Database Service

01 What Is HUAWEI CLOUD DDS?

Document Database Service (DDS) is compatible with MongoDB and supports security audit, multi-account management, and point-in-time recovery (PITR). It provides DB instances in the cluster and replica set architectures.

02 DDS—Four Core Values

MongoDB Compatibility, Three Types of Architecture Covering Various Scenarios

Cluster: Powerful horizontal scaling, 100 times higher than self-built databases.
Replica set: Three-node architecture, providing high availability and up to 2 TB of storage, you can add up to either 5 or 7 nodes.
Single node: Cost-effective option, suitable for the deployment of non-core businesses.

Multi-AZ Deployment, Ensuring Service Security

A replica set consists of three nodes. To ensure high availability, these nodes can be deployed in three separated AZs. If one AZ becomes faulty, services will not be interrupted.

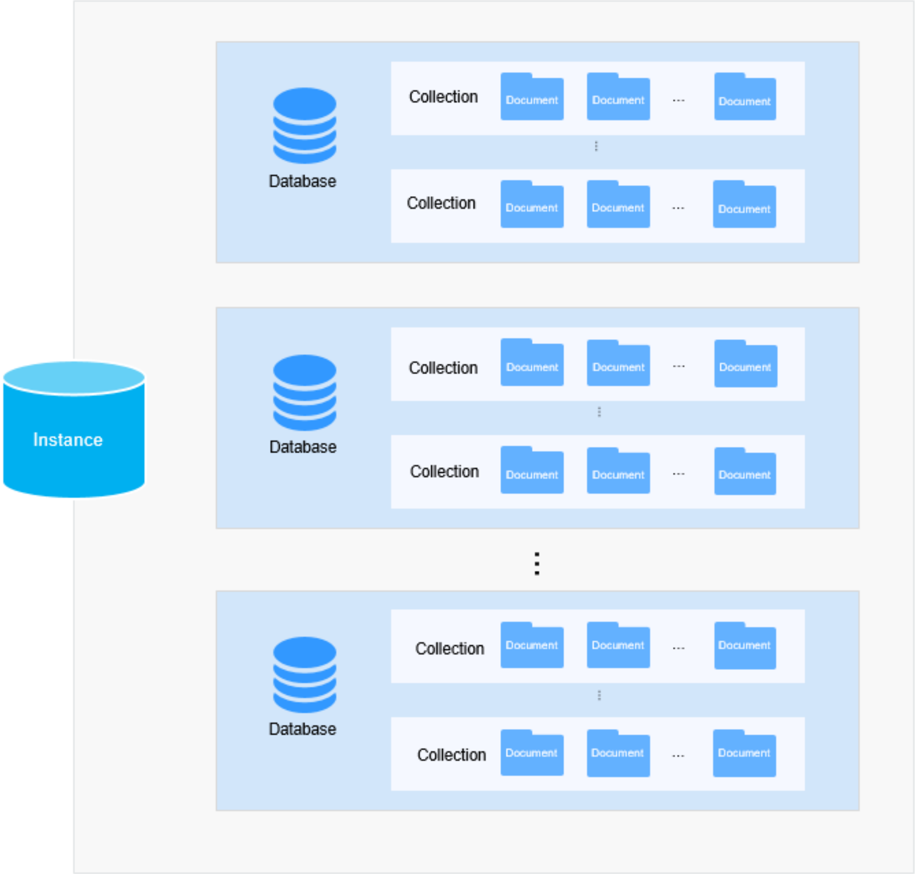
2 What Is DDS?

Document Database Service (DDS), compatible with MongoDB, is a secure, high availability (HA) database service that is reliable, scalable, and easy to use. It provides functions such as one-click deployment, elastic capacity expansion, disaster recovery, backup, restoration, monitoring, and alarm reporting.

Before using DDS, you need to be familiar with MongoDB and the MongoDB protocol. For details, see **official documents**.

Storage Structure

Figure 2-1 Storage structure



The basic management unit of DDS is an instance. A DDS instance consists of database, collection, and document.

DDS is fully compatible with the MongoDB protocol, so some terms, data structures, and basic syntax of DDS are the same as those of MongoDB. The following table lists the difference between common terms in MongoDB and relational databases to help you better understand and use DDS.

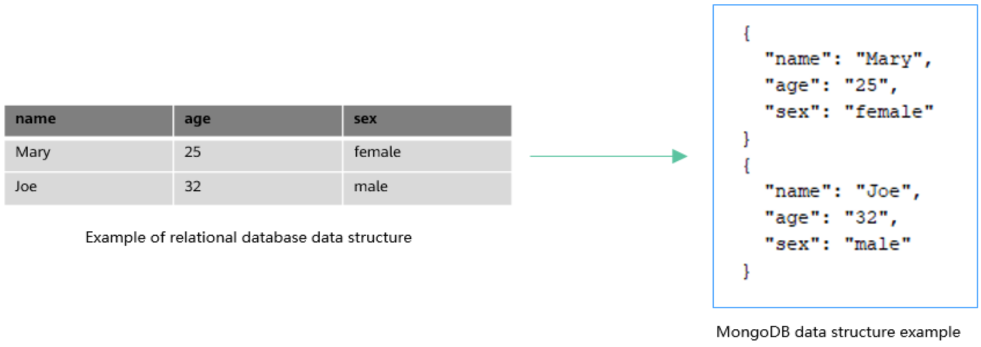
Table 2-1 Terms

MongoDB	Description	Relational Database
Database	A DDS instance contains multiple databases, and a database contains multiple collections.	Database
Collection	A collection is a group of MongoDB documents.	Table
Document	A document is a group of key-value pairs (BSONs) and is the most basic unit in MongoDB.	Row

Data Structure

MongoDB uses JSON-like documents when storing documents in collections. The following figure compares data structures between relational databases and MongoDB, helping you have a better understanding of MongoDB concepts.

Figure 2-2 Data structures



name	age	sex
Mary	25	female
Joe	32	male

Example of relational database data structure

```
{
  "name": "Mary",
  "age": "25",
  "sex": "female"
}
{
  "name": "Joe",
  "age": "32",
  "sex": "male"
}
```

MongoDB data structure example

Why DDS?

For details, see **Advantages** and **Typical Application Scenarios**.

Product Pricing

For details, see **Billing**.

Compatible DB Engines and Versions

For details about the compatible DB engines and versions, see **DB Engines and Versions**.

Deployment Suggestions

Before deploying DDS, consider the following factors:

- Region and Availability Zone: You should select a region and AZs based on the geographical locations of your users, product prices, DR capabilities, and network latency. Once DB instances are successfully created in specific AZs in a region, the region cannot be changed. For more information, see **Region and AZ**.
- Network planning: When creating a DDS DB instance, you are advised to use the Virtual Private Network (VPC) and subnet where the Elastic Cloud Server (ECS) is deployed.
- Data security: DDS provides comprehensive assurance for data security. You can deploy DB instances in multiple AZs and use audit logs, isolated networks, security groups, and data encryption to ensure data security.

3 Advantages

DDS provides several types of DB instances with high reliability and scalability. You can manage them using various O&M tools in a visual way.

Diverse Instance Types

There are many instance types to meet your requirements in different scenarios.

Table 3-1 Architecture

Instance Architecture	Description	Application Scenario
Cluster	A DDS cluster consists of three types of nodes: dds mongos, shard, and config. You can select the number of dds mongos and shard nodes and the cluster configuration you need based on your performance requirements.	Cluster instances are recommended for service systems that require both high availability and scalability.
Replica set	A DDS replica set consists of three nodes: primary, secondary, and hidden. Data is automatically synchronized between nodes to ensure high data reliability. If a primary node goes down or becomes faulty, the replica set elects a secondary node as a new primary node and continues normal operations.	Replica set instance suits small- and medium-sized service systems that require high availability.

Instance Architecture	Description	Application Scenario
Single node **NOTE** Huawei Cloud has discontinued the sale of DDS single node instances since July 15, 2023.	While not as robust as replica sets, single node architecture can be a less expensive way to visualize O&M and provide elastic scaling.	Single node instances are useful for R&D, testing, and non-core data storage of enterprises.

Elastic Scaling

DDS allows you to change the storage space, node quantity, CPU, and memory of an instance.

Friendly UI

A web-based console provides comprehensive monitoring information, making your operations easy and visual.

Secure

- VPCs, subnets, security groups, storage encryption, SSL, and DDoS protection are provided to defend against various malicious attacks and ensure data security.
- DDS supports audit logs that record your operations on databases or collections, helping you audit database security and analyze root cause of faults.
- Fine-grained permission management is supported.

Abundant O&M Tools

- Backup and restoration

  DDS supports full backup, incremental backup, and point-in-time backup and restoration of replica sets. Backup data can be stored for up to 732 days.
- Monitoring and alarm reporting

DDS monitors instance metrics. You can customize the objects you want to monitor and configure notification policies to learn about the database service status in a timely manner.

Other advantages

DDS also boasts outstanding service availability, data durability, system security, and maintenance cost. For details, see **Comparison Between DDS and On-Premises Databases**.

4 Comparison Between DDS and On-Premises Databases

DDS provides high availability. It is more reliable, secure, and cost-effective than self-built databases.

Table 4-1 Comparison between DDS and on-premises databases

Item	DDS	On-Premises Database
Service availability	99.95%	Requires setting up the primary/standby relationship and HA environment for ensuring the availability.
Data durability	99.99999999%	Requires self-guarantee, primary/standby relationship setup, and RAID setup.
System security	● Anti-Distributed denial of service (DDoS) protection is provided, and system vulnerabilities are automatically addressed in a timely manner. ● Supports log auditing.	● Requires procurement of expensive devices and software, as well as manual detection and fix of security vulnerabilities. ● Requires procurement of an audit system.

Item	DDS	On-Premises Database
Database backup	● Automated backup is supported. You can configure backup policies based on service requirements. ● You can create manual or physical backups at any time. DDS helps increase backup efficiency by 3 times. ● Backup files are automatically uploaded to Object Storage Service (OBS) for storage.	● Requires self-setup and maintenance. ● The open-source version supports only logical backup, resulting in low backup efficiency.
Monitoring and alarm reporting	DDS interconnects with Cloud Eye, which provides a visualized platform for you to view monitoring metrics and set threshold alarms.	You have to purchase three servers and hosting fees can be expensive.
Hosting	There are no hosting fees.	Requires purchase of three servers and payment of high hosting fees.
Maintenance cost	There are no additional O&M costs, and second-by-second performance monitoring, threshold alarms, and event alarms can all be configured.	Requires a lot of workforce investment and professional database administrator (DBA) for maintenance.
Deployment and scaling	Supports quick deployment, flexible application, elastic scaling, and one-click specification change.	Requires procurement, deployment, and coordination of hardware that matches original devices.
Log transfer	Stores slow query logs and error logs for one month.	Requires you to transfer, export, and query log information on your own.
High availability	Provides high availability capabilities, supporting second-level switchover and failover.	Requires self-setup of HA monitoring. Data may be lost after a manual primary/ secondary switchover is performed using commands.

5 Typical Application Scenarios

Games

Player information generated, like their equipment and bonus points, are stored in DDS databases. During peak hours, DDS cluster instances can handle large amounts of concurrent requests. DDS cluster and replica set provide high availability to ensure the games are stable in high-concurrency scenarios.

In addition, DDS is compatible with MongoDB and provides a non-schema mode, which frees you from having to change table structures when the play modes change. DDS can meet the flexible gaming requirements. You can store structured data with fixed schemas in Relational Database Service (RDS), store services with flexible schemas in DDS, and store hot data in GeminiDB Redis, improving data efficiency and reducing data storage costs.

Advantages:

- **Supports Embedded Documents**: Embedded documents eliminate the need for JOIN statements, which simplifies application development. Flexible schemas also facilitate rapid development and iteration.
- **Easy to Cope with Peak Pressure**: Sharded clusters provide enough capacity to store data into the TB range.

IoT

DDS is compatible with MongoDB and provides high-performance and asynchronous data writes. In certain scenarios, DDS can deliver performance comparable to an in-memory database. In addition, cluster instances can dynamically add dds mongos and shard nodes or upgrade specifications. The performance and storage space can be quickly expanded, making cluster instances suitable for IoT scenarios with high concurrent writes.

Intelligent IoT terminals need to collect various types of data, store device logs, and analyze various types of information. In recent years, IoT services have grown rapidly, generating huge volumes of data and increasing access traffic. IoT has created demand for horizontal storage scaling.

DDS provides a secondary index to meet dynamic query requirements and uses the MapReduce aggregation framework, which is compatible with MongoDB, to analyze data from multiple dimensions.

Advantages:

- **High Write Performance**: DDS sharded clusters provide the robust write performance needed to handle terabyte-scale databases.
- **High Performance and Scalability**: DDS supports applications with high QPS rates, and its sharding architecture can be scaled in or out to flexibly cope with application changes.

Internet

DDS replica sets use the three-node HA architecture. Three data nodes form an anti-affinity group and are deployed on different physical servers to automatically synchronize data. The primary and secondary nodes provide services. Each node has a private IP address and works with Driver to allocate read workloads.

Many organizations need to process and store data into the TB range, requiring data to be written to databases in real time and dynamic analysis capabilities in big data computing.

Advantages:

- **MapReduce:** With a complete data analysis utility, you can query statements or scripts, and distribute requests to DDS.
- **Excellent Scalability**: DDS DB instances can be scaled up to support growing services and data volumes in content management systems.

Others

- Social: DDS allows you to easily discover people or places nearby through geographical indexing. DDS provides a wide range of queries. It is suitable for storing chat content, and data can be read and written quickly.
- Big data: DDS can used as the cloud storage system of big data. Its flexible aggregation facilitates data extraction and analysis.
- Logistics: Order information is stored in arrays embedded in a DDS DB instance. Even if the order status is continuously updated during delivery, all order changes can be read through one query.

6 Functions and Features

This section describes the main functions and features of DDS.

Migrating Data

With **Data Replication Service (DRS)**, you can migrate self-built MongoDB databases or other cloud MongoDB databases to DDS in real time. DRS simplifies data transmission between databases and reduces data transmission costs.

Scaling up Storage

If the storage you purchased cannot meet your requirements, you can **scale up the instance**.

Changing Instance Class

If the CPU or memory of an instance cannot meet your service requirements, you can **change the instance class** on the console.

