阿里云
云数据库RDS

产品简介

文档版本：20180911

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止：** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告：** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明：** 您也可以通过按**Ctrl** + **A**选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 隐藏目录

1.1 临时文档

- MySQL实例数据同步模式须知
  - MySQL 5.5主备之间的数据同步采用半同步模式。在该模式下写性能会有所下降，但优点是可以极大降低出现主备数据不一致的概率。如果您对数据可靠性要求非常高（例如金融系统），建议您购买MySQL 5.5及以上版本的实例。
  - MySQL 5.6主备之间的数据同步采用MySQL 5.6的新特性GTID，该特性既能保证性能，又能保证主备数据的一致性。

说明：

因为多可用区之间存在一定的网络延迟，因此多可用区RDS实例在采用半同步数据复制方案时，对于单个更新的响应时间会比单可用区实例长。这种情况最好通过提高并发量的方式来实现整体吞吐量的提高。

高可用策略

高可用策略是根据用户自身业务的特点，采用服务优先级和数据复制方式之间的不同组合，以组合出适合自身业务特点的高可用策略。

服务优先级有以下两个级别：

- RTO（Recovery Time Objective）优先：数据库应该尽快恢复服务，即可用时间最长。对于数据库在线时间要求比较高的用户建议使用 RTO 优先策略。
- RPO（Recovery Point Objective）优先：数据库应该尽可能保障数据的可靠性，即数据丢失量最少。对于数据一致性要求比较高的用户建议使用 RPO 优先策略。

数据复制方式有以下三种方式：

- 异步复制（Async）：应用发起更新（含增加、删除、修改操作）请求，Master 完成相应操作后立即响应应用，Master 向 Slave 异步复制数据。因此异步复制方式下，Slave 不可用不影响主库上的操作，而 Master 不可用有较小概率会引起数据不一致。
- 强同步复制（Sync）：应用发起更新（含增加、删除、修改操作）请求，Master 完成操作后向 Slave 复制数据，Slave 接收到数据后向 Master 返回成功信息，Master 接到 Slave 的反馈后再

响应应用。Master 向 Slave 复制数据是同步进行的，因此 Slave 不可用会影响 Master 上的操作，而 Master 不可用不会引起数据不一致。

- 半同步复制（Semi-Sync）：正常情况下数据复制方式采用强同步复制方式，当 Master 向 Slave 复制数据出现异常的时候（Slave 不可用或者双节点间的网络异常），Master 会暂停对应用的响应，直到复制方式超时退化成异步复制。如果允许应用在此时更新数据，则 Master 不可用会引起数据不一致。当双节点间的数据复制恢复正常（Slave 恢复或者网络恢复），异步复制会恢复成强同步复制。恢复成强同步复制的时间取决于半同步复制的实现方式，阿里云数据库 MySQL 5.5 版和 MySQL 5.6 版有所不同。

用户可以根据自身业务特点，选择服务优先级和数据复制方式的不同组合方式，提高可用性。

云数据引擎	服务优先级	数据复制方式	组合特点
MySQL 5.1	RPO	Async	在 Master 发生故障的情况下，切换会发生在 Slave 应用完所有的 Relay Log 之后。在 Slave 发生故障的情况下，应用操作 Master 不受影响。在 Slave 恢复之后再同步 Master 上面的数据。
MySQL 5.5	RPO	Async	在 Master 发生故障的情况下，切换会发生在 Slave 应用完所有的 Relay Log 之后。在 Slave 发生故障的情况下，应用操作 Master 不受影响。在 Slave 恢复之后再同步 Master 上面的数据。
MySQL 5.5	RTO	Semi-Sync	在 Master 发生故障且数据复制未退化的情况下，因为数据一致性已经得到保障，RDS 将立即触发切换操作把流量导向 Slave。在 Slave 发生故障的情况下，应用操作 Master 将会出现超时，而后数据复制方式退化为异步复制方式；在 Slave 恢复并同步完 Master 上的数据之后，数据复制方式恢复为强同步。在双节点数据不一致且数据复制方式已经退化为异步复制方式的情况下，如果 Master 发生了故障，则切换会发生在 Slave 应用完所有的 Relay Log 之后。
MySQL 5.6	RPO	ASync	在 Master 发生故障的情况下，切换会发生在 Slave 应用完所有的 Relay Log 之后。在 Slave 发生故障的情况下，应用操作 Master 不受影响。在 Slave 恢复之后再同步 Master 上面的数据。
MySQL 5.6	RTO	Semi-Sync	在 Master 发生故障且数据复制未退化的情况下，因为数据一致性已经得到保障，RDS 将立即触发切换操作把流量导向 Slave。在 Slave 发生故障的情况下，应用操作 Master 将会出现超时，而后数据复制方式退化为异步复制方式；在

云数据引擎	服务优先级	数据复制方式	组合特点
			Slave 恢复并同步完 Master 上的数据之后，数据复制方式恢复为强同步。在双节点数据不一致且数据复制方式已经退化为异步复制方式的情况下，如果 Master 发生了故障，则切换会发生在 Slave 应用完所有的 Relay Log之后。
MySQL 5.6	RPO	Semi-Sync	在 Master 发生故障且数据复制未退化的情况下，因为数据一致性已经得到保障，RDS 将立即触发切换操作把流量导向 Slave。在 Slave 发生故障的情况下，应用操作 Master 将会出现超时，而后数据复制方式退化为异步复制方式；在 Slave 重新获取到 Master 信息时（Slave 恢复或者网络故障恢复），数据复制方式恢复为强同步方式。在双节点数据不一致且 Slave 上的数据差异无法补全的情况下，如果 Master 发生了故障，则用户可以通过 API 获取 Slave 的时间点并决定何时切换以及补全数据的方法。
MySQL 5.7	X	X	目前不支持调整
SQL Server 2008 R2	X	X	目前不支持调整
SQL Server 2012	X	X	目前不支持调整
PostgreSQL	X	X	目前不支持调整
PPAS	X	X	目前不支持调整

High-availability policy

The high-availability policies use a combination of service priorities and data replication modes to meet the business needs.

The service priorities are as follows:

- RTO (Recovery Time Objective) priority: The database must restore services as soon as possible within a specified time frame. This is best for users who require their databases to provide uninterrupted online service.

- RPO (Recovery Point Objective) priority: The database must guarantee the data reliability, that is, as little data loss as possible. This is best for users whose highest priority is data consistenc y.