Backup and Restoration

- You can **configure the automated backup policy** and **create backups**. The maximum retention period for an automated backup is 732 days. A manual backup can be retained until you delete it.
- Data can be **restored using backup files**. Replica set instances support point-in-time recovery at the instance level, database level and table level.

Monitoring Metrics

DDS **monitors key performance metrics** of instances and DB engines in real time, including the CPU usage, memory usage, storage space usage, command execution frequency, delete statement execution frequency, insert statement execution frequency, and number of active connections.

Managing Logs

DDS allows you to query and download database **error logs** and **slow query logs**.

Configuring Parameters

DDS allows you to manage parameter templates and **modify parameter template settings** on the console.

Cross-AZ Disaster Recovery

If the region where your instance is located supports three or more AZs, you can deploy the instance across three AZs to achieve **cross-AZ disaster recovery**.

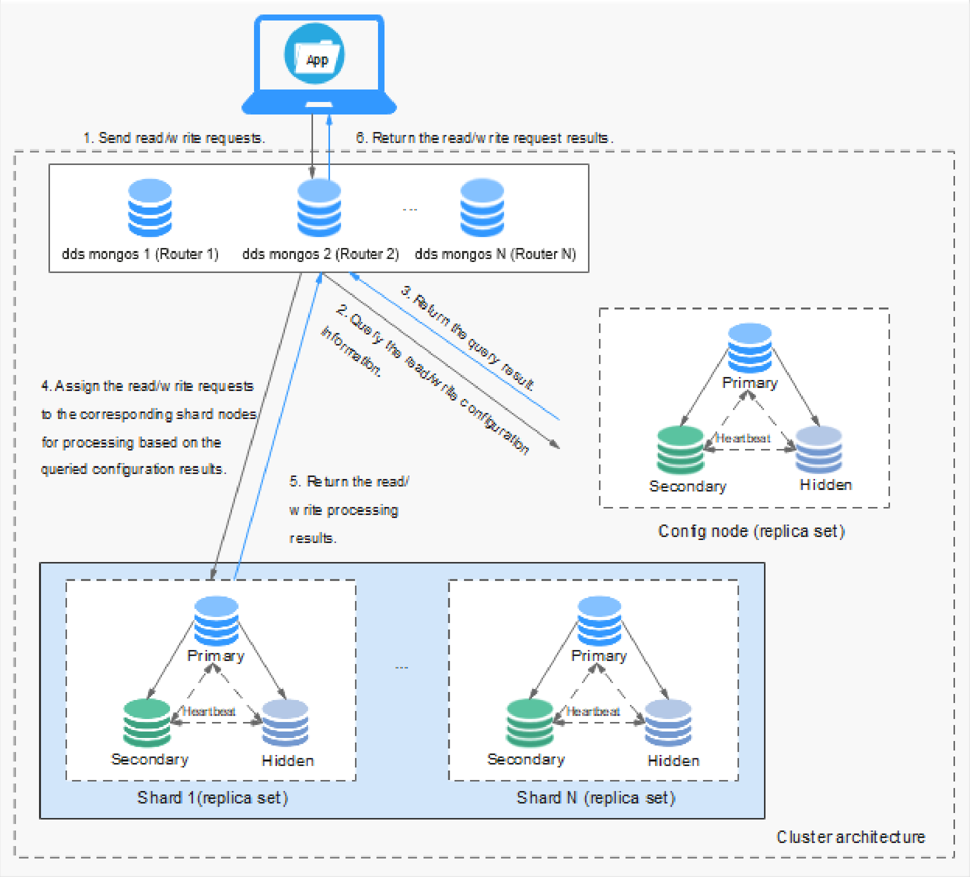
7 System Architecture

7.1 Cluster

A cluster consists of a config node, and multiple dds mongos and shard nodes.

Data read and write requests are forwarded by the dds mongos nodes, which read configuration settings from config, and then allocate the read and write requests to the shards, making it easy to cope with high concurrency scenarios. In addition, each config node, along with the shards in its cluster, is replicated in triplicate to ensure high availability.

Figure 7-1 Cluster architecture


App
1. Send read/write requests.
6. Return the read/write request results.
dds mongos 1 (Router 1)
dds mongos 2 (Router 2)
...
dds mongos N (Router N)
2. Query the read/write configuration information.
3. Return the query result.
4. Assign the read/write requests to the corresponding shard nodes for processing based on the queried configuration results.
5. Return the read/write processing results.
Primary
Heartbeat
Secondary
Hidden
Config node (replica set)
Primary
Heartbeat
Secondary
Hidden
Shard 1 (replica set)
...
Primary
Heartbeat
Secondary
Hidden
Shard N (replica set)
Cluster architecture

- Each dds mongos is a single node, but you can provision multiple dds mongos nodes for load balancing and failovers. A single cluster can contain 2 to 32 dds mongos nodes.
- Each shard is a three-node replica set. A single cluster supports 2 to 32 shards.
- A config node is a necessary part of a cluster instance, and is also deployed as a three-node replica set. The config node stores instance configuration data.
- Cluster is suitable for handling large amounts of data and can meet the requirements for high availability and scalability.

7.2 Replica Set

A replica set consists of a set of mongod processes. It is a collection of nodes that help ensure data redundancy and reliability.

NOTE

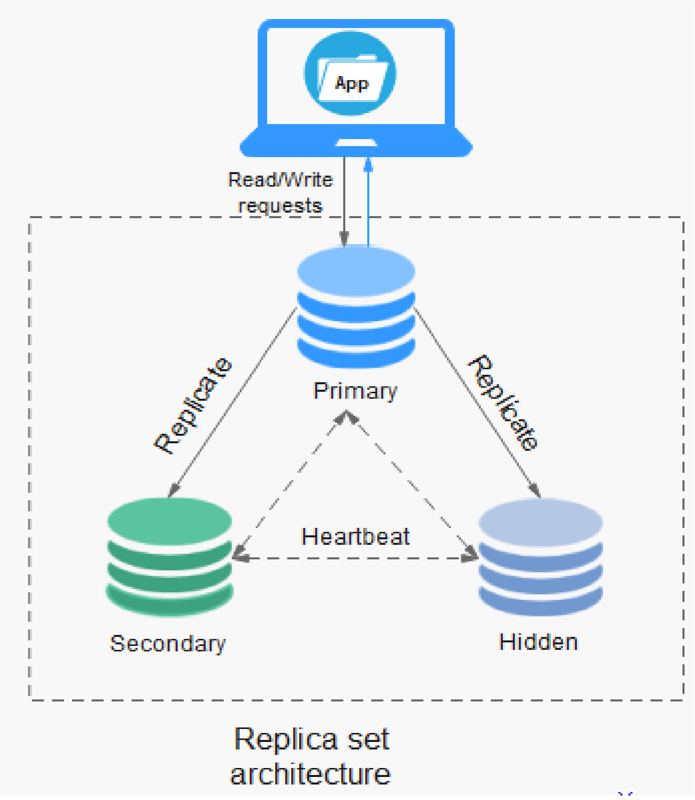
For details about the mongod process, see the MongoDB **official documentation**.

A replica set consists of three nodes: primary, secondary, and hidden. The three-node architecture is set up automatically, and the three nodes automatically synchronize data with each other to ensure data reliability. Replica sets are recommended for small- and medium-sized service systems that require high availability.

- Primary node: Primary nodes are used to process both read and write requests.
- Secondary node: Secondary nodes are used to process read requests only.
- Hidden node: Hidden nodes are used to back up service data.

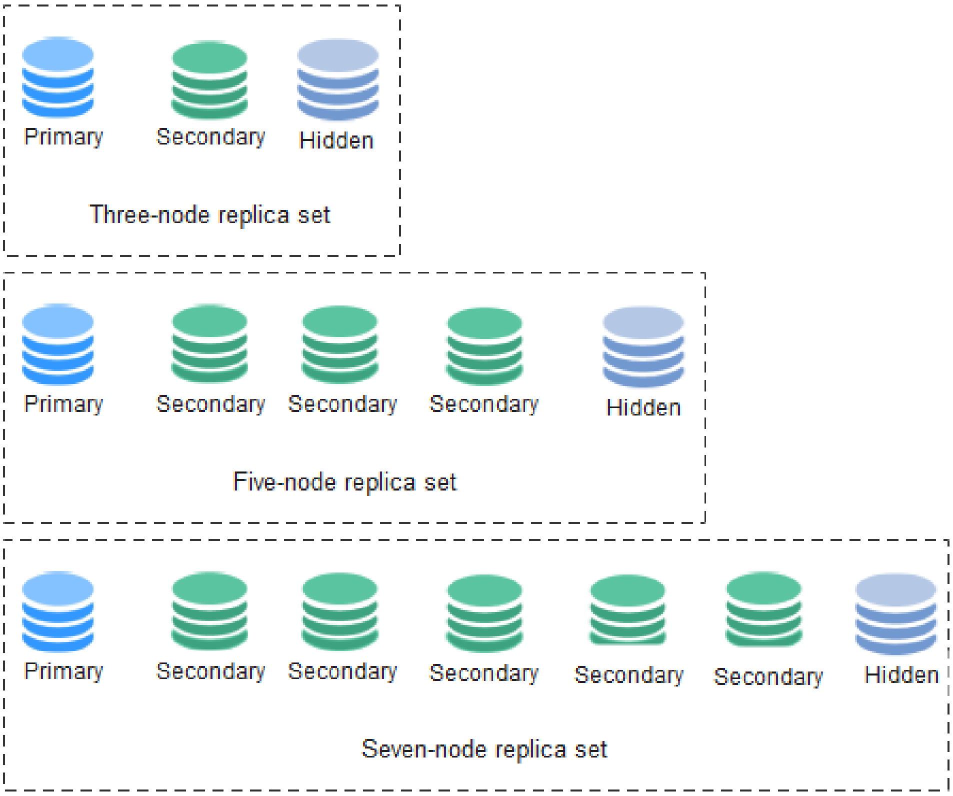
You can perform operations on the primary and secondary nodes. If the primary node is faulty, the system automatically selects a new primary node. The following figure shows the replica set architecture.

Figure 7-2 Three-node replica set architecture



After a replica set is created, you can add up to either 5 or 7 nodes. For details about how to add a replica set node, see **Adding Replica Set Instance Nodes**.

Figure 7-3 Replica set instance nodes



7.3 Single Node

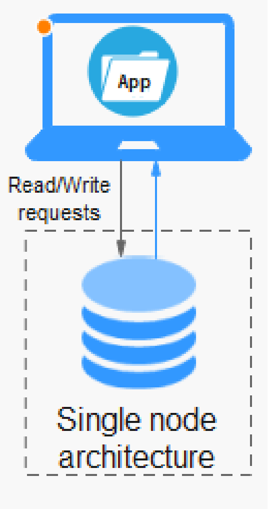
NOTE

Huawei Cloud has discontinued the sale of DDS single node instances since July 15, 2023.

A single node instance, as its name suggests, contains only one node and that node can be accessed directly.

As a supplement to the cluster and replica set architectures, the single-node architecture is useful for R&D, SQL commissioning, and lab project testing. While not as robust as replica sets, single node architecture can be a less expensive way to visualize O&M and provide elastic scaling.

Figure 7-4 Single node architecture



8 Instances

8.1 DB Engines and Versions

DDS supports versions 3.4, 4.0, 4.2, 4.4 and 5.0, and you need to use a driver compatible with MongoDB 3.0 or later to access DDS. You can select the DB engine and version you need based on your service requirements.

Table 8-1 Supported DB engines and versions

Compatibility	CPU Type	DB Instance Type	Storage Engine
5.0	● x86 ● Kunpeng	● Cluster ● Replica set	RocksDB
4.4	● x86 ● Kunpeng	● Cluster ● Replica set	RocksDB
4.2	● x86 ● Kunpeng	● Cluster ● Replica set	RocksDB
4.0	● x86 ● Kunpeng	● Cluster ● Replica set	WiredTiger
3.4	● x86 ● Kunpeng	● Cluster ● Replica set	WiredTiger

NOTE

In versions 4.2, 4.4, and 5.0, the Kunpeng type is available only to whitelisted users. To use this function, you need to submit a service ticket. In the upper right corner of the management console, choose **Service Tickets > Create Service Ticket** to submit a service ticket.

For details about the compatibility between DDS and MongoDB Community Editions, see **Versions**.

8.2 Instance Specifications

8.2.1 Cluster

Instances of the same type have different memory specifications. You can select instances of different specifications based on application scenarios.

This section describes cluster instance specifications. The instance specifications depend on the selected CPU model.

NOTE

The default maximum number of connections has been adjusted. Query the number of connections based on the instance creation time. For details about the number of connections for instances created before July 2021, see **Historical Node Specifications**.

Node Specifications

DDS cluster instances have two specification types: general-purpose and enhanced II. For details, see **Table 8-2**.

Table 8-2 Specification type

Specification Type	Description	Applicable Scenario
General-purpose	CPU resources are shared with other general-purpose DB instances on the same physical machine. CPU usage is maximized through resource overcommitment. This instance class is a cost-effective option and suitable for scenarios where performance stability is not critical.	Suitable for scenarios that have high requirements on cost-effectiveness.
Enhanced II	With a leading network acceleration engine and Data Plane Development Kit (DPDK) fast packet processing mechanism, this instance class provides higher network performance and computing power.	Suitable for websites and web applications that require high database computing and network performance.

Table 8-3 config specifications

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
x86	Enhanced II	2	4	dds.mongodb.c6.large.2.config	2000	200-2000
		2	16	dds.mongodb.c6.large.8.config	2000	200-2000
		4	32	dds.mongodb.c6.xlarge.8.config	4000	200-4000
		8	64	dds.mongodb.c6.2xlarge.8.config	16000	200-16000
Kunpeng	-	2	4	dds.mongodb.large.arm2.config	2000	200-2000
		4	8	dds.mongodb.xlarge.arm2.config	4000	200-4000
		8	16	dds.mongodb.2xlarge.arm2.config	16000	200-16000

Table 8-4 dds mongos and shard specifications

CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
x86	General-purpose	dds mongos	1	4	dds.mongodb.s6.medium.4.mongos	1000	200-1000
			2	4	dds.mongodb.s6.large.2.mongos	2000	200-2000
			2	8	dds.mongodb.s6.large.4.mongos	2000	200-2000
			4	8	dds.mongodb.s6.xlarge.2.mongos	4000	200-4000

CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
			4	16	dds.mongodb.s6.xlarge.4.mongos	4000	200-4000
			8	16	dds.mongodb.s6.2xlarge.2.mongos	16000	200-16000
			8	32	dds.mongodb.s6.2xlarge.4.mongos	16000	200-16000
		shard	1	4	dds.mongodb.s6.medium.4.shard	1000	200-1000
			2	4	dds.mongodb.s6.large.2.shard	2000	200-2000
			2	8	dds.mongodb.s6.large.4.shard	2000	200-2000
			4	8	dds.mongodb.s6.xlarge.2.shard	4000	200-4000
			4	16	dds.mongodb.s6.xlarge.4.shard	4000	200-4000
			8	16	dds.mongodb.s6.2xlarge.2.shard	16000	200-16000
			8	32	dds.mongodb.s6.2xlarge.4.shard	16000	200-16000
	Enhanced II	dds mongos	2	8	dds.mongodb.c6.large.4.mongos	2000	200-2000
			2	16	dds.mongodb.c6.large.8.mongos	2000	200-2000
			4	16	dds.mongodb.c6.xlarge.4.mongos	4000	200-4000
			4	32	dds.mongodb.c6.xlarge.8.mongos	4000	200-4000
			8	32	dds.mongodb.c6.2xlarge.4.mongos	16000	200-16000
			8	64	dds.mongodb.c6.2xlarge.8.mongos	16000	200-16000

CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
			16	64	dds.mongodb.c6.4xlarge.4.mongos	16000	200-16000
			16	128	dds.mongodb.c6.4xlarge.8.mongos	16000	200-16000
			32	128	dds.mongodb.c6.8xlarge.4.mongos	16000	200-16000
			32	256	dds.mongodb.c6.8xlarge.8.mongos	16000	200-16000
			64	256	dds.mongodb.c6.16xlarge.4.mongos	16000	200-16000
		shard	2	8	dds.mongodb.c6.large.4.shard	2000	200-2000
			2	16	dds.mongodb.c6.large.8.shard	2000	200-2000
			4	16	dds.mongodb.c6.xlarge.4.shard	4000	200-4000
			4	32	dds.mongodb.c6.xlarge.8.shard	4000	200-4000
			8	32	dds.mongodb.c6.2xlarge.4.shard	16000	200-16000
			8	64	dds.mongodb.c6.2xlarge.8.shard	16000	200-16000
			16	64	dds.mongodb.c6.4xlarge.4.shard	16000	200-16000
			16	128	dds.mongodb.c6.4xlarge.8.shard	16000	200-16000
			32	128	dds.mongodb.c6.8xlarge.4.shard	16000	200-16000
			32	256	dds.mongodb.c6.8xlarge.8.shard	16000	200-16000

CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
			64	256	dds.mongodb.c6.16xlarge.4.shard	16000	200-16000
Kunpeng	-	dds mongos	2	4	dds.mongodb.large.arm2.mongos	2000	200-2000
	-		2	8	dds.mongodb.large.arm4.mongos	2000	200-2000
	-		4	8	dds.mongodb.xlarge.arm2.mongos	4000	200-4000
	-		4	16	dds.mongodb.xlarge.arm4.mongos	4000	200-4000
	-		8	16	dds.mongodb.2xlarge.arm2.mongos	16000	200-16000
	-		8	32	dds.mongodb.2xlarge.arm4.mongos	16000	200-16000
	-		16	32	dds.mongodb.4xlarge.arm2.mongos	16000	200-16000
	-		16	64	dds.mongodb.4xlarge.arm4.mongos	16000	200-16000
	-	shard	2	4	dds.mongodb.large.arm2.shard	2000	200-2000
	-		2	8	dds.mongodb.large.arm4.shard	2000	200-2000
	-		4	8	dds.mongodb.xlarge.arm2.shard	4000	200-4000
	-		4	16	dds.mongodb.xlarge.arm4.shard	4000	200-4000
	-		8	16	dds.mongodb.2xlarge.arm2.shard	16000	200-16000

CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
	-		8	32	dds.mongodb.2xlarge.arm4.shard	16000	200-16000
	-		16	32	dds.mongodb.4xlarge.arm2.shard	16000	200-16000
	-		16	64	dds.mongodb.4xlarge.arm4.shard	16000	200-16000

Historical Node Specifications

For details about the range of connections of instances created before July 2021, see the following table.

Table 8-5 config specifications

Instance Type	CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
Cluster	x86	General-purpose	2	4	dds.mongodb.s6.large.2.config	400	200-2000
		Enhanced	2	4	dds.mongodb.c3.large.2.config		
		Enhanced II	2	4	dds.mongodb.c6.large.2.config		
	Kunpeng	-	2	4	dds.mongodb.large.arm2.config		

Table 8-6 dds mongos and shard specifications

Instance Type	CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
Cluster	x86	General-purpose	dds mongos	1	4	dds.mongodb.s6.medium.4.mongos	400	200-1000
				2	4	dds.mongodb.s6.large.2.mongos	400	200-2000
				2	8	dds.mongodb.s6.large.4.mongos	400	200-2000
				4	8	dds.mongodb.s6.xlarge.2.mongos	1000	200-4000
				4	16	dds.mongodb.s6.xlarge.4.mongos	1000	200-4000
				8	16	dds.mongodb.s6.2xlarge.2.mongos	4000	200-16000
				8	32	dds.mongodb.s6.2xlarge.4.mongos	4000	200-16000
			shard	1	4	dds.mongodb.s6.medium.4.shard	400	200-1000
				2	4	dds.mongodb.s6.large.2.shard	400	200-2000
				2	8	dds.mongodb.s6.large.4.shard	400	200-2000
				4	8	dds.mongodb.s6.xlarge.2.shard	1000	200-4000

Instance Type	CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
				4	16	dds.mongodb.s6.xlarge.4.shard	1000	200-4000
				8	16	dds.mongodb.s6.2xlarge.2.shard	4000	200-16000
				8	32	dds.mongodb.s6.2xlarge.4.shard	4000	200-16000
		Enhanced	dds mongos	2	8	dds.mongodb.c3.large.4.mongos	400	200-2000
				4	16	dds.mongodb.c3.xlarge.4.mongos	1000	200-4000
				8	32	dds.mongodb.c3.2xlarge.4.mongos	4000	200-16000
				16	64	dds.mongodb.c3.4xlarge.4.mongos	8000	200-16000
				32	128	dds.mongodb.c3.8xlarge.4.mongos	8000	200-16000
				60	256	dds.mongodb.c3.15xlarge.4.mongos	8000	200-16000
			shard	2	8	dds.mongodb.c3.large.4.shard	400	200-2000
				4	16	dds.mongodb.c3.xlarge.4.shard	1000	200-4000

Instance Type	CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
				8	32	dds.mongodb.c3.2xlarge.4.shard	4000	200-16000
				16	64	dds.mongodb.c3.4xlarge.4.shard	8000	200-16000
				32	128	dds.mongodb.c3.8xlarge.4.shard	8000	200-16000
				60	256	dds.mongodb.c3.15xlarge.4.shard	8000	200-16000
		Enhanced II	dds mongos	2	8	dds.mongodb.c6.large.4.mongos	400	200-2000
				4	16	dds.mongodb.c6.xlarge.4.mongos	1000	200-4000
				8	32	dds.mongodb.c6.2xlarge.4.mongos	4000	200-16000
				16	64	dds.mongodb.c6.4xlarge.4.mongos	8000	200-16000
				32	128	dds.mongodb.c6.8xlarge.4.mongos	8000	200-16000
				64	256	dds.mongodb.c6.16xlarge.4.mongos	8000	200-16000
			shard	2	8	dds.mongodb.c6.large.4.shard	400	200-2000

Instance Type	CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
				4	16	dds.mongodb.c6.xlarge.4.shard	1000	200-4000
				8	32	dds.mongodb.c6.2xlarge.4.shard	4000	200-16000
				16	64	dds.mongodb.c6.4xlarge.4.shard	8000	200-16000
				32	128	dds.mongodb.c6.8xlarge.4.shard	8000	200-16000
				64	256	dds.mongodb.c6.16xlarge.4.shard	8000	200-16000
	Kunpeng	-	dds mongos	2	4	dds.mongodb.large.arm2.mongos	400	200-2000
		-		2	8	dds.mongodb.large.arm4.mongos	400	200-2000
		-		4	8	dds.mongodb.xlarge.arm2.mongos	1000	200-4000
		-		4	16	dds.mongodb.xlarge.arm4.mongos	1000	200-4000
		-		8	16	dds.mongodb.2xlarge.arm2.mongos	4000	200-16000
		-		8	32	dds.mongodb.2xlarge.arm4.mongos	4000	200-16000

Instance Type	CPU Type	Specifications	Component	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
		-		16	32	dds.mongodb.4xlarge.arm2.mongos	8000	200-16000
		-		16	64	dds.mongodb.4xlarge.arm4.mongos	8000	200-16000
		-	shard	2	4	dds.mongodb.large.arm2.shard	400	200-2000
		-		2	8	dds.mongodb.large.arm4.shard	400	200-2000
		-		4	8	dds.mongodb.xlarge.arm2.shard	1000	200-4000
		-		4	16	dds.mongodb.xlarge.arm4.shard	1000	200-4000
		-		8	16	dds.mongodb.2xlarge.arm2.shard	4000	200-16000
		-		8	32	dds.mongodb.2xlarge.arm4.shard	4000	200-16000
		-		16	32	dds.mongodb.4xlarge.arm2.shard	8000	200-16000
		-		16	64	dds.mongodb.4xlarge.arm4.shard	8000	200-16000

Helpful Links

1. How do I change the maximum number of connections of a cluster instance?

You can change the maximum number of connections to a DB instance by modifying the **net.maxIncomingConnections** parameter. For details about how to change parameter values, see **Editing a Parameter Template**.

2. How do I purchase a cluster instance?

   For details, see **Buying a Cluster Instance**.

3. How do I change the CPU or memory of a cluster instance?

   You can change the CPU and memory of dds mongos and shards nodes. For details, see **Changing the CPU and Memory of a Cluster Instance**.

4. How do I scale up the storage space of a cluster instance?

   You can scale up the storage of shards nodes. For details, see **Scaling Up Storage**.

8.2.2 Replica Set

Instances of the same type have different memory specifications. You can select instances of different specifications based on application scenarios.

This section describes replica set instance specifications. The instance specifications depend on the selected CPU model.

NOTE

The default maximum number of connections has been adjusted. Query the number of connections based on the instance creation time. For details about the number of connections for instances created before July 2021, see **Historical Instance Specifications**.

Instance Specifications

DDS replica set instances have two specification types: general-purpose and enhanced II. For details, see **Table 8-7**.

Table 8-7 Specification type

Specification Type	Description	Applicable Scenario
General-purpose	CPU resources are shared with other general-purpose DB instances on the same physical machine. CPU usage is maximized through resource overcommitment. This instance class is a cost-effective option and suitable for scenarios where performance stability is not critical.	Suitable for scenarios that have high requirements on cost-effectiveness.

Specification Type	Description	Applicable Scenario
Enhanced II	With a leading network acceleration engine and Data Plane Development Kit (DPDK) fast packet processing mechanism, this instance class provides higher network performance and computing power.	Suitable for websites and web applications that require high database computing and network performance.

Table 8-8 lists the specifications of replica set instances.

Table 8-8 Replica set instance specifications

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
x86	General-purpose	2	4	dds.mongodb.s6.large.2.repset	1000	200-1000
		2	8	dds.mongodb.s6.large.4.repset	1000	200-1000
		4	8	dds.mongodb.s6.xlarge.2.repset	3000	200-3000
		4	16	dds.mongodb.s6.xlarge.4.repset	3000	200-3000
		8	16	dds.mongodb.s6.2xlarge.2.repset	8000	200-8000
		8	32	dds.mongodb.s6.2xlarge.4.repset	8000	200-8000
	Enhanced II	2	8	dds.mongodb.c6.large.4.repset	1000	200-1000
		2	16	dds.mongodb.c6.large.8.repset	1000	200-1000
		4	16	dds.mongodb.c6.xlarge.4.repset	3000	200-3000

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
		4	32	dds.mongodb.c6.xlarge.8.repset	3000	200-3000
		8	32	dds.mongodb.c6.2xlarge.4.repset	8000	200-8000
		8	64	dds.mongodb.c6.2xlarge.8.repset	8000	200-8000
		16	64	dds.mongodb.c6.4xlarge.4.repset	16000	200-16000
		16	128	dds.mongodb.c6.4xlarge.8.repset	16000	200-16000
		32	128	dds.mongodb.c6.8xlarge.4.repset	16000	200-16000
		32	256	dds.mongodb.c6.8xlarge.8.repset	16000	200-16000
		64	256	dds.mongodb.c6.16xlarge.4.repset	16000	200-16000
Kunpeng	-	2	4	dds.mongodb.large.arm2.repset	1000	200-1000
	-	2	8	dds.mongodb.large.arm4.repset	1000	200-1000
	-	4	8	dds.mongodb.xlarge.arm2.repset	3000	200-3000
	-	4	16	dds.mongodb.xlarge.arm4.repset	3000	200-3000
	-	8	16	dds.mongodb.2xlarge.arm2.repset	8000	200-8000
	-	8	32	dds.mongodb.2xlarge.arm4.repset	8000	200-8000
	-	16	32	dds.mongodb.4xlarge.arm2.repset	16000	200-16000
	-	16	64	dds.mongodb.4xlarge.arm4.repset	16000	200-16000

Historical Instance Specifications

For details about the range of connections of instances created before July 2021, see the following table.