There are three data replication methods:

- Asynchronous replication (Async):

  In this mode, the master node does not immediately synchronize data to the slave node. When an application initiates an update request, which may include add, delete, or modify operations , the master node responds to the application immediately after completing the operation but does not necessarily replicate that data to the slave node right away. This means that the operation of the primary database is not affected if the slave node is unavailable, but data inconsistencies may occur if the master node is unavailable.
- Forced synchronous replication (Sync):

  When an application initiates an update (add, delete, or modify) request, the primary database completes the operation and then replicates data to the standby database. After the standby database receives the data, it returns a success message to the primary database. The primary database waits for a feedback from the standby database before responding to the application . This means that the operation of the master node is affected if the slave node is unavailable, but the data on the master and slave nodes is always consistent.
- Semi-synchronous replication (Semi-Sync):

  This functions as a hybrid of the two preceeding replication modes. In this mode, when both nodes are functioning normally, data replication is identical to the forced synchronous replicatio n mode. However, when there is an exception, such as the slave node becoming unavailable or a network exception occurring between the two nodes, the master node only attempts to replicate data to the slave node and suspend its response to the application for a set period of time. Once the replication mode has timed out, the master node degrades to asynchronous replication. At this point, if the master node becomes unavailable and the application updates its data from the slave node, it is consistent with the data on the master node. When data replication between the two nodes returns to normal, because the slave node or network connection is recovered, forced synchronous replication is reinstated. The amount of time it takes for the nodes to return to forced synchronous replication depends on how the semi-synchronous replication mode was implemented. For instance, ApsaraDB for MySQL 5.5 is different from ApsaraDB for MySQL 5.6 in this regard.

Several combinations of service priorities and data replication modes are available to meet your database and business needs. The characteristics of key combinations are detailed in the following table.

Cloud data engine	Service priority	Data replication mode	Combination characteristics
MySQL 5.1	RPO	Async	• If the master node fails, the slave node switches over after applying all of the relay logs. • If the slave node fails, application operations on the master node are not affected. The data on the master node is synchronized after the slave node recovers.
MySQL 5.5	RPO	Async	• If the master node fails, the slave node switches over after applying all of the relay logs. • If the slave node fails, application operations on the master node are not affected. The data on the master node is synchronized after the slave node recovers.
MySQL 5.5	RTO	Semi-Sync	• If the master node fails and data replication degrades, RDS immediately triggers the

Cloud data engine	Service priority	Data replication mode	Combination characteristics
			switchover and direct traffic to the slave node because data consistency is guaranteed. • If the slave node fails, application operations on the master node times out, and data replication degrades to asynchronous replication. After the slave node recovers and the data on the master node is synchronized completely, data replication returns to forced synchronization. • If the master node fails while the two nodes have inconsistent data and the data replication mode degrads to asynchronous replication, the slave node switches over after applying all of the relay logs.
MySQL 5.6	RPO	Async	• If the master node fails, the slave node switches over after

Cloud data engine	Service priority	Data replication mode	Combination characteristics
			applying all of the relay logs. • If the slave node fails, application operations on the master node are not affected. The data on the master node is synchronized after the slave node recovers.
MySQL 5.6	RTO	Semi-Sync	• If the master node fails and data replication degrads, RDS immediately triggers the switchover and direct traffic to the slave node because data consistency is guaranteed. • If the slave node fails, application operations on the master node times out, and data replication degrades to asynchronous replication. After the slave node recovers and the data on the master node is synchronized completely, data replication returns to forced synchronization.

Cloud data engine	Service priority	Data replication mode	Combination characteristics
			• If the master node fails while the two nodes have inconsistent data and the data replication mode degrads to asynchronous replication, the slave node switches over after applying all of the relay logs.
MySQL 5.6	RPO	Semi-Sync	• If the master node fails and data replication has not degraded, RDS immediately triggers the switchover and direct traffic to the slave node because data consistency has been guaranteed. • If the slave node fails, application operations on the master node times out, and data replication degrades to asynchronous replication. When the slave node can obtain information from the master node again, because the slave node or

Cloud data engine	Service priority	Data replication mode	Combination characteristics
			network connection recovers, data replication returns to forced synchronization. • If the master node fails while the two nodes have inconsistent data and the data difference on the slave node cannot be reconciled completely, you can obtain the time of the slave node through the API. Then you can decide when to switchover and which method you plan to use to reconcile the data.
MySQL 5.7	X	X	Currently this engine does not support policy adjustments.
SQL Server 2008 R2	X	X	Currently this engine does not support policy adjustments.
SQL Server 2012	X	X	Currently this engine does not support policy adjustments.
PostgreSQL	X	X	Currently this engine does not support policy adjustments.
PPAS	X	X	Currently this engine does not support policy adjustments.

1.2 Migration service

The migration service can migrate data from a local database to ApsaraDB, or migrate an ApsaraDB instance to another. ApsaraDB includes a Data Transfer Service (DTS) tool to facilitate quick database migration.

DTS is a cloud data transfer service for efficient instance migration from local databases to the cloud, and between RDS instances. For more information, see *DTS product overview*.

DTS provides three migration modes, that is, structure migration, full migration, and incremental migration:

- Structure migration: DTS migrates the structure definitions of migration objects to the target instance. Currently, tables, views, triggers, stored procedures, and stored functions can be migrated in this mode.
- Full migration: DTS migrates all existing data of migration objects in source databases to the target instance.
- Incremental migration: DTS synchronizes data changes made in the migration process to the target instance.

1.3 Dedicated instances

Product introduction

As a new instance type of ApsaraDB for RDS, dedicated instances feature fixed computing capabilities, storage space, and IO performance. It differs from other instance types in terms of resource allocation strategies. With more stable performance, a dedicated instance is the best option for business scenarios where a database-centric system is used, such as finance, e-commerce, government, medium- and large-sized Internet businesses.

Dedicated instances have multiple specification types. For more information, see *Instance type list*. Dedicated-host instances have the top-level configurations among dedicated instances. You can change the instance configurations as needed. Configuration changes are free of restrictions, which means configuration changes are allowed across different instance types.

Design principles and performance features

- With the help of the OS (Linux/Windows) kernel, RDS isolates computing resources of instances located on different physical servers. A dedicated instance uses a slightly different CPU allocation strategy from a common instance. A number of fully dedicated CPU cores and

threads are assigned to the dedicated instance to ensure long-term stability and predictability in computing performance. This avoids noisy neighborhood on the physical server.