Table 8-9 Replica set instance specifications

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
x86	General-purpose	2	4	dds.mongodb.s6.large.2.repset	400	200-1000
		2	8	dds.mongodb.s6.large.4.repset	400	200-1000
		4	8	dds.mongodb.s6.xlarge.2.repset	1000	200-3000
		4	16	dds.mongodb.s6.xlarge.4.repset	1000	200-3000
		8	16	dds.mongodb.s6.2xlarge.2.repset	4000	200-8000
		8	32	dds.mongodb.s6.2xlarge.4.repset	4000	200-8000
	Enhanced	2	8	dds.mongodb.c3.large.4.repset	400	200-1000
		4	16	dds.mongodb.c3.xlarge.4.repset	1000	200-3000
		8	32	dds.mongodb.c3.2xlarge.4.repset	4000	200-8000
		16	64	dds.mongodb.c3.4xlarge.4.repset	8000	200-16000
		32	128	dds.mongodb.c3.8xlarge.4.repset	8000	200-16000
		60	256	dds.mongodb.c3.15xlarge.4.repset	8000	200-16000

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
	Enhanced II	2	8	dds.mongodb.c6.large.4.repset	400	200-1000
		4	16	dds.mongodb.c6.xlarge.4.repset	1000	200-3000
		8	32	dds.mongodb.c6.2xlarge.4.repset	4000	200-8000
		16	64	dds.mongodb.c6.4xlarge.4.repset	8000	200-16000
		32	128	dds.mongodb.c6.8xlarge.4.repset	8000	200-16000
		64	256	dds.mongodb.c6.16xlarge.4.repset	8000	200-16000
Kunpeng	-	2	4	dds.mongodb.large.arm2.repset	400	200-1000
	-	2	8	dds.mongodb.large.arm4.repset	400	200-1000
	-	4	8	dds.mongodb.xlarge.arm2.repset	1000	200-3000
	-	4	16	dds.mongodb.xlarge.arm4.repset	1000	200-3000
	-	8	16	dds.mongodb.2xlarge.arm2.repset	4000	200-8000
	-	8	32	dds.mongodb.2xlarge.arm4.repset	4000	200-8000
	-	16	32	dds.mongodb.4xlarge.arm2.repset	8000	200-16000
	-	16	64	dds.mongodb.4xlarge.arm4.repset	8000	200-16000

Helpful Links

1. How do I change the maximum number of connections of a replica set instance?

You can change the maximum number of connections to an instance by modifying the **net.maxIncomingConnections** parameter. For details about how to change parameter values, see **Editing a Parameter Template**.

2. How do I buy a replica set instance?

   For details, see **Buying a Replica Set Instance**.

3. How do I change the CPU and memory of a replica set instance?

   For details, see **Changing the CPU or Memory of a Replica Set Instance**.

4. How do I scale up the storage space of a replica set instance?

   For details, see **Scaling Up Storage**.

8.2.3 Single Node

NOTE

Huawei Cloud has discontinued the sale of DDS single node instances since July 15, 2023.

Instances of the same type have different memory specifications. You can select instances of different specifications based on application scenarios.

This section describes single node instance specifications. The instance specifications depend on the selected CPU model.

NOTE

The default maximum number of connections has been adjusted. Query the number of connections based on the instance creation time. For details about the number of connections for instances created before July 2021, see **Historical Instance Specifications**.

Instance Specifications

DDS single node instances have two specification types: general-purpose and enhanced II. For details, see **Table 8-10**.

Table 8-10 Specification type

Specification Type	Description	Applicable Scenario
General-purpose	CPU resources are shared with other general-purpose DB instances on the same physical machine. CPU usage is maximized through resource overcommitment. This instance class is a cost-effective option and suitable for scenarios where performance stability is not critical.	Suitable for scenarios that have high requirements on cost-effectiveness.

Specification Type	Description	Applicable Scenario
Enhanced II	With a leading network acceleration engine and Data Plane Development Kit (DPDK) fast packet processing mechanism, this instance class provides higher network performance and computing power.	Suitable for websites and web applications that require high database computing and network performance.

Table 8-11 Single Node

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
x86	General-purpose	1	4	dds.mongodb.s6.medium.4.single	500	200-500
		2	4	dds.mongodb.s6.large.2.single	1000	200-1000
		2	8	dds.mongodb.s6.large.4.single	1000	200-1000
		4	8	dds.mongodb.s6.xlarge.2.single	8000	200-8000
		4	16	dds.mongodb.s6.xlarge.4.single	8000	200-8000
		8	16	dds.mongodb.s6.2xlarge.2.single	10000	200-10000
		8	32	dds.mongodb.s6.2xlarge.4.single	10000	200-10000
	Enhanced II	1	8	dds.mongodb.c6.medium.8.single	500	200-500
		2	8	dds.mongodb.c6.large.4.single	1000	200-1000
		2	16	dds.mongodb.c6.large.8.single	1000	200-1000
		4	16	dds.mongodb.c6.xlarge.4.single	8000	200-8000

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
		4	32	dds.mongodb.c6.xlarge.8.single	8000	200-8000
		8	32	dds.mongodb.c6.2xlarge.4.single	10000	200-10000
		8	64	dds.mongodb.c6.2xlarge.8.single	10000	200-10000
		16	64	dds.mongodb.c6.4xlarge.4.single	16000	200-16000
		16	128	dds.mongodb.c6.4xlarge.8.single	16000	200-16000
		32	128	dds.mongodb.c6.8xlarge.4.single	16000	200-16000
		64	256	dds.mongodb.c6.16xlarge.4.single	16000	200-16000
Kunpeng	-	2	4	dds.mongodb.large.arm2.single	1000	200-1000
	-	2	8	dds.mongodb.large.arm4.single	1000	200-1000
	-	4	8	dds.mongodb.xlarge.arm2.single	8000	200-8000
	-	4	16	dds.mongodb.xlarge.arm4.single	8000	200-8000
	-	8	16	dds.mongodb.2xlarge.arm2.single	10000	200-10000
	-	8	32	dds.mongodb.2xlarge.arm4.single	10000	200-10000
	-	16	32	dds.mongodb.4xlarge.arm2.single	16000	200-16000
	-	16	64	dds.mongodb.4xlarge.arm4.single	16000	200-16000

Historical Instance Specifications

For details about the range of connections of instances created before July 2021, see the following table.

Table 8-12 Single Node

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
x86	General-purpose	1	4	dds.mongodb.s6.medium.4.single	400	200-500
		2	4	dds.mongodb.s6.large.2.single	400	200-1000
		2	8	dds.mongodb.s6.large.4.single	400	200-1000
		4	8	dds.mongodb.s6.xlarge.2.single	1000	200-8000
		4	16	dds.mongodb.s6.xlarge.4.single	1000	200-8000
		8	16	dds.mongodb.s6.2xlarge.2.single	4000	200-10000
		8	32	dds.mongodb.s6.2xlarge.4.single	4000	200-10000
	Enhanced	2	8	dds.mongodb.c3.large.4.single	400	200-1000
		4	16	dds.mongodb.c3.xlarge.4.single	1000	200-8000
		8	32	dds.mongodb.c3.2xlarge.4.single	4000	200-10000
		16	64	dds.mongodb.c3.4xlarge.4.single	8000	200-16000
		32	128	dds.mongodb.c3.8xlarge.4.single	8000	200-16000
		60	256	dds.mongodb.c3.15xlarge.4.single	8000	200-16000

CPU Type	Specifications	vCPUs	Memory (GB)	Flavor ID	Default Maximum Number of Connections	Range of Connections
	Enhanced II	2	8	dds.mongodb.c6.large.4.single	400	200-500
		4	16	dds.mongodb.c6.xlarge.4.single	1000	200-1000
		8	32	dds.mongodb.c6.2xlarge.4.single	4000	200-4000
		16	64	dds.mongodb.c6.4xlarge.4.single	8000	200-8000
		32	128	dds.mongodb.c6.8xlarge.4.single	8000	200-16000
		64	256	dds.mongodb.c6.16xlarge.4.single	8000	200-16000
Kunpeng	-	2	4	dds.mongodb.large.arm2.single	400	200-1000
	-	2	8	dds.mongodb.large.arm4.single	400	200-1000
	-	4	8	dds.mongodb.xlarge.arm2.single	1000	200-8000
	-	4	16	dds.mongodb.xlarge.arm4.single	1000	200-8000
	-	8	16	dds.mongodb.2xlarge.arm2.single	4000	200-10000
	-	8	32	dds.mongodb.2xlarge.arm4.single	4000	200-10000
	-	16	32	dds.mongodb.4xlarge.arm2.single	8000	200-16000
	-	16	64	dds.mongodb.4xlarge.arm4.single	8000	200-16000

Helpful Links

1. How do I change the maximum number of connections of a single node instance?

   You can change the maximum number of connections to an instance by modifying the **net.maxIncomingConnections** parameter. For details about how to change parameter values, see **Editing a Parameter Template**.

2. How do I change the CPU and memory of a single node instance?

   For details, see **Changing the CPU or Memory of a Single Node Instance**.

3. How do I scale up the storage of a single node instance?

   For details, see **Scaling Up Storage**.

8.3 Read Replica

To improve the read capability of the primary node, DDS provides read replicas that can be accessed independently, which is good for handling read-intensive workloads and reducing memory pressure on the primary node. You can also create one or more read replicas to process a large number of read requests and improve throughput.

Differences Between Read Replica and Secondary Node

Node	Description	Application Scenario
Secondary nodes	The secondary node ensures high availability. If a primary node fails, the system automatically switches traffic to the secondary node. If the primary node is faulty, each secondary node may be elected as a new primary node to execute data write requests.	DDS provides an HA connection address to connect to both the primary and secondary nodes and separate read and write operations. In this way, a DDS instance can handle heavy read workloads with ease. In addition, the impact of node faults on services is wakened while the performance is improved.
Read replicas in a replica set instance	A read replica in a replica set instance synchronizes data only from the primary node. It does not ensure high availability and will not be elected as the primary node. A read replica can be accessed through an independent connection. It will not interfere with connections between your applications and the primary and secondary nodes of a replica set instance.	Read replicas are useful when applications need to read large amounts of data from existing instances.

Node	Description	Application Scenario
Read replicas in a cluster instance	Read replicas in a cluster instance ensure high availability. If a read replica fails, the system automatically switches workloads to another available read replica or secondary node. A read replica in a cluster instance can be accessed through an independent connection. It will not interfere with connections between your applications and the primary and secondary nodes of a cluster instance. All read replicas can be connected using a unified address, so you can expand the read capability of your applications by adding read replicas without modifying the connection address. A read replica in a cluster instance will not be elected as the primary node.	Read replicas are for scenarios where applications need to read large amounts of data from existing instances, read request load balancing is required, and read replica HA is required.

Constraints

- Only replica set instances of versions 3.4, 4.0, 4.2, 4.4 and 5.0 and cluster instances of version 3.4 can have read replicas.
- Instances that only have read replicas cannot be created.
- Read replicas are used only for processing read requests and do not participate in primary/secondary node election.
- A maximum of five read replicas can be added to a replica set instance. For details, see **Adding Read Replicas to a Replica Set Instance**.
- Up to five read replicas can be added to each shard of a cluster DB instance.
- Data is asynchronously replicated between the primary node and the read replica, and this process will delay by a few seconds.

Functions

- You can change the number of read replicas to meet fluctuating service demands and reduce service costs.
- The specifications of a read replica are the same as those of the primary node. Data is automatically synchronized to the read replica.
- Resources on the primary node are not occupied. Adding or deleting read replicas does not affect services on the primary node.

- Read replicas can be connected independently and do not interfere with existing primary and secondary nodes. For details about read replicas in a replica set instance, see **Connecting to Read Replicas Using Mongo Shell**.
- You can monitor the performance of read replicas.

Billing

- After a read replica is added to the current instance, you will be billed for using the VM and disks of the read replica.
- The price of a read replica is equal to that of a single node in a replica set instance or a shard node in a cluster instance. For example, if the price of a three-node replica set instance is USD $3,000, the price of a read replica is USD $1,000.

9 Notes and Constraints

Before using DDS, you must familiarize yourself with the constraints of different types of DDS DB instances to help you select the instance architectures that best suit your business scenarios.

Table 9-1 Constraints on cluster instances

Function	Constraints
Instance Deployment	Cluster DB instances cannot be deployed in all regions and AZs.
Database Minor Version Upgrade	● The instance version must be 3.4, 4.0, 4.2, 4.4 or 5.0. ● If the instance status is abnormal or the instance is being operated, the upgrade cannot be performed. ● The upgrade cannot be performed if there are abnormal instance nodes.

Function	Constraints
Instance Modifications	- Scaling Up Storage Space **NOTE** - Only shard nodes can be scaled up. dds mongos and config nodes cannot be scaled up. - Storage space can only be scaled up. It cannot be scaled down. - If the storage space is about to be used up, the database is set to read-only, and data cannot be modified. - The maximum storage space of a single shard is 5000 GB. If the storage space exceeds 5000 GB, scale-up will fail. - Adding and Reverting Instance Nodes **NOTE** - You can add nodes for a DB instance only when the DB instance is normal. - You can roll back a node only when the node fails to be added. - Changing a DB Instance Class **NOTE** - When the instance class is being changed, a primary/secondary switchover may occur once or twice and the database connection will be interrupted each time for up to 30s. You are advised to change the class during off-peak hours to reduce impacts and ensure that the service system of your client can reconnect to the database if the connection is interrupted.
Data Backups	Physical and snapshot backup methods are supported.
Data Restorations	- Backup data can be restored to a new DB instance. - Backup data can be restored to the original DB instance. - Backup data can be restored to a specified point in time. - Backup data can be restored to an on-premises database.
Data Security	SSL encryption can be set. **NOTE** Enabling or disabling SSL will cause instances to restart. Exercise caution when performing this operation.
Data Migration	DDS provides multiple migration schemes to migrate MongoDB databases in different service scenarios. For details, see **Migration Scheme Overview**. - Migrating data using the export and import tools - Migrating data from other cloud MongoDB to DDS - Migrating data from on-premises MongoDB to DDS - Migrating data from ECS-hosted MongoDB to DDS - Migrating data from DDS to MongoDB

Function	Constraints
Logs	Error logs, slow query logs, and audit logs are supported.
Billing	DDS allows you to pay only for what you use. There is no minimum fee requirement. For more information, see **Billing**.
Connections	- You can enable shard and config IP addresses. - You can change a private IP address of a DB instance. - You can change a database port. **NOTE** The shard node port is 8637, and the config node port is 8636, which cannot be changed. - You can apply for and modify a private domain name for database connections.
Recycle Bin	- Instances in the recycle bin are retained for 7 days by default. - Up to 100 instances can be moved to the recycle bin. Once the recycle bin is full, you can still delete instances, but they cannot be placed in the recycle bin.

Table 9-2 Constraints on replica set instances

Function	Constraints
Instance Deployment	Replica set DB instances cannot be deployed in all regions and AZs.
Database Minor Version Upgrade	- The instance version must be 3.4, 4.0, 4.2, 4.4 or 5.0. - If the instance status is abnormal or the instance is being operated, the upgrade cannot be performed. - The upgrade cannot be performed if there are abnormal instance nodes. - Read replicas do not support minor version upgrade.