- A dedicated instance has a reserved storage space. As compared with a common instance, the dedicated instance can fully avoid instance migration across physical machines caused by disk capacity increase of your instance or other instances to provide higher stability. Furthermore, the dedicated instance supports hot standby, so you can fail over at any time in case of disk failure on one instance, guaranteeing the availability of the instance. After the failover, you can change the host in host replacement mode. The process is transparent and does not affect your services.

Advantages

It is not likely to make direct comparison between dedicated and common instances, because their metrics do not fully match. However, to help make purchase decisions, we select two similar specifications for a cost-effectiveness analysis.

Type	Type ID	CPU/ Memory	Disk space	Maximum number of connections	Maximum IOPS	Monthly price (effective before January 2017)
Common instances	rds.mysql.m1.medium	4-core 16 GB	500 GB	4000	7000	2,100 RMB
Dedicated instances	mysql.x8.large. 2	4-core 32 GB	500 GB	5000	9000	3,650 RMB

The above table shows that a dedicated instance costs 70% more than a common instance, but it offers twice the memory capacity as the common instance, 25% more in the maximum number of connections, and 28% more in the maximum IOPS. In addition, it has stable CPU computing performance. Therefore, the dedicated instance provides higher overall cost-effectiveness. For more information about the prices of dedicated instances, see *Pricing*.

1.4 Standard and certification

For standards and certification of Alibaba Cloud, refer to *Trust Center*.

2 What is RDS

ApsaraDB for RDS (Relational Database Service) is a stable, reliable, and auto-scaling online database service. Based on the Alibaba Cloud's distributed file system and high-performance SSD storage, RDS supports MySQL, SQL Server, PostgreSQL, and PPAS (highly compatible with Oracle) engines. It offers a complete set of solutions for backup, recovery, monitoring, migration, and disaster recovery to free you from worries about database O&M.

Learning Path

RDS Learning Path will walk you through the concepts and operations of RDS.

RDS for MySQL

MySQL is the world's most popular open source database. As an important part of LAMP and a combination of open source software (Linux + Apache + MySQL + Perl/PHP/Python), MySQL is widely used in a variety of applications.

In the Web 2.0 era, MySQL serves as the basis of the underlying architecture of the popular BBS software system Discuz! and blogging platform WordPress. In the Web 3.0 era, leading Internet companies including Alibaba, Facebook, and Google have all taken advantage of the flexibility of MySQL to build their large-scale mature database clusters.

Based on Alibaba's MySQL source code branch, RDS for MySQL proves to have excellent performance and throughput. It withstands the massive data traffic and large number of concurrent users during many November 11 (Singles' Day) shopping festivals - the Chinese equivalent of Cyber Monday. RDS for MySQL also offers a range of advanced functions including optimized read/write splitting, data compression, and intelligent optimization.

RDS for MySQL currently supports versions 5.5, 5.6, and 5.7.

RDS for SQL Server

SQL Server is one of the first commercial databases and is an important part of the Windows platform (IIS + .NET + SQL Server), with support for a wide range of enterprise applications. The SQL Server Management Studio software comes with a rich set of built-in graphical tools and script editors. You can quickly get started with a variety of database operations through visual interfaces.

Powered by high-availability architecture and the ability to recover to any point in time, RDS for SQL Server provides strong support for a variety of enterprise applications. It also covers Microsoft’s licensing fee.

RDS for SQL Server currently supports the following versions:

- SQL Server 2008 R2 Enterprise
- SQL Server 2012 Web, Standard, and Enterprise
- SQL Server 2016 Web, Standard, and Enterprise

RDS for PostgreSQL

PostgreSQL is the world's most advanced open source database. As the forerunner among academic relational database management systems, PostgreSQL excels for its full compliance with SQL specifications and robust support for a diverse range of data formats such as JSON, IP, and geometric data, which are not supported by most commercial databases.

In addition to excellent support for features such as transactions, subqueries, Multi-Version Concurrency Control (MVCC), and data integrity check, RDS for PostgreSQL integrates a series of important functions including high availability, backup, and recovery that help ease your O&M burden.

RDS for PostgreSQL currently supports version 9.4.

RDS for PPAS

Postgres Plus Advanced Server (PPAS) is a stable, secure, and scalable enterprise-level relational database. Based on PostgreSQL, PPAS brings enhancements in terms of performance , application solutions, and compatibility. It also provides the capability to run Oracle applications directly. It is a reliable and cost-effective option for running a variety of enterprise applications.

RDS for PPAS provides account management, resource monitoring, backup, recovery, and security control, and more functions, and is continuously updated and improved.

RDS for PPAS currently supports version 9.3.

3 Benefits

3.1 Inexpensive and easy-to-use

Simple deployment

You can customize RDS specifications through Alibaba Cloud's official website or APIs. After the order is confirmed, RDS generates the specified instance instantly. RDS can work with ECS to reduce the application response time and public traffic fees.

On-demand upgrades

Initially, you can purchase an RDS instance that meets the existing business requirements. When requirements on the database and data storage capacity change, you can flexibly adjust the instance specifications without any interruptions to the service.

Effortless migration

The use method of RDS is similar to that of the native database engine. You can get started easily without further learning. In addition, RDS is compatible with your current programs and tools. Data can be migrated to RDS using a data import and export tool with minimal labor required.

Ease of management

Alibaba Cloud ensures the normal operation of RDS through routine maintenance and management, such as hardware/software fault processing and database patch upgrades. You can independently perform database addition, deletion, restart, backup, recovery, and other management operations through the Alibaba Cloud console.

Related Topics

- *High performance*
- *High Security*
- *High reliability*
- *Comparison between ApsaraDB for RDS and local databases*

3.2 High performance

Parameter optimization

Alibaba Cloud has accumulated years of experience in production and optimization by gathering key opinions from top database experts in China and aggregating performance data of all the RDS

instances. DBA continuously manages RDS over its life cycle to ensure that RDS is running at the optimal performance.

SQL optimization

Based on your application scenario, RDS locks low-efficiency SQL statements and offers recommendations for optimizing your business code.

High-end backend hardware

All servers used by RDS have been evaluated by multiple parties to ensure the exceptional performance and stability.

Extended reading

- *Instance type list*
- *Performance White Paper (RDS for MySQL)*
- *Performance White Paper (RDS for SQL Server)*
- *Precautions on comparing ECS user-created databases and ApsaraDB for RDS instances in terms of performance*

3.3 High security

DDoS protection

If the RDS instance is configured to be accessible from the Internet, the instance may suffer from DDoS attacks. If a DDoS attack is detected, the RDS security system enables traffic cleaning first. If traffic cleaning fails or the attack reaches the blackhole threshold, blackhole filtering is triggered.

Note:
We recommend that RDS instances are accessed over the intranet to avoid DDoS attacks.

Access control policy

- You can define the IP addresses that are allowed to access RDS. Other IP addresses cannot access RDS.
- Each account can only view and operate its own databases.

System security

- ApsaraDB for RDS is protected by multiple firewall layers that can effectively block a variety of malicious attacks and ensure data security.

- Direct logon to the ApsaraDB for RDS server is not allowed. Only the ports required by the specific database services are open.
- The ApsaraDB for RDS server cannot initiate an external connection. It can only accept access requests.

For more information, see *Network isolation*.

Professional security team

Alibaba Cloud's security team provide rapid security technology support for RDS.

Related Topics

- *Cheap and ease-to-use*
- *High performance*
- *High reliability*
- *Comparison between ApsaraDB for RDS and local databases*

3.4 High reliability

Hot standby

RDS adopts a hot standby architecture. If the master server fails, services will be failed over in seconds. The entire failover process is transparent to applications.

Multi-copy redundancy

The data on the ApsaraDB for RDS server is stored on RAID, and backed up on OSS.

Data backup

ApsaraDB for RDS provides an automatic backup mechanism. You can set a backup schedule or initiate a temporary backup at any time. For more information, see *Backup recovery*.

Data recovery

You can recover data by selecting a backup set or point in time. Generally, you can recover data from any point in time of the past 7 days to a clone or temporary instance. After the data is verified, you can migrate the data back to the master RDS instance. For more information, see *Backup recovery*.

3.5 Comparison between ApsaraDB for RDS and local databases

Performance comparison

Item	ApsaraDB for RDS	Local databases
Service availability	99.95%	You need to manually build master-slave replication and RAID.
Data reliability	99.9999%	You need to manually build master-slave replication and RAID.
System security	DDoS protection, traffic cleaning, timely repair of various database security vulnerabilities	You need to deploy your database at high costs and repair database security vulnerabilities yourself.
Database backup	Automatic backup	Independent backups are possible, but you have to find storage space for backup and regularly validate whether the backup data can be recovered.
Software and hardware investment	Pay as you go, with no software and hardware investment	Database servers are rather costly and the licensing fees for SQL Server have to be paid.
System hosting	No hosting fee	A single 2U server costs more than RMB 5,000 RMB/year (more than 10,000 RMB/year for master and slave servers, if needed).
Maintenance cost	Maintenance not needed	Professional DBAs have to be hired, resulting in high labor cost.
Deployment and resizing	Instant activation, fast deployment, automatic resizing	Hardware procurement, data center hosting, and host deployment, and other tasks are time-consuming.
Resource utilization rate	Charged based on actual usage, resulting in 100% utilization.	Resource utilization is low because there are peak and off-peak hours.

Price comparison

Item	ApsaraDB for RDS	Local databases
Costs of hardware, spare	Take the following instance type as an example: An instance with the memory of 1,200 MB and storage	• A minimum of 2 servers are required for a database cluster,

Item	ApsaraDB for RDS	Local databases
parts, and accessories	space of 50 GB (IOPS is up to 600) costs 2,040 RMB/year.	and a single server whose IOPS is up to 600 costs approximately 6,000 RMB. • An intranet switch is used to connect the front-end Web server (an inexpensive 1U non-NMS switch costs about 1,000 RMB). Later hardware damages and replacements take at least 30% of the cost. • Hardware cost: (6,000 x 2 + 1,000) x 130% = 16,900 RMB Annual hardware cost: 16,900 RMB/3= 5,633 RMB (the cost of hardware is calculated over a 3-year depreciation period)
Data center hosting fee	It is service provider’s responsibilities , and no hosting fee is required.	The hosting fee for a 1U cabinet is 3,000 RMB/year, and hosting fees for two 1U servers and a 1U intranet switch are charged. Data center hosting fees: 3,000 x 3 = 9,000 RMB
Bandwidth fee	Communication between ECS and ApsaraDB for RDS in the same region is available through the intranet with no cost. However, communication between ECS and ApsaraDB for RDS across different regions is available over the Internet with a certain cost for Internet traffic. For more information, see *Pricing*.	Available only in the intranet with no cost for Internet traffic.
Cost of database maintenance engineers	There is no labor cost because database maintenance is taken care of by a service provider.	The monthly salary for an entry-level DBA engineer is at least 5,000 RMB. If 30% workload for a full-time engineer is needed for the current project: Labor cost: 5,000 x 12 x 30% = 18,000 RMB
Total annual cost	2,040 RMB/year	32,633 RMB/year

4 System architecture

4.1 Data link service

Alibaba Cloud ApsaraDB provides all of the data link services, including DNS, Server Load Balancer (SLB), and Proxy. Since RDS uses native DB engines, and database operations are highly similar across engines, there is essentially no learning cost for users who are familiar with the engines.

DNS

The DNS module supports the dynamic resolution of domain names to IP addresses, to prevent IP address changes from affecting the performance of your RDS instance. After its domain name is configured in the connection pool, an ApsaraDB instance continues to be accessed even if its IP address changes.

For example, the domain name of an ApsaraDB instance is test.rds.aliyun.com, and the IP address corresponding to this domain name is 10.10.10.1. If either test.rds.aliyun.com or 10.10.10.1 is configured in the connection pool of a program, the instance can be accessed.

After performing a zone migration or version upgrade for this ApsaraDB instance, the IP address may change to 10.10.10.2. If the domain name configured in the connection pool is test.rds.aliyun.com, the instance can still be accessed. However, if the IP address configured in the connection pool is 10.10.10.1, the instance is no longer accessible.

SLB

The SLB module provides instance IP addresses (including both intranet and Internet IP addresses) to prevent physical server changes from affecting the performance of your RDS instance.

For example, the intranet IP address of an RDS instance is 10.1.1.1, and the corresponding Proxy or DB Engine runs on 192.168.0.1. Normally, the SLB module redirects all traffic destined for 10.1.1.1 to 192.168.0.1. If 192.168.0.1 fails, another server in hot standby status with the IP address of 192.168.0.2 takes over for 192.168.0.1. In this case, the SLB module redirects all traffic destined for 10.1.1.1 to 192.168.0.2, and the RDS instance continues to offer its services normally.