Function	Constraints
Instance Modifications	● Scaling Up Storage Space **NOTE** ● Storage space can only be scaled up. It cannot be scaled down. ● If the storage space is about to be used up, the database is set to read-only, and data cannot be modified. ● Adding and Reverting Instance Nodes **NOTE** ● You can add nodes for a DB instance only when the DB instance is normal. ● You can roll back a node only when the node fails to be added. ● Deleting Instance Nodes or Read Replicas **NOTE** ● Nodes cannot be deleted from instances that have abnormal nodes. ● Changing a DB Instance Class **NOTE** ● When the instance class is being changed, a primary/secondary switchover may occur once or twice and the database connection will be interrupted each time for up to 30s. You are advised to change the class during off-peak hours to reduce impacts and ensure that the service system of your client can reconnect to the database if the connection is interrupted.
Data Backups	Physical and snapshot backup methods are supported.
Data Restorations	● Backup data can be restored to a new DB instance. ● Backup data can be restored to the original DB instance. ● Backup data can be restored to a specified point in time. ● Backup data can be restored to an on-premises database.
Data Security	SSL encryption can be set. **NOTE** Enabling or disabling SSL will cause instances to restart. Exercise caution when performing this operation.

Function	Constraints
Data Migration	DDS provides multiple migration schemes to migrate MongoDB databases in different service scenarios. For details, see **Migration Scheme Overview**. ● Migrating data using the export and import tools ● Migrating data from other cloud MongoDB to DDS ● Migrating data from on-premises MongoDB to DDS ● Migrating data from ECS-hosted MongoDB to DDS ● Migrating data from DDS to MongoDB
Logs	● Error logs, slow query logs, and audit logs are supported. ● Read replicas do not support slow query logs and error logs.
Billing	DDS allows you to pay only for what you use. There is no minimum fee requirement. For more information, see **Billing**.
Connections	● You can configure access across CIDR blocks. ● You can change a private IP address of a DB instance. ● You can change a database port. **NOTE** The default port is 8635. You can change the port after the instance is created. ● You can apply for and modify a private domain name for database connections.
Recycle Bin	● Instances in the recycle bin are retained for 7 days by default. ● Up to 100 instances can be moved to the recycle bin. Once the recycle bin is full, you can still delete instances, but they cannot be placed in the recycle bin.

Table 9-3 Constraints on single node instances

Function	Constraints
Instance Deployment	Single node DB instances cannot be deployed in all regions and AZs.
Database Minor Version Upgrade	Not supported.

Function	Constraints
Instance Modifications	● Scaling Up Storage Space **NOTE** ● Storage space can only be scaled up. It cannot be scaled down. ● If the storage space is about to be used up, the database is set to read-only, and data cannot be modified. ● Changing a DB Instance Class **NOTE** ● When the instance class is being changed, the database connection will be interrupted for 5 to 10 minutes. You are advised to change the class during off-peak hours to reduce impacts and ensure that the service system of your client can reconnect to the database if the connection is interrupted.
Data Backups	Logical and snapshot backup methods are supported.
Data Restorations	● Backup data can be restored to a new DB instance. ● Backup data can be restored to the original DB instance. ● Backup data can be restored to an on-premises database.
Data Security	SSL encryption can be set. **NOTE** Enabling or disabling SSL will cause instances to restart. Exercise caution when performing this operation.
Data Migration	DDS provides multiple migration schemes to migrate MongoDB databases in different service scenarios. For details, see **Migration Scheme Overview**. ● Migrating data using the export and import tools ● Migrating data from other cloud MongoDB to DDS ● Migrating data from on-premises MongoDB to DDS ● Migrating data from ECS-hosted MongoDB to DDS ● Migrating data from DDS to MongoDB
Logs	Error logs, slow query logs, and audit logs are supported.
Billing	DDS allows you to pay only for what you use. There is no minimum fee requirement. For more information, see **Billing**.
Connections	● You can change a private IP address of a DB instance. ● You can change a database port. **NOTE** The default port is 8635. You can change the port after the instance is created. ● You can apply for and modify a private domain name for database connections.

Function	Constraints
Recycle Bin	● Instances in the recycle bin are retained for 7 days by default. ● Up to 100 instances can be moved to the recycle bin. Once the recycle bin is full, you can still delete instances, but they cannot be placed in the recycle bin.

10 Basic Concepts

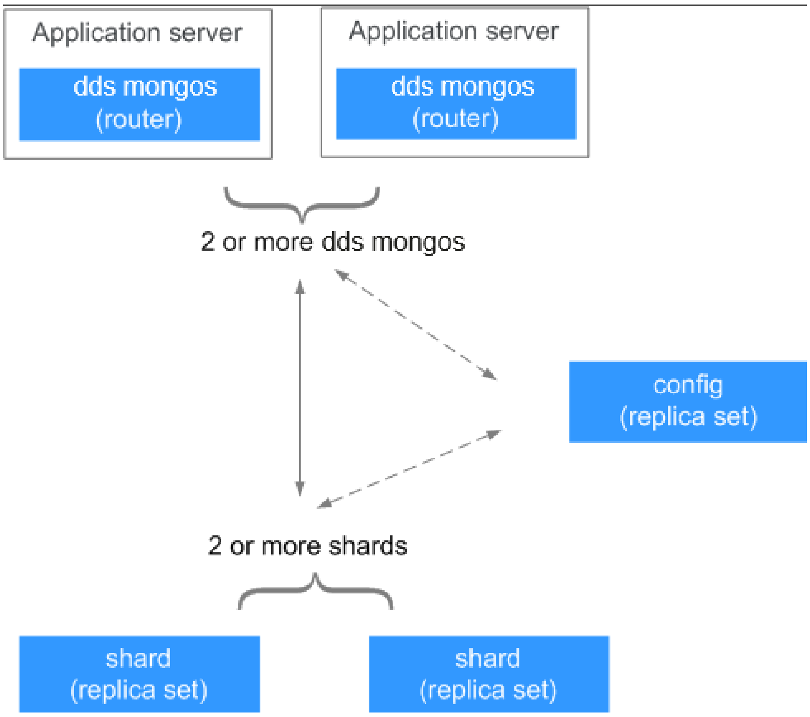
10.1 Instance

- An instance is a basic management unit of DDS. One or more databases can be created in a single DDS instance, and one or more collections can be created in each database.
- You can create and manage DB instances of various types and versions on the management console. For details about DB instance types, versions, and specifications, see **Instances**.
- Resources, such as the CPU, memory, and I/O, of each DDS DB instance are isolated from each other.

10.2 Cluster

Each DDS cluster consists of a config node, and multiple dds mongos and shard nodes. The following diagram shows the node relationships.

Figure 10-1 Diagram of node relationships



dds mongos

A dds mongos is a router for reading and writing data, providing a unified interface for accessing DB instances.

- Each DB instance has 2 to 16 dds mongos nodes. You can specify the quantity.
- A dds mongos reads configuration settings from the config node and allocates read and write requests to shard nodes. You can connect to a dds mongos directly.

config

A config stores configuration settings for DB instances and consists of one replica set.

You cannot connect to a config node directly.

shard

Shards are used to store user data.

- Each Community Edition instance has 2 to 16 shard nodes. You can specify the quantity.
- Each shard node is deployed as a replica set to ensure data redundancy and high reliability.
- You cannot connect to a shard node directly.

10.3 Database Parameter Template

A database parameter template is a collection of configuration parameters and values and can be applied to multiple DB instances.

10.4 Region and AZ

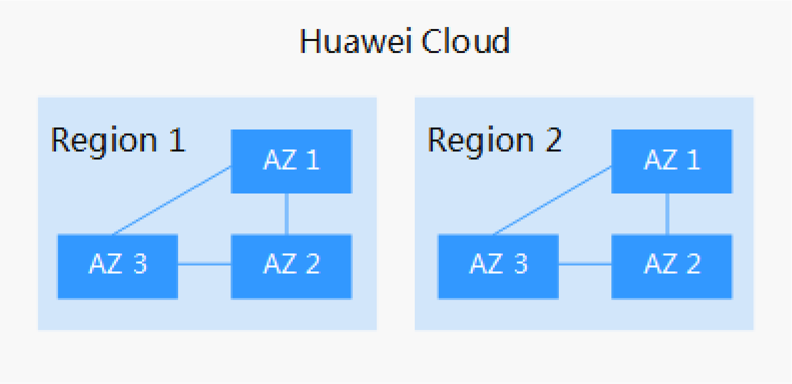
Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- Regions are defined by their geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. There are universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region only provides services of a specific type and only or for specific tenants.
- An AZ contains one or multiple physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electrical facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

Figure 10-2 shows the relationship between regions and AZs.

Figure 10-2 Region and AZ



Huawei Cloud provides services in many regions around the world. You can select a region and an AZ as needed. For more information, see **Huawei Cloud Global Regions**.

Selecting a Region

When selecting a region, consider the following factors:

- Location

  You are advised to select a region close to you or your target users. This reduces network latency and improves the access rate. However, since all Chinese mainland regions provide basically the same infrastructure, BGP network quality, as well as operations and resource configurations, any of them will work well if your users are located there.

- Resource price

  Resource prices may vary in different regions. For details, see **Product Pricing Details**.

Selecting an AZ

When determining whether to deploy resources in the same AZ, consider your applications' requirements for disaster recovery (DR) and network latency.

- For better DR resilience, deploy resources in different AZs in the same region.
- For lower network latency, deploy resources in the same AZ.

Regions and Endpoints

Before using an API to call resources, specify its region and endpoint. For more details, see **Regions and Endpoints**.

10.5 DB Connections

The number of connections is the number of databases that can be simultaneously connected to an application. The number of connections is irrelevant to the maximum number of users allowed by your applications or websites.

- For a cluster instance, the number of connections is the number of connections between the client and the dds mongos nodes.
- For a replica set instance, the number of connections is the number of connections between the client and the primary and secondary nodes.
- For a single-node instance, the number of connections is the number of connections between the client and the node.

10.6 Memory

The maximum available memory that can be used by a DDS DB instance.

10.7 Storage Space

The storage space in GB specified when you purchase a DDS DB instance.

- Both your collection data and the data required for the operation of your DB instance such as system database data and indexes occupy the storage space on your purchased DB instance.

- Ensure that your DDS DB instance has sufficient storage space, or your DB instance may be read-only. If your DB instance is read-only due to insufficient storage space, you can scale up the storage space.

11 Compatibility

11.1 Versions

This section describes the compatibility between DDS versions and MongoDB, aggregation operation compatibility, and DDS features of different versions.

MongoDB-Compatible Features

Table 11-1 MongoDB-compatible features

Category	Details	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	Community Edition MongoDB 4.2	Community Edition MongoDB 4.4	Community Edition MongoDB 5.0
Database commands	The find command supports the **allowDiskUse** field, which allows MongoDB to use temporary files on disk to store data exceeding the system memory limit while processing a non-index sort operation.	×	×	√	√	×	√	√

Category	Details	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	Community Edition MongoDB 4.2	Community Edition MongoDB 4.4	Community Edition MongoDB 5.0
	New syntax and usage of projection. ● In projection, syntax of aggregation functions is supported, for example, using aggregation operators. ● In projection, data is encoded in JSON format and nested to map specified fields. ● In projection, the **$** character can be used to specify a specific index subelement for a mapped array element.	×	×	√	√	×	√	√
Time series collections	Time series collections are supported.	×	×	×	√	×	×	√

Category	Details	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	Community Edition MongoDB 4.2	Community Edition MongoDB 4.4	Community Edition MongoDB 5.0
	Geospatial indexes can be created for metaField.	×	×	×	√	×	×	√
	Constrained update and deletion operations are supported.	×	×	×	√	×	×	√
	Sharded time series collections are supported.	×	×	×	×	×	×	×
	Time series collections can be compressed.	×	×	×	√	×	×	√
Distributed transactions	Distributed transactions are supported. Distributed transactions refer to multi-document transactions on sharded clusters and replica sets. Multi-document transactions (whether on sharded clusters or replica sets) are also known as distributed transactions starting in MongoDB 4.2.	×	√	√	√	√	√	√

Category	Details	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	Community Edition MongoDB 4.2	Community Edition MongoDB 4.4	Community Edition MongoDB 5.0
	The 16 MB total size limit for a transaction was removed.	×	×	√	√	√	√	√
	The MongoDB drivers must be updated for MongoDB 4.2.	×	√	√	√	√	√	√
Aggregation	- The aggregation capability is improved. Data can be updated in the aggregation pipeline. - Added $merge. - Added $accumulator and $function. - Supported union All. - Added $unionWith. - For details about aggregation operations, see **Table 11-2**.	√	√	√	√	√	√	√
Wildcard indexes	Wildcard indexes are supported.	√	√	√	√	√	√	√

Category	Details	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	Community Edition MongoDB 4.2	Community Edition MongoDB 4.4	Community Edition MongoDB 5.0
Sharded cluster	A document's shard key field value can be changed.	×	×	√	√	√	√	√
	The fields associated with the shard key can be adjusted.	√	√	√	√	×	√	√
	Collections can be sharded using a composite shard key that consists of a single hash field.	×	×	√	√	×	√	√
	The 512-byte size limit on a shard key was removed.	×	√	√	√	×	√	√
	Multi-document transactions allow the creation of indexes and collections (non-distributed transactions).	×	×	√	√	×	√	√
	The **$lookup** and **$graphLookup** operation supports sharded collections.	×	×	√	√	×	×	√

Category	Details	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	Community Edition MongoDB 4.2	Community Edition MongoDB 4.4	Community Edition MongoDB 5.0
Indexes	The size limit on indexes was removed.	√	√	√	√	√	√	√
	The length limit on index names was removed.	√	√	√	√	√	√	√
	Composite indexes can contain hash indexes.	×	×	√	√	×	√	√
	Hidden indexes are supported for commissioning.	√	√	√	√	×	√	√
	The size limit on collection namespaces is changed. The length of <database>.<collection> is increased from 120 bytes to 255 bytes.	√	√	√	√	√	√	√
Other optimized features	Default Read and Write Concern Global default read/write concerns, involving connection and command read/write contexts.	×	×	√	√	×	×	√