Proxy

The Proxy module provides a number of functions including data routing, traffic detection, and session holding.

- Data routing: This supports distributed complex query aggregation for big data and provides the corresponding capacity management.
- Traffic detection: This reduces SQL injection risks and supports SQL log backtracking when necessary.
- Session holding: This prevents database connection interruptions if any failure occurs.

4.2 High-availability service

The high-availability service consists of several modules including the Detection, Repair, and Notification modules. In combination, these modules guarantee the availability of the data link service and process any internal database exception.

In addition, RDS can improve the performance of its high-availability service by migrating instances to a region that supports multi-zone instances and by adopting the appropriate high-availability policies.

Detection

The Detection module checks whether the master and slave nodes of the DB engine are providing their services normally. The HA node uses heartbeat information, acquired at an interval of 8 to 10 seconds, to determine the health status of the master node. This information, combined with the health status of the slave node and heartbeat information from other HA nodes, allows the Detection module to eliminate any risk of judgements caused by exceptions such as network jitter. As a result, a failover can be completed within 30 seconds.

Repair

The Repair module maintains the replication relationship between the master and slave nodes of the DB engine. It can also repair any errors that may occur at either node.

For example:

- Automatic restoration of master/slave replication in case of an unexpected disconnection
- Automatic repair of table-level damage to either node
- On-site saving and automatic repair in case of crashes

Notice

The Notice module informs the SLB or Proxy of status changes to the master and slave nodes to guarantee that you always access the correct node.

For example, the Detection module discovers that the master node has an exception and instructs the Repair module to fix it. If the Repair module fails to resolve the problem, the Notice module will be informed of this information. The Notice module then forwards the failover request to the SLB or Proxy module, which begins to redirect all traffic to the slave node. At the same time, the Repair module creates a new slave node on another physical server and synchronizes this change back to the Detection module. The Detection module starts to recheck the health status of the instance.

Multi-zone

A multi-zone is a physical area that is formed by combining multiple individual zones within the same region. Multi-zone RDS instances can withstand higher level disasters than single-zone ones. For example, a single-zone RDS instance can withstand server and rack failures, while a multi-zone RDS instance can withstand the failure of an entire equipment room.

There is currently no extra charge for multi-zone RDS instances. Users in a region where multi-zone has been enabled can buy multi-zone RDS instances directly or convert single-zone RDS instances into multi-zone ones by using inter-zone migration.

Note:
Certain network latency may exist between multiple zones. As a result, when a multi-zone RDS instance uses a semi-synchronous data replication solution, its response time to any individual update may be longer than that of a single-zone instance. In this case, the best way to improve overall throughput is to increase concurrency.

4.3 Backup and recovery service

This service supports the offline backup, dumping, and recovery of data.

Backup

The Backup module compresses and uploads the data and logs on both the master and slave nodes.

RDS uploads this data to OSS by default, but the backup files can also be dumped to a cheaper and more persistent Archive Storage. Normally, backup is initiated on the slave node so as not to affect the performance of the master node. However, if the slave node is unavailable or damaged, the Backup module initiates backup on the master node.

Recovery

The Recovery module recovers backup files stored on OSS to the target node.

- Master node rollback: This can be used to restore a node to the state that it was in at a specific point in time.
- Slave node repair: This can be used to automatically create a new slave node to reduce risks when an irreparable failure occurs to the slave node.
- Read-only instance creation: This creates a read-only instance from a backup.

Storage

The Storage module uploads, dumps, and downloads backup files.

Currently, all backup data is uploaded to OSS for storage, and users can obtain temporary links to download data as needed. The Storage module also supports dumping backup files from OSS to an Archive Storage for cheaper and steady offline storage.

4.4 Monitoring service

ApsaraDB provides multilevel monitoring services across the physical, network, and application layers to ensure business availability.

Service

The Service module tracks service-level statuses. It monitors whether the SLB, OSS, Archive Storage, Log Service, and other cloud products on which RDS depends are normal, including their functionality and response time. It also uses logs to determine whether the internal RDS services are operating normally.

Network

The Network module tracks statuses at the network layer. It monitors the connectivity between ECS and RDS and between physical RDS servers, as well as the rates of packet loss on the router and switch.

OS

The OS module tracks statuses at the hardware and OS kernel layer, including:

- Hardware overhaul: Constantly checks the operational status of the CPU, memory, main board, and storage, pre-judges whether a fault will occur, and automatically submits a repair report in advance.
- OS kernel monitoring: It tracks all database calls and uses kernel status to analyze the reasons for slowdowns or call errors.

Instance

The Instance module collects RDS instance-level information, including:

- Available instance information
- Instance capacity and performance indicators
- Instance SQL execution records

4.5 Scheduling service

The scheduling service consists of the Resource and Version modules. It mainly allocates resources and manages instance versions.

Resource

The Resource module allocates and integrates the underlying RDS resources. For example, when you create an instance through the *RDS console* or API, the Resource module determines which physical server is best suited to carry traffic. This module also allocates and integrates the underlying resources required for the inter-zone migration of RDS instances. After lengthy instance creation, deletion, and migration operations, the Resource module calculates the degree of resource fragmentation in a zone and initiates resource integration regularly to improve the service carrying capacity of the zone.

Version

The Version module is applicable to version upgrades of RDS instances. For example:

- Major version upgrade: Upgrade MySQL 5.1 to MySQL 5.5, and upgrade MySQL 5.5 to MySQL 5.6.
- Minor version upgrade: Fix a bug in the MySQL source code.

5 Product series

5.1 General introduction to product series

Currently, RDS instances are divided into three series: Basic Edition, High-availability Edition and Finance Edition. Different series support different engine types and instance types. For more information, see *Instance type overview*.

Note:
Currently the Finance Edition is applicable only to regions in China.