Category	Details	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	Community Edition MongoDB 4.2	Community Edition MongoDB 4.4	Community Edition MongoDB 5.0
	Jumbo chunks can be migrated.	×	×	√	√	×	×	√
	Hedge query is supported.	×	×	√	√	×	×	√
	The dds mongos node supports JavaScripts to support some special aggregation operators.	×	×	√	√	×	×	√
	In non-multi-document transactions, some read operations support read concern **snapshot**.	×	×	×	×	×	×	√
	Using **collMod** to change the timeout interval of TTL indexes of a single field is supported.	×	×	×	√	×	×	√

NOTE

- **√** indicates that an item is supported, and **×** indicates that an item is not supported, and **N.A.** indicates that an item is not applicable.
- For details about how to evaluate the compatibility when migrating MongoDB from a later version to an earlier version, see **How Do I Evaluate the Compatibility When Migrating MongoDB from a Later Version to an Earlier Version?**

Aggregation Operations

Table 11-2 Aggregation operations

Operator	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	MongoDB 4.2 Community Edition	MongoDB 4.4 Community Edition	MongoDB 5.0 Community Edition
$sin	√	√	√	√	√	√	√
$cos	√	√	√	√	√	√	√
$tan	√	√	√	√	√	√	√
$asin	√	√	√	√	√	√	√
$acos	√	√	√	√	√	√	√
$atan	√	√	√	√	√	√	√
$atan2	√	√	√	√	√	√	√
$asinh	√	√	√	√	√	√	√
$acosh	√	√	√	√	√	√	√
$atanh	√	√	√	√	√	√	√
$degreesToRadians	√	√	√	√	√	√	√
$radiansToDegrees	√	√	√	√	√	√	√
$round	√	√	√	√	√	√	√
$trunc	√	√	√	√	√	√	√
$regexFind	√	√	√	√	√	√	√
$regexFindAll	√	√	√	√	√	√	√
$regexMatch	√	√	√	√	√	√	√
$merge	√	√	√	√	√	√	√
$planCacheStats	×	×	×	×	√	√	×
$replaceWith	√	√	√	√	√	√	√
$set	√	√	√	√	√	√	√
$unset	√	√	√	√	√	√	√

Operator	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	MongoDB 4.2 Community Edition	MongoDB 4.4 Community Edition	MongoDB 5.0 Community Edition
NOW	√	√	√	√	√	√	√
CLUSTER_TIME	√	√	√	√	√	√	√
findAndModify	√	√	√	√	×	√	√
update	√	√	√	√	√	√	√
$accumulator	√	√	√	√	×	√	√
$binarySize	√	√	√	√	×	√	√
$bsonSize	√	√	√	√	×	√	√
$first	√	√	√	√	×	√	√
$function	√	√	√	√	×	√	√
$last	√	√	√	√	×	√	√
$isNumber	√	√	√	√	×	√	√
$replaceOne	√	√	√	√	×	√	√
$unionWith	√	√	√	√	×	√	√
$dateAdd	×	×	√	√	×	×	√
$dateSubtract	×	×	√	√	×	×	√
$dateDiff	×	×	√	√	×	×	√
$dateTrunc	×	×	√	√	×	×	√
$rand	×	×	×	√	×	×	√
$sampleRate	×	×	×	√	×	×	√
$getField	×	×	√	√	×	×	√
$setField	×	×	√	√	×	×	√
$unsetField	×	×	√	√	×	×	√

Operator	DDS 4.0	DDS 4.2	DDS 4.4	DDS 5.0	MongoDB 4.2 Community Edition	MongoDB 4.4 Community Edition	MongoDB 5.0 Community Edition
$setWindowFields	×	×	×	√	×	×	√
let	×	×	×	√	×	×	√
$expr (using indexes)	×	×	×	×	×	×	√
$ifNull	×	×	×	√	×	×	√

NOTE

- √ indicates that an item is supported, and × indicates that an item is not supported.
- For details about how to evaluate the compatibility when migrating MongoDB from a later version to an earlier version, see **How Do I Evaluate the Compatibility When Migrating MongoDB from a Later Version to an Earlier Version?**

DDS Features

Table 11-3 DDS features

Category	Feature	Constraint	DDS 4.0			DDS 4.2			DDS 4.4			DDS 5.0		Remarks
DB Instance Type			Single Node	Replica Set	Cluster	Single Node	Replica Set	Cluster	Single Node	Replica Set	Cluster	Replica set	Cluster	-
Instance Management	Creating an instance	-	√	√	√	√	√	√	√	√	√	√	√	-

	Deleting an instance	-	√	√	√	√	√	√	√	√	√	√	√	-
	Restarting an instance	-	√	√	√	√	√	√	√	√	√	√	√	-
	Resetting a password	-	√	√	√	√	√	√	√	√	√	√	√	-
	Changing an instance name	-	√	√	√	√	√	√	√	√	√	√	√	-
	Changing an instance port	-	√	√	√	√	√	√	√	√	√	√	√	-

	Performing a switchover	-	√	√	√	√	√	√	×	√	√	√	√	-
	Exporting the instance list	-	√	√	√	√	√	√	√	√	√	√	√	-
	Compute scale-up	-	√	√	√	√	√	√	√	√	√	√	√	-
	Compute scale-down	-	√	√	√	√	√	√	√	√	√	√	√	-
	Adding nodes	-	-	√	√	-	√	√	√	√	√	√	√	-
	Deleting nodes	-	-	×	×	-	×	×	×	×	×	×	×	-

	Storage scale-up	-	√	√	√	√	√	√	√	√	√	√	√	-
	Storage scale-down	-	×	×	×	×	×	×	×	×	×	×	×	-
	Restarting a node	-	√	√	√	√	√	√	√	√	√	√	√	-
	Open API	-	√	√	√	√	√	√	√	√	√	√	√	-
	SDK	-	√	√	√	√	√	√	√	√	√	√	√	-
	Task center	-	√	√	√	√	√	√	√	√	√	√	√	-
	Tags	-	√	√	√	√	√	√	√	√	√	√	√	-
	Quotas	-	√	√	√	√	√	√	√	√	√	√	√	-

Network Management	EIP-based public network access	-	√	√	√	√	√	√	√	√	√	√	√	-
	Changing a private IP address	-	√	√	√	√	√	√	√	√	√	√	√	-

	Cross-CIDR access	-	×	√	×	×	√	×	×	√	×	√	×	In DDS 4.2, 4.4 and 5.0, the client and an instance can communicate only when they are in the same VPC and subnet. If the

														client is a container, the client can communicate with an instance using an IP address in the default CIDR block 192.168.0.0/16, 17

														2.16.0.0/24, or 10.0.0.0/8.
	Changing an AZ	-	×	√	√	×	√	√	×	√	√	√	√	In DDS 4.2, 4.4 and 5.0, deploying an instance across multiple AZs is recommended.

	Domain name access	-	√	√	√	√	×	×	×	×	×	×	×	In DDS 4.2, 4.4 and 5.0, you are advised to use the IP address to access an instance.
Security	Security group	-	√	√	√	√	√	√	√	√	√	√	√	-
	SSL	-	√	√	√	√	√	√	√	√	√	√	√	-
	Instance recycle bin	-	√	√	√	√	√	√	√	√	√	√	√	-

	Storage encryption	-	√	√	√	×	×	×	×	×	×	×	×	-
Billing	Converting a single node to a replica set	-	×	-	-	×	-	-	×	-	-	-	-	-
	Pay-per-use	-	√	√	√	√	√	√	√	√	√	√	√	-

	Yearly/Monthly	-	√	√	√	√	√	√	√	√	√	√	×	In open beta testing (OBT), DDS 5.0 does not support the yearly/monthly billing.

	Changing pay-per-use to yearly/monthly	-	√	√	√	√	√	√	√	√	√	√	×	In open beta testing (OBT), DDS 5.0 does not support the yearly/monthly billing.

	Changing yearly/monthly to pay-per-use	-	√	√	√	√	√	√	√	√	√	√	×	In open beta testing (OBT), DDS 5.0 does not support the yearly/monthly billing.
Versions	Minor version upgrade (manual)	-	√	√	√	√	√	√	√	√	√	√	√	-

	Minor version upgrade (automatic)	-	×	×	×	×	×	×	×	×	×	×	×	-
	Major version upgrade (manual)	-	×	×	×	×	×	×	×	×	×	×	×	-
Logs	Viewing slow query logs	-	√	√	√	√	√	√	√	√	√	√	√	-
	Downloading slow query logs	-	√	√	√	√	√	√	√	√	√	√	√	-

	Masking slow query logs	-	√	√	√	√	√	√	√	√	√	√	√	-
	Querying error logs	-	√	√	√	√	√	√	√	√	√	√	√	-
	Downloading error logs	-	√	√	√	√	√	√	√	√	√	√	√	-
	Configuring the audit log policy	-	√	√	√	√	√	√	√	√	√	√	√	-
	Querying audit logs	-	√	√	√	√	√	√	√	√	√	√	√	-

	Downloading audit logs	-	√	√	√	√	√	√	√	√	√	√	√	-
Parameters	Changing a parameter template	-	√	√	√	√	√	√	√	√	√	√	√	-
	Creating a parameter template	-	√	√	√	√	√	√	√	√	√	√	√	-
	Deleting a parameter template	-	√	√	√	√	√	√	√	√	√	√	√	-

	Modifying a parameter template	-	√	√	√	√	√	√	√	√	√	√	√	-
	Comparing parameter templates	-	√	√	√	√	√	√	√	√	√	√	√	-
Backup and Restoration	Automated backup	-	√	√	√	√	√	√	√	√	√	√	√	-
	Manual backup	-	√	√	√	√	√	√	√	√	√	√	√	-
	Deleting a backup	-	√	√	√	√	√	√	√	√	√	√	√	-

	Restoring to a new DB instance	-	√	√	√	√	√	√	√	√	√	√	√	-
	Restoring data to the original DB instance	-	√	√	√	√	√	√	√	√	√	√	√	-
	Restoring to an existing DB instance	-	×	×	×	×	×	×	×	×	×	×	×	-

	Configuring an automated backup policy	-	√	√	√	√	√	√	√	√	√	√	√	-
	Incremental backup	-	√	√	√	×	√	√	×	√	√	√	√	-
	PITR	-	×	√	√	×	√	√	×	√	√	√	√	-
	Downloading a full backup file	-	√	√	√	√	√	√	√	√	√	√	√	-

	Downloading an incremental backup file	-	-	-	-	-	-	-	-	-	-	-	-	-
	Cross-region backup	-	√	√	√	√	√	√	√	√	√	√	√	-
	Cross-region restoration	-	√	√	√	√	√	√	√	√	√	√	√	-

O&M	Creating a database role	This operation can only be performed by calling a specified API.	√	√	√	√	√	√	√	√	√	√	√	-
	Creating a database account	This operation can only be performed by calling a specified API.	√	√	√	√	√	√	√	√	√	√	√	-

	Deleting a database role	This operation can only be performed by calling a specified API.	√	√	√	√	√	√	√	√	√	√	√	-
	Dropping a database user	This operation can only be performed by calling a specified API.	√	√	√	√	√	√	√	√	√	√	√	-

	Monitoring by seconds	Minimum interval: 5s	√	√	√	√	√	√	√	√	√	√	√	-

NOTE

√ indicates that an item is supported, and × indicates that an item is not supported.

11.2 Browsers

For details about the compatibility between DDS and browsers, see **Which Browsers Are Supported?**

12 Security

12.1 Shared Responsibilities

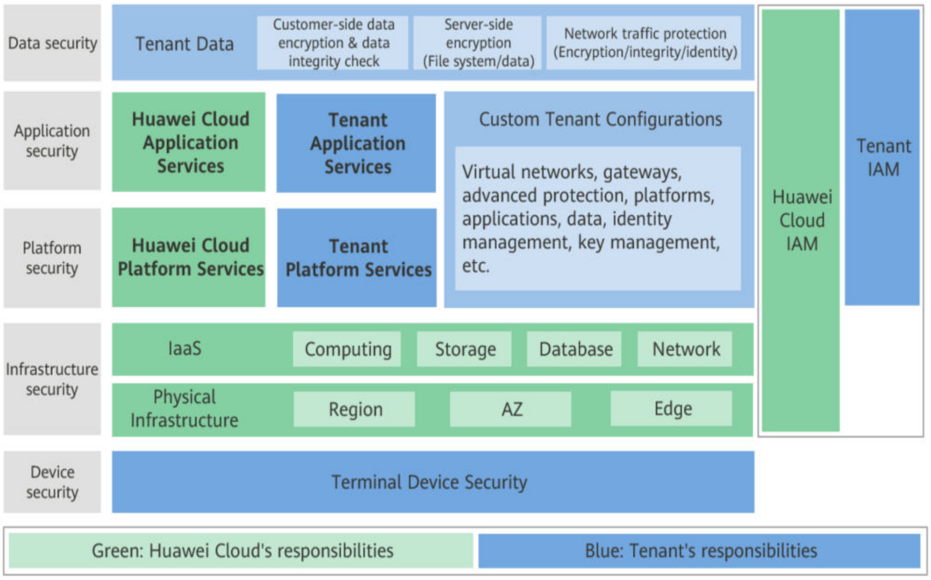
Huawei guarantees that its commitment to cyber security will never be outweighed by consideration of commercial interests. To address emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud has built a comprehensive cloud service security assurance system for different regions and industries. This system is based on Huawei's unique software and hardware advantages, and on relevant laws, regulations, industry standards, and on the security ecosystem as a whole.

The shared responsibility model for Huawei Cloud and the tenants who use Huawei Cloud services is illustrated in **Figure 12-1**. Responsibilities are as follows:

- **Huawei Cloud**: Ensure the security of cloud services and provide secure clouds. Huawei Cloud's security responsibilities include ensuring the security of our IaaS, PaaS, and SaaS services, as well as the physical environments of the Huawei Cloud data centers where our IaaS, PaaS, and SaaS services operate. Huawei Cloud is responsible for not only the security functions and performance of our infrastructure, cloud services, and technologies, but also for the overall cloud O&M security and, in more widely speaking, the security compliance of our infrastructure and services.
- **Tenant**: Use the cloud securely. Tenants of Huawei Cloud are responsible for the secure and effective management of the tenant-customized configurations of cloud services including IaaS, PaaS, and SaaS. This includes but is not limited to virtual networks, the OSs of virtual machine hosts and guests, virtual firewalls, API Gateway, advanced security services, all types of cloud services, tenant data, identity accounts, and key management.