Brief introductions

Series	Introduction	Use cases
Basic Edition	It uses the storage-computing-isolated architecture and a single computing node, realizing super high cost effectiveness.	• Personal learning • Micro websites • Development and testing environments for small- and medium-sized enterprises
High-availability Edition	It uses the classic high-availability architecture with one master node and one slave node. The ephemeral SSD storage provides the best and balanced performance.	• Production databases of large-sized enterprises • Applications for Internet, Internet of things (IoT), retail e-commerce sales, logistics, gaming, and other industries

Feature differences

Feature	Basic Edition	High-availability Edition
Engine	• MySQL 5.7 • SQL Server 2012 Web, Enterprise • SQL Server 2016 Web	• MySQL 5.5/5.6/5.7 • SQL Server 2008 R2 Enterprise • SQL Server 2012 Standard, Enterprise • SQL Server 2016 Standard, Enterprise • PostgreSQL 9.4

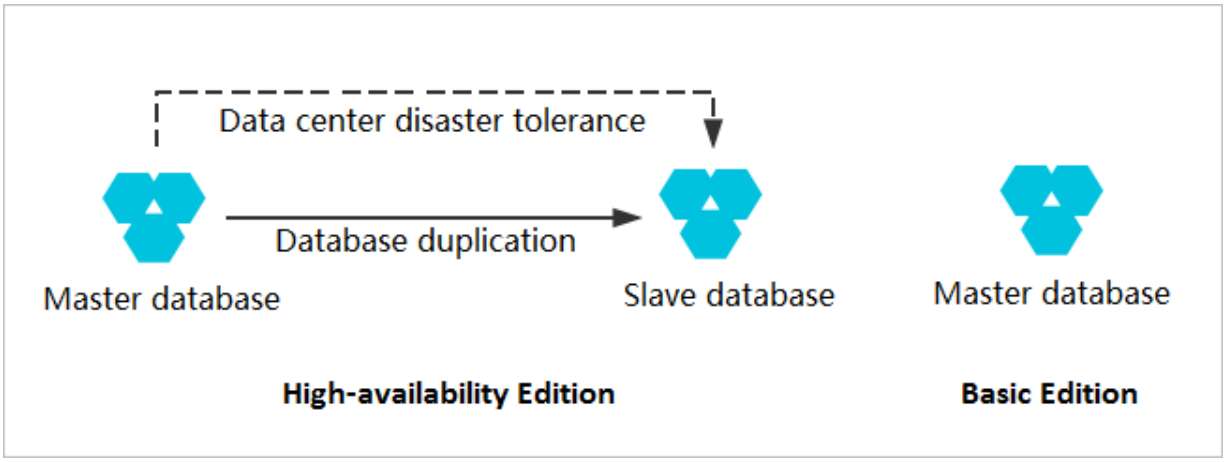
Feature	Basic Edition	High-availability Edition
		• PPAS 9.3
Number of nodes	1	2
Maximum configuration	32-core 128 GB/2 TB	60-core 470 GB/3 TB
Set monitoring and *alarms*	Supported	Supported
IP whitelist	Supported	Supported
Backup and recovery	Supported	Supported
Parameter settings	Supported	Supported
SSL and *TDE*	Not supported	Supported (Currently MySQL 5.7 does not support TDE)
Log management	Not supported	Supported
Performance optimization	Not supported	Supported
Read-only instance (Additional instances required)	Not supported	Supported only by MySQL 5.6/5.7
Read/write splitting	Not supported	Supported only by MySQL 5.6
Built-in read/write splitting	Not supported	Not supported
SQL audit	Not supported	Supported (Additional payment required)
High-frequency monitoring	Not supported	Supported (Additional payment required)

5.2 Basic Edition

General introduction

The Basic Edition is a new series of ApsaraDB for RDS, which is deployed based on a single-node architecture. Compared with the mainstream master/slave High-availability Edition, the Basic Edition contains only one node and has no slave node for fault recovery. Currently, both MySQL and SQL Server support this new series.

The following picture shows the architecture of the Basic Edition and High-availability Edition.



Comparative advantages

The slave database in the High-availability Edition is used only for failover but does not provide services, and the database replication causes extra performance cost to the master database. In this respect, the Basic Edition is not inferior but superior to the High-availability Edition in terms of performance.

The Basic Edition uses the underlying data distributed storage layer to guarantee the stability of multiple replicas, that is, the fault in or damage to one physical node does not result in data loss. Besides, the costs can be greatly saved by reducing one database node. As a result, the price of the Basic Edition is only half of the High-availability Edition.

Note:
Because there is only one database node, when the node fails, it takes longer to recover the node. Therefore, the High-availability Edition is recommended for sensitive businesses that have higher requirements on database availability.

Restrictions

Compared with the High-availability Edition, the Basic Edition does not support the following functions:

- Master/slave failover
- Zone failover
- Safe connection mode
- Log management
- Performance diagnosis
- Read-only instances

- Disaster recovery instances

5.3 国际站没有

Brief Introduction

Financial edition (formerly: Three-Point Enterprise Edition) alibaba Cloud relational database RDS is a fully self-developed series of cloud databases for high-end enterprise-class users. Currently , financial edition only supports MySQL An instance of version 5.6, in addition to maintaining the original MySQL compatibility and availability, we introduced the raft Protocol into the alisql kernel using MySQL semi-sync. Replication enables simultaneous replication of multiple copies of the log to ensure strong consistency of data, provide financial-level reliability.

The topology diagram for the financial edition instance is shown in the following figure.

Only multi-availability zones (Multi-availability zones with three availability zones) are supported) the region only supports financial edition examples, so currently, there are only four regions that support an RDS financial edition instance, that is, East China 1, East China 2, South China 1, North China 2.

The financial edition only provides instance specifications for exclusive/exclusive physical machine types, its largest supported specification is a 60-core 3000 GB instance with a storage space of 470 GB.

Functions

An example of a financial edition has the following features:

- Provides complete product functionality, includes instance lifecycle, elastic scaling, instance management, network link, Backup recovery, performance optimization, functions, triggers, stored procedures, and so on.
- Read-Only instances and read-write separation are supported to meet the application scenarios with a large number of read requests.
- Provide free SQL auditing, keep all SQL Execution records for 30 days, provide access to the database with proof of reference to ensure the security of the core data.
- Supports data link encryption for SSL and Data Store transparent encryption for TDE, target customers with very high requirements for data security, availability, and reliability. In the future , we will also introduce a number of value-added services in conjunction with the actual scene, and guarantee the core data in all directions, help enterprises to develop rapidly.

Restrictions

- No single-available area instances are provided, and the financial edition instances are deployed across the available zone.
- Cross-regional deployment is not available based on performance considerations.