Huawei Cloud Security White Paper elaborates on the ideas behind and measures used to ensure Huawei Cloud security, including cloud security strategies, the shared responsibility model, compliance and privacy, security organizations and personnel, infrastructure security, tenant service and security, engineering security, O&M security, and ecosystem security.

Figure 12-1 Huawei Cloud shared security responsibility model


Data security
Tenant Data
Customer-side data encryption & data integrity check
Server-side encryption (File system/data)
Network traffic protection (Encryption/integrity/identity)
Application security
Huawei Cloud Application Services
Tenant Application Services
Custom Tenant Configurations
Virtual networks, gateways, advanced protection, platforms, applications, data, identity management, key management, etc.
Platform security
Huawei Cloud Platform Services
Tenant Platform Services
Infrastructure security
IaaS
Computing
Storage
Database
Network
Physical Infrastructure
Region
AZ
Edge
Device security
Terminal Device Security
Huawei Cloud IAM
Tenant IAM
Green: Huawei Cloud's responsibilities
Blue: Tenant's responsibilities

12.2 Identity Authentication and Access Control

Identity Authentication

When you access DDS, the system authenticates your identity using password and IAM authentication.

- **Password Verification**

  To manage your instance, you need to use Data Admin Service (DAS) to log in to your instance. The login is successful only after your account and password are verified.

- **IAM Verification**

  You can use **Identity and Access Management (IAM)** to provide fine-grained control over DDS permissions. IAM provides identity authentication, permissions management, and access control, helping you efficiently manage access to your Huawei Cloud resources. IAM users can use DDS resources only after their accounts and passwords are verified. For details, see **Step 2: Create IAM Users and Log In**.

Access Control

- **Permissions control**

  If you need to assign different permissions to different employees in your enterprise to access your instance resources, IAM is a good choice. For details, see **Permissions**.

- **VPCs and subnets**

  A VPC is a logically isolated, configurable, and manageable virtual network. It helps improve the security of cloud resources and simplifies network deployment. You can define security groups, virtual private networks (VPNs), IP address ranges, and bandwidth for a VPC. This makes it easy for you to manage and configure private networks and improves network security.

A subnet provides dedicated network resources that are logically isolated from other networks for security.

For details, see **Creating a VPC**.

- **Security groups**

  A security group is a logical group that provides access control policies for the ECSs and DDS instances that have the same security protection requirements and are mutually trusted in a VPC. To ensure database security and reliability, you need to configure security group rules to allow only specific IP addresses and ports to access your DDS instances.

  For details, see **Configuring Security Group Rules**.

12.3 Data Protection

DDS provides a series of methods and features to ensure data security and reliability.

Table 12-1 Methods for data security

Method	Description	Reference
Transmission encryption (HTTPS)	HTTP and HTTPS are both supported, but HTTPS is recommended for enhanced security.	**Making an API Request**
Data backup	You can back up and restore databases to ensure data reliability.	**Backup Principles and Solutions**
Critical operation protection	With this function enabled, the system authenticates your identify when you perform critical operations like deleting an instance, to further secure your data and configurations.	**Critical Operation Protection**
SSL	You can use SSL to encrypt the connection between DDS and the client. It provides privacy, authentication, and integrity to Internet communications.	**Enabling or Disabling SSL**

12.4 Audit and Logs

Audit

Cloud Trace Service (CTS) records operations on cloud resources in your account. You can use the logs generated by CTS to perform security analysis, track resource changes, audit compliance, and locate faults.

For details about how to enable and configure CTS, see **Enabling CTS**.

With CTS, you can record operations associated with DDS for future query, audit, and backtracking. For details, see **Key Operations Recorded by CTS**.

Logs

DDS provides a variety of log types and functions for database analysis or audit. You can view logs on the management console.

- Error Logs

  DDS allows you to view database-level logs, including error logs and slow SQL query logs. For details, see **Error Logs**.

- Slow Query Logs

  Slow query logs record statements that exceed **operationProfiling.slowOpThresholdMs** (500 ms by default). You can view log details and statistics to identify slow statements, so you can optimize them. For details, see **Slow Query Logs**.

- Audit Logs

  An audit log records operations performed on your databases and collections. The generated log files are stored in OBS. Auditing logs can enhance your database security and help you analyze the cause of failed operations. For details, see **Audit Logs**.

12.5 Risk Monitoring

Cloud Eye is a comprehensive monitoring platform for resources like cloud databases and cloud servers. It enables you to monitor resources, configure alarm rules, identify resource exceptions, and quickly respond to resource changes.

Monitoring Metrics

You can monitors resources and operations, such as CPU usage and network throughput using Cloud Eye. For details about supported monitoring metrics and how to create alarm rules, see **DDS Metrics**.

The monitoring interval can be 1 minute, 5 seconds, or 1 second. The default monitoring interval is 1 minute. To improve the accuracy of monitoring metrics, you can enable **Monitoring by Seconds**.

12.6 Fault Rectification

Automated backups are created during the backup time window of your DB instances. DDS saves automated backups based on the retention period (1 to 732 days) you specified. For details, see **Configuring an Automated Backup Policy**.

Based on your service requirements, you can:

- **Restoring a Cluster Backup to a New Instance**
- **Restoring a Cluster Backup to the Original Instance**
- **Restoring a Cluster Instance to a Point in Time**
- **Restoring a Cluster Backup to an On-premises Database**

Cross-Region Backup

DDS can store backup files in the storage space that is in a different region from the DB instance for disaster recovery. If a DB instance in a region is faulty, you can use the backup files in another region to restore data to a new DB instance. After you enable cross-region backup, the backup files are automatically stored in the region you specify.

Cross-AZ Disaster Recovery

An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through a private network. DDS supports multiple-AZ deployment for cross-AZ DR.

Failover

If a primary node becomes unavailable, DDS automatically fails over to a standby node.

NOTE

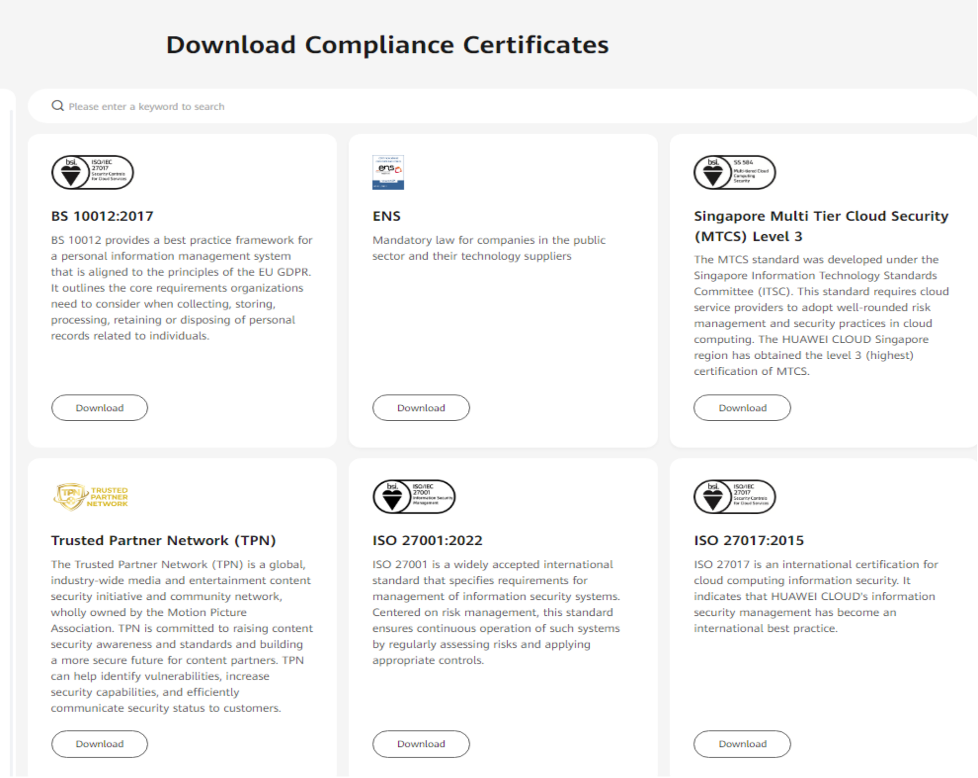
Only cluster and replica set instances support this function.

12.7 Certificates

Compliance Certificates

Huawei Cloud services and platforms have obtained various security and compliance certifications from authoritative organizations, such as International Organization for Standardization (ISO), system and organization controls (SOC), and Payment card industry (PCI) compliance standards. These certifications are available for **download**.

Figure 12-2 Downloading compliance certificates



Download Compliance Certificates

BS 10012:2017

BS 10012 provides a best practice framework for a personal information management system that is aligned to the principles of the EU GDPR. It outlines the core requirements organizations need to consider when collecting, storing, processing, retaining or disposing of personal records related to individuals.

ENS

Mandatory law for companies in the public sector and their technology suppliers

Singapore Multi Tier Cloud Security (MTCS) Level 3

The MTCS standard was developed under the Singapore Information Technology Standards Committee (ITSC). This standard requires cloud service providers to adopt well-rounded risk management and security practices in cloud computing. The HUAWEI CLOUD Singapore region has obtained the level 3 (highest) certification of MTCS.

Trusted Partner Network (TPN)

The Trusted Partner Network (TPN) is a global, industry-wide media and entertainment content security initiative and community network, wholly owned by the Motion Picture Association. TPN is committed to raising content security awareness and standards and building a more secure future for content partners. TPN can help identify vulnerabilities, increase security capabilities, and efficiently communicate security status to customers.

ISO 27001:2022

ISO 27001 is a widely accepted international standard that specifies requirements for management of information security systems. Centered on risk management, this standard ensures continuous operation of such systems by regularly assessing risks and applying appropriate controls.

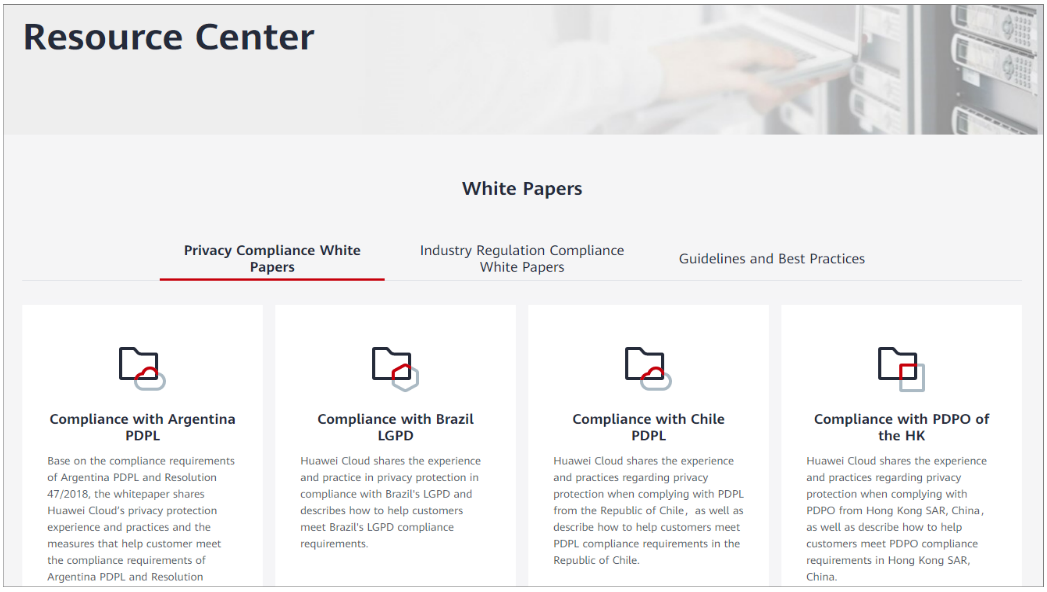
ISO 27017:2015

ISO 27017 is an international certification for cloud computing information security. It indicates that HUAWEI CLOUD's information security management has become an international best practice.

Resource Center

Huawei Cloud also provides the following resources to help users meet compliance requirements. For details, see **Resource Center**.

Figure 12-3 Resource center


Resource Center
White Papers
Privacy Compliance White Papers
Industry Regulation Compliance White Papers
Guidelines and Best Practices

Compliance with Argentina PDPL

Base on the compliance requirements of Argentina PDPL and Resolution 47/2018, the whitepaper shares Huawei Cloud's privacy protection experience and practices and the measures that help customer meet the compliance requirements of Argentina PDPL and Resolution

Compliance with Brazil LGPD

Huawei Cloud shares the experience and practice in privacy protection in compliance with Brazil's LGPD and describes how to help customers meet Brazil's LGPD compliance requirements.

Compliance with Chile PDPL

Huawei Cloud shares the experience and practices regarding privacy protection when complying with PDPL from the Republic of Chile， as well as describe how to help customers meet PDPL compliance requirements in the Republic of Chile.

Compliance with PDPO of the HK

Huawei Cloud shares the experience and practices regarding privacy protection when complying with PDPO from Hong Kong SAR, China， as well as describe how to help customers meet PDPO compliance requirements in Hong Kong SAR, China.

13 Permissions

If you need to assign different permissions to personnel in your enterprise to access your DDS resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you to securely access your Huawei Cloud resources.

With IAM, you can create IAM users and assign permissions to control their access to specific resources. For example, if you want some software developers in your enterprise to use DDS resources but do not want them to delete DDS instances or perform any other high-risk operations, you can create IAM users and grant permission to use DDS instances but not permission to delete them.

If your Huawei account does not require individual IAM users for permissions management, you can skip this section.

IAM is a free service. You only pay for the resources in your account. For more information about IAM, see **IAM Service Overview**.

DDS Permissions

New IAM users do not have any permissions assigned by default. You need to first add them to one or more groups and attach policies or roles to these groups. The users then inherit permissions from the groups and can perform specified operations on cloud services based on the permissions they have been assigned.

DDS is a project-level service deployed for specific regions. When you set **Scope** to **Region-specific projects** and select the specified projects in the specified regions, the users only have permissions for DDS instances in the selected projects. If you set **Scope** to **All resources**, the users have permissions for DDS instances in all region-specific projects. When accessing DDS, the users need to switch to the authorized region.

You can grant users permissions by using roles and policies.