Feature

An example of a financial edition has the following characteristics:

- Reliability

  Three database nodes are maintained at the bottom of the financial edition. In the one-master-two-slave replication topology structure, all nodes contain the complete data, and the database transaction logs are synchronized to all slave databases from the master database. The transaction can only be submitted when more than half of the nodes within the cluster have been written successfully. Although it uses synchronized replication, the fault of an individual node will not affect the availability of the entire instance, as there are three nodes in total. Advantages of this design are obvious. It uses relatively high data redundancy to ensure higher reliability without affecting the availability. It also supports deployment across different data centers, to allow data center disaster tolerance.
- Compatibility

  Financial edition in alisql Based on 100% development, 5.6 compatible with MySQL protocol , any application that used to use the MySQL database can be seamlessly switched to a financial edition instance. Then, the RDS will provide a one-click thermal upgrade from the available version of the dual machine to the financial edition, continue the life cycle of the original Instance, eliminating the data migration, the old instance recovery and so on a series of tricky operation.
- Alisql Research and Development

  At the kernel level, alisql has made a number of functional improvements, such:

  - Distributed consistency protocol (RAFT) is used to ensure the reliability and atomic of multi-node state switching.
  - Roll back unfinished transactions with flashback to ensure data consistency.
  - Take advantage of parallel replication Replication) increases the efficiency of the standby library Application Log and reduces the database failover recovery time (RTO), guarantees the availability of instances.

6 Instance types

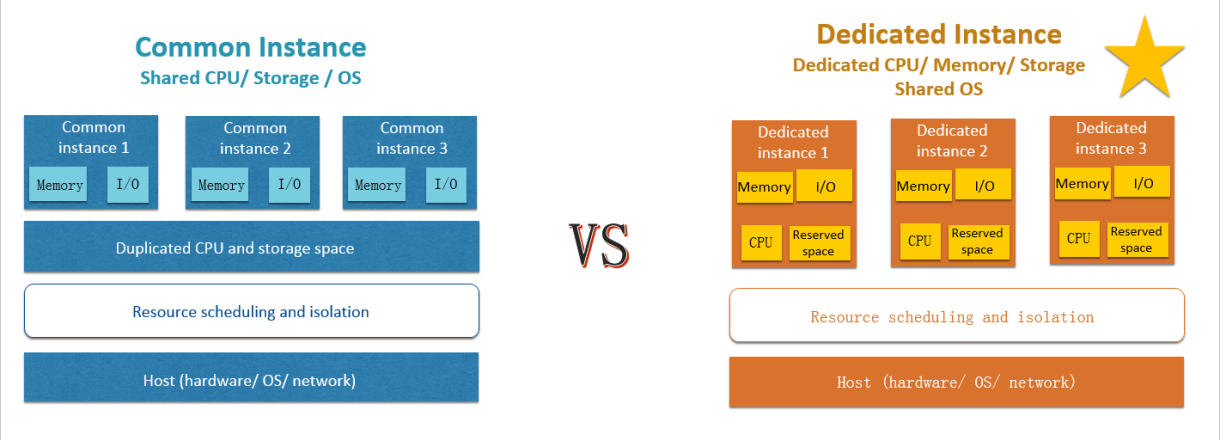
6.1 Instance type overview

Different types of ApsaraDB for RDS instances are provided: common instances, dedicated instances, and dedicated-host instances. To distinguish from the instance type provided before January 2017, which we defined as common instances, the new instance types are defined as dedicated instances, among which dedicated-host instances have top-level configurations. A dedicated-host instance can be upgrade to the top level so that it can use a full physical machine.

The following list shows the features and applicable scenarios of various instance types:

Types	Description	Applicable scenarios
Common instances	• It is a highly cost-effective instance type which maximizes utilization by reusing resources and allows you to enjoy the benefits of scale. • Its storage capacity is not subject to CPU/memory, allowing flexible configurations.	• Price-sensitive customers • Scenarios where high-performance stability is not required
Dedicated instances/dedicated-host instances	• The two types are new RDS instance types which feature fixed computing capabilities, storage space, and IO performance. • A number of fully dedicated CPU cores and threads are assigned to each instance to ensure long-term stability in computing performance. • The fixed storage space is reserved for higher stability. • Dedicated-host instances are dedicated instances with top-level configurations.	It is applicable for business scenarios where a database-centric system is used, such as finance, e-commerce, government, medium and large-size Internet businesses.

The differences between common and dedicated instances are shown in the following figure.



6.2 Instance type list

RDS for MySQL

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
Basic Edition	5.7	Common instances	mysql.n1.micro.1	1-core 1 GB	2,000	IOPS= min{30 x storage capacity, 20,000}	20 GB - 1000 GB
			mysql.n2.small.1	1-core 2 GB	2,000		
			mysql.n2.medium.1	2-core 4 GB	4,000		
			mysql.n4.medium.1	2-core 8 GB	6,000		20 GB - 2000 GB
			mysql.n4.large.1	4-core 16 GB	8,000		
			mysql.n4.xlarge.1	8-core 32 GB	10,000		
			mysql.n4.2xlarge.1	16-core 64 GB	15,000		
			mysql.n4.4xlarge.1	32-core 128 GB	20,000		

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
			mysql.n8.4xlarge.1	32-core 256 GB	64,000		
			mysql.n4.8xlarge.1	56-core 224 GB	64,000		
			mysql.n8.8xlarge.1	56-core 480 GB	64,000		
High-availability Edition	5.5/5.6/5.7	Common instances	rds.mysql.t1.small	1-core 1 GB	300	600	5 GB - 2,000 GB
			rds.mysql.s1.small	1-core 2 GB	600	1,000	
			rds.mysql.s2.large	2-core 4 GB	1,200	2,000	
			rds.mysql.s2.xlarge	2-core 8 GB	2,000	4,000	
			rds.mysql.s3.large	4-core, 8 GB	2,000	5,000	
			rds.mysql.m1.medium	4-core 16 GB	4,000	7,000	
			rds.mysql.c1.large	8-core 16 GB	4,000	8,000	
			rds.mysql.c1.xlarge	8-core 32 GB	8,000	12,000	
			rds.mysql.c2.xlarge	16-core 64 GB	16,000	14,000	5 GB - 3,000 GB
			rds.mysql.c2.xlp2	16-core 96 GB	24,000	16,000	
		Dedicated instances (with large memory)	mysql.x8.medium.2	2-core 16 GB	2,500	4,500	250 GB