- Roles: A coarse-grained authorization strategy provided by IAM to assign permissions based on users' job responsibilities. Only a limited number of service-level roles are available for authorization. Huawei Cloud services depend on each other. When you grant permissions using roles, you also need to attach any existing role dependencies. Roles are not ideal for fine-grained authorization and least privilege access.

- Policies: A fine-grained authorization strategy that defines permissions required to perform operations on specific cloud resources under certain conditions. This type of authorization is more flexible and is ideal for least privilege access. For example, you can grant users only permission to manage DDS instances of a certain type.

Table 13-1 lists all the system-defined permissions for DDS.

Table 13-1 System-defined permissions for DDS

Role/Policy Name	Description	Type	Dependencies
DDS FullAccess	Full permissions for Document Database Service.	System-defined policy	To create yearly/monthly instances, configure the CBC actions: ● bss:balance:view ● bss:balance:update ● bss:order:view ● bss:order:pay ● bss:order:update ● bss:renewal:view ● bss:renewal:update To unsubscribe from a yearly/monthly instance, you need to configure the following CBC action: ● bss:unsubscribe:update
DDS ReadOnlyAccess	Read-only permissions for Document Database Service. Users with these permissions can only view DDS data.	System-defined policy	None

Table 13-2 lists the common operations supported by system-defined permissions for DDS.

Table 13-2 Common operations supported by system-defined permissions

Operation	DDS FullAccess	DDS ReadOnlyAccess
Creating an instance	√	x
Querying instances	√	√
Deleting an instance	√	x
Restarting an instance	√	x
Performing a primary/ secondary switchover	√	x
Changing a database port	√	x
Resetting a password	√	x
Modifying an SSL policy	√	x
Changing a security group	√	x
Binding or unbinding an EIP	√	x
Scaling up storage space	√	x
Changing instance specifications	√	x
Adding nodes	√	x
Deleting the node that fails to be added	√	x
Modifying an automated backup policy	√	x
Renaming an instance	√	x
Changing a private IP address	√	x
Changing the parameter template associated with the node in a DB instance	√	x
Showing original slow logs	√	x
Enabling or disabling local log auditing	√	x
Downloading audit logs	√	x
Deleting audit logs	√	x

Operation	DDS FullAccess	DDS ReadOnlyAccess
Downloading a backup	√	x
Creating a manual backup	√	x
Querying the backup list	√	√
Restoring data to a new DB instance	√	x
Restoring data to an existing instance	√	x
Deleting a backup	√	x
Creating a parameter template	√	x
Viewing parameter templates	√	√
Modifying a parameter template	√	x
Deleting a parameter template	√	x
Viewing a task center list	√	x
Stopping a backup	√	x

Table 13-3 lists common DDS operations and corresponding actions. You can refer to this table to customize permission policies.

Table 13-3 Common operations and supported actions

Operation	Actions	Authorization Scope	Description
Instance creation page	● vpc:vpcs:list ● vpc:subnets:get ● vpc:securityGroups:get	Supported: ● IAM projects ● Enterprise projects	The VPC, subnet, and security group are displayed on the instance creation page.

Operation	Actions	Authorization Scope	Description
Creating an instance	● dds:instance:create ● vpc:vpcs:list ● vpc:vpcs:get ● vpc:subnets:get ● vpc:securityGroups:get ● vpc:ports:get	Supported: ● IAM projects ● Enterprise projects	If the default VPC, subnet, and security group are used, the vpc:*:create permission must be configured. To create an encrypted instance, configure the KMS Administrator permission for the project.
Querying DB instances	dds:instance:list	Supported: ● IAM projects ● Enterprise projects	-
Querying details of a DB instance	dds:instance:list	Supported: ● IAM projects ● Enterprise projects	If the VPC, subnet, and security group need to be displayed on the instance details page, add the vpc:*:get and vpc:*:list actions.
Exporting DB instance lists	dds:instance:list	Supported: ● IAM projects ● Enterprise projects	If the VPC, subnet, and security group are required, add the vpc:*:get and vpc:*:list actions.
Deleting a DB instance	dds:instance:deleteInstance	Supported: ● IAM projects ● Enterprise projects	When deleting a DB instance, delete the IP address on the data side.
Restarting a DB instance	dds:instance:reboot	Supported: ● IAM projects ● Enterprise projects	-
Performing a primary/secondary switchover	dds:instance:switchover	Supported: ● IAM projects ● Enterprise projects	-
Changing a port	dds:instance:modifyPort	Supported: ● IAM projects ● Enterprise projects	-

Operation	Actions	Authorization Scope	Description
Resetting a password	dds:instance:resetPasswd	Supported: - IAM projects - Enterprise projects	-
Modified SSL	dds:instance:modifySSL	Supported: - IAM projects - Enterprise projects	-
Modifying a security group	dds:instance:modifySecurityGroup	Supported: - IAM projects - Enterprise projects	-
Binding an EIP	dds:instance:bindPublicIp	Supported: - IAM projects	When binding an EIP, you need to query the created EIP. - Enterprise projects are not supported. - Fine-grained authentication is not supported. For details, see **Floating IP Address**.
Unbinding an EIP	dds:instance:unbindPublicIp	Supported: - IAM projects	- Enterprise projects are not supported. - Fine-grained authentication is not supported. For details, see **Floating IP Address**.
Scaling up storage space	dds:instance:extendVolume	Supported: - IAM projects - Enterprise projects	-
Changing DB instance classes	dds:instance:modifySpec	Supported: - IAM projects - Enterprise projects	-

Operation	Actions	Authorization Scope	Description
Adding nodes	● dds:instance:extendNode ● vpc:vpcs:list ● vpc:vpcs:get ● vpc:subnets:get ● vpc:securityGroups:get ● vpc:ports:get	Supported: ● IAM projects ● Enterprise projects	-
Deleting the node that fails to be added	dds:instance:extendNode	Supported: ● IAM projects ● Enterprise projects	If the IP address has been created but the subsequent procedure fails, delete the IP address on the data side.
Modifying an automated backup policy	dds:instance:modifyBackupPolicy	Supported: ● IAM projects ● Enterprise projects	-
Renaming a DB instance	dds:instance:modify	Supported: ● IAM projects ● Enterprise projects	-
Changing a private IP address	● dds:instance:modifyVIP ● vpc:subnets:get ● vpc:ports:get	Supported: ● IAM projects ● Enterprise projects	Before changing the private IP address, query available IP addresses.
Changing the parameter template associated with the node in a DB instance	dds:instance:modifyParameter	Supported: ● IAM projects ● Enterprise projects	-
Showing original slow logs	dds:instance:modifySlowLogPlaintextSwitch	Supported: ● IAM projects ● Enterprise projects	-

Operation	Actions	Authorization Scope	Description
Enabling or disabling local log auditing	dds:instances:modifyAuditLogSwitch	Supported: ● IAM projects ● Enterprise projects	-
Downloading audit logs	dds:instances:downloadAuditLog	Supported: ● IAM projects ● Enterprise projects	-
Deleting audit logs	dds:instance:deleteAuditLog	Supported: ● IAM projects ● Enterprise projects	-
Downloading a backup file	dds:backup:download	Supported: ● IAM projects ● Enterprise projects	-
Changing the billing mode from pay-per-use to yearly/monthly	dds:instances:renew	Supported: ● IAM projects ● Enterprise projects	-
Creating a Manual Backup	dds:instance:createManualBackup	Supported: ● IAM projects ● Enterprise projects	-
Querying the backup list	dds:backup:list	Supported: ● IAM projects ● Enterprise projects	-

Operation	Actions	Authorization Scope	Description
Restoring data to a new DB instance	● dds:backup:createInstanceFromBackup ● vpc:vpcs:list ● vpc:vpcs:get ● vpc:subnets:get ● vpc:securityGroups:get ● vpc:ports:get	Supported: ● IAM projects ● Enterprise projects	The KMS Administrator permission needs to be configured for the encrypted instance in the project.
Restoring to an existing DB instance	dds:backup:refreshInstanceFromBackup	Supported: ● IAM projects ● Enterprise projects	-
Deleting a backup	dds:backup:delete	Supported: ● IAM projects ● Enterprise projects	-
Creating a parameter template	dds:param:create	Supported: ● IAM projects ● Enterprise projects	-
Viewing parameter templates	dds:param:list	Supported: ● IAM projects ● Enterprise projects	-
Modifying a parameter template	dds:param:modify	Supported: ● IAM projects ● Enterprise projects	-
Deleting a parameter template	dds:param:delete	Supported: ● IAM projects ● Enterprise projects	-
Task Center list	dds:task:list	Supported: ● IAM projects ● Enterprise projects	-

Operation	Actions	Authorization Scope	Description
Stopping a backup	dds:backup:stop	Supported: ● IAM projects ● Enterprise projects	-
Querying a log group	lts:groups:get	Supported: ● IAM projects ● Enterprise projects	-
Querying a log stream	lts:topics:get	Supported: ● IAM projects ● Enterprise projects	-

Helpful Links

- **What Is IAM?**
- **Creating a User Group, a User, and Granting DDS Permissions**
- **Permissions Policies and Supported Actions**

14 Related Services

The following figure shows the relationship between DDS and other services.

Figure 14-1 Related services

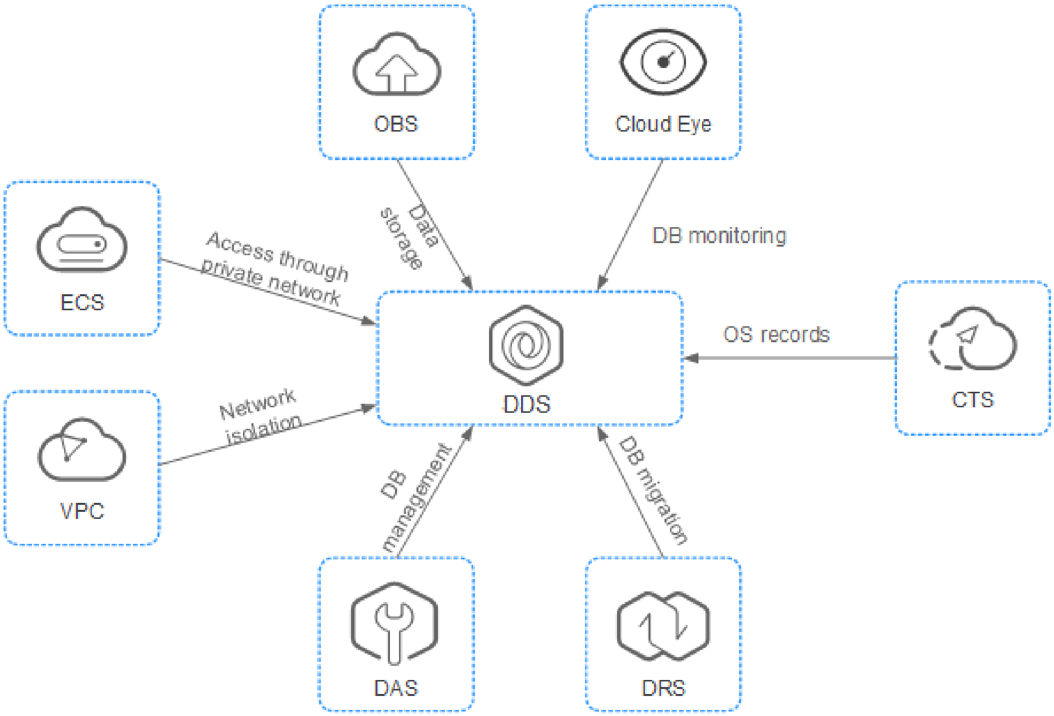


Table 14-1 Related services

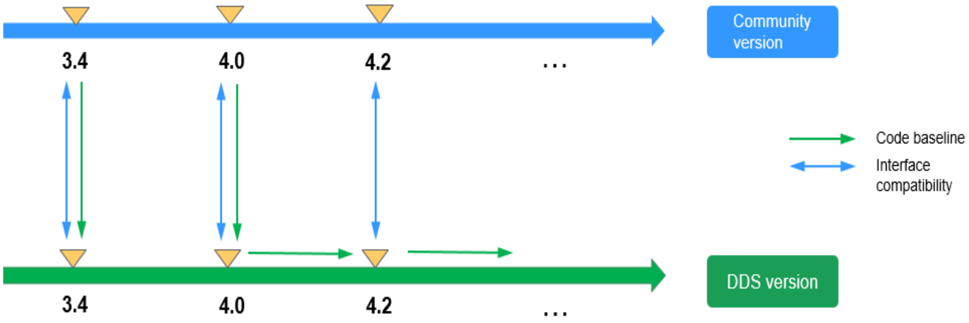
Service Name	Function
Elastic Cloud Service (ECS)	ECS provides DDS with elastic computing resources and a running environment for DB instances.
Virtual Private Cloud (VPC)	VPC provides DDS with elastic network resources and implements network isolation and access control for your DB instances.

Service Name	Function
Object Storage Service (OBS)	OBS stores your DDS instance backup files.
Cloud Eye	Cloud Eye serves as a monitoring platform, monitoring DDS resources for you in real time. It reports alarms and issues warnings promptly to ensure that services are running properly.
Cloud Trace Service (CTS)	CTS records operations related to DDS, facilitating your further queries, audits, and retrievals.
Data Replication Service (DRS)	DRS smoothly migrates databases to the cloud. Source databases remain operational during migration, minimizing downtime and impact.

15 Mapping Between DDS Versions and Community Versions

Document Database Service (DDS) is a cloud database service compatible with MongoDB. DDS major versions select community versions that have not reached EOL and have new major functions as candidate versions for compatibility. The DDS version does not correspond to the community version. To help you understand the mapping between DDS versions and community versions, DDS uses compatible community versions. For details, see **Figure 15-1**.

Figure 15-1 Mapping between DDS versions and community versions



- DDS 3.4 and 4.0 are developed based on the corresponding community versions. The implementation of the same interface is consistent with that in the community. Compared with a community version, DDS has higher security and more O&M functions. DDS can better meet commercial application requirements.
- After October 16, 2018, DDS 4.0 uses the community version 4.0.3 as the baseline version. New features are independently developed and evolved.
- DDS 4.2 and later versions use the community version 4.0.3 as the baseline version. New features are independently developed and evolved. The storage engine is switched to RocksDB to provide better user experience.
- DDS source code has been opened in the GitHub community. For details, see **https://github.com/hwCloudDBSDDS/dds**.