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
			mysql.x8.large.2	4-core 32 GB	5,000	9,000	500 GB
			mysql.x8.xlarge.2	8-core 64 GB	10,000	18,000	1,000 GB
			mysql.x8.2xlarge.2	16-core 128 GB	20,000	36,000	2,000 GB or 3,000 GB
		Dedicated instances (with high CPU)	mysql.x4.large.2	4-core 16 GB	2,500	4,500	250 GB or 500 GB
			mysql.x4.xlarge.2	8-core 32 GB	5,000	9,000	500 GB or 1,000 GB
			mysql.x4.2xlarge.2	16-core 64 GB	10,000	18,000	1,000 GB, 2,000 GB or 3,000 GB
			mysql.x4.4xlarge.2	32-core 128 GB	20,000	36,000	2,000 GB or 3,000 GB
		Dedicated-host instances	rds.mysql.st.d13	30-core 220 GB	64,000	20,000	3,000 GB
			rds.mysql.st.h43	60-core 470 GB	100,000	50,000	3,000 GB
Financial Edition (formerly: Three-node Enterprise Edition)	5.6	Dedicated instances (with high CPU)	mysql.x4.large.3	4-core 16 GB	2,500	4,500	250 GB or 500 GB
			mysql.x4.xlarge.3	8-core 32 GB	5,000	9,000	500 GB or 1,000 GB
			mysql.x4.2xlarge.3	16-core 64 GB	10,000	18,000	1,000 GB, 2,000 GB or 3,000 GB

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
			mysql.x4.4xlarge.3	32-core 128 GB	20,000	36,000	2,000 GB or 3,000 GB
		Dedicated instances (with large memory)	mysql.x8.medium.3	2-core 16 GB	2,500	4,500	250 GB
			mysql.x8.large.3	4-core 32 GB	5,000	9,000	500 GB
			mysql.x8.xlarge.3	8-core 64 GB	10,000	18,000	1,000 GB
			mysql.x8.2xlarge.3	16-core 128 GB	20,000	36,000	2,000 GB or 3,000 GB
			mysql.x8.4xlarge.3	32-core 256 GB	40,000	72,000	3,000 GB
		Dedicated-host instances	mysql.st.8xlarge.3	60-core 470 GB	100,000	120,000	3,000 GB
Read-only instance	5.6/5.7	Common instances	rds.mysql.t1.small	1-core 1 GB	300	600	5 GB - 2,000 GB
			rds.mysql.s1.small	1-core 2 GB	600	1,000	
			rds.mysql.s2.large	2-core 4 GB	1,200	2,000	
			rds.mysql.s2.xlarge	2-core 8 GB	2,000	4,000	
			rds.mysql.s3.large	4-core, 8 GB	2,000	5,000	
			rds.mysql.m1.medium	4-core 16 GB	4,000	7,000	

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
			rds.mysql.c1.large	8-core 16 GB	4,000	8,000	
			rds.mysql.c1.xlarge	8-core 32 GB	8,000	12,000	
			rds.mysql.c2.xlarge	16-core 64 GB	16,000	14,000	5 GB - 3,000 GB
			rds.mysql.c2.xlp2	16-core 96 GB	24,000	16,000	
		Dedicated instances (with large memory)	mysqlro.x8.medium.1	2-core 16 GB	2,500	4,500	250 GB
			mysqlro.x8.large.1	4-core 32 GB	5,000	9,000	500 GB
			mysqlro.x8.xlarge.1	8-core 64 GB	10,000	18,000	1,000 GB
			mysqlro.x8.2xlarge.1	16-core 128 GB	20,000	36,000	2,000 GB or 3,000 GB
		Dedicated instances (with high CPU)	mysqlro.x4.large.1	4-core 16 GB	2,500	4,500	250 GB or 500 GB
			mysqlro.x4.xlarge.1	8-core 32 GB	5,000	9,000	500 GB or 1,000 GB
			mysqlro.x4.2xlarge.1	16-core 64 GB	10,000	18,000	1,000 GB, 2,000 GB or 3,000 GB
			mysqlro.x4.4xlarge.1	32-core 128 GB	20,000	36,000	2,000 GB or 3,000 GB

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
		Dedicated-host instances	rds.mysql.st.d13	30-core 220 GB	64,000	20,000	3,000 GB

RDS for SQL Server

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
Basic Edition	2012 Enterprise (formerly 2012)	Common instances	rds.mssql.s2.large	2-core 4 GB	Not limited	IOPS= min{30 x storage capacity, 20,000}	20 GB - 2000 GB
			rds.mssql.s2.xlarge	2-core 8 GB			
			rds.mssql.s3.large	4-core, 8 GB			
			rds.mssql.m1.medium	4-core 16 GB			
			rds.mssql.c1.large	8-core 16 GB			
			rds.mssql.c1.xlarge	8-core 32 GB			
			rds.mssql.c2.xlarge	16-core 64 GB			
	2012/2016 Web	Dedicated instances	mssql.x2.medium.w1	2-core 4 GB	Not limited	IOPS= min{30 x storage capacity, 20,000}	20 GB - 2000 GB
			mssql.x2.large.w1	4-core, 8 GB			

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
			mssql.x2.xlarge.w1	8-core 16 GB			
			mssql.x2.2xlarge.w1	16-core 32 GB			
			mssql.x4.medium.w1	2-core 8 GB			
			mssql.x4.large.w1	4-core 16 GB			
			mssql.x4.xlarge.w1	8-core 32 GB			
			mssql.x4.2xlarge.w1	16-core 64 GB			
High-availability Edition	2008 R2 Enterprise	Common instances	rds.mssql.s1.small	1-core 2 GB	600	1,000	10 GB - 2,000 GB
			rds.mssql.s2.large	2-core 4 GB	1,200	2,000	
			rds.mssql.s2.xlarge	2-core 8 GB	2,000	4,000	
			rds.mssql.s3.large	4-core, 8 GB	2,000	5,000	
			rds.mssql.m1.medium	4-core 16 GB	4,000	7,000	
			rds.mssql.c1.large	8-core 16 GB	4,000	8,000	
			rds.mssql.c1.xlarge	8-core 32 GB	8,000	12,000	

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
			rds.mssql.c2.xlarge	16-core 64 GB	16,000	14,000	
			rds.mssql.c2.xlp2	16-core 96 GB	24,000	16,000	
		Dedicated instances	mssql.x8.medium.2	2-core 16 GB	2,500	4,500	250 GB
			mssql.x8.large.2	4-core 32 GB	5,000	9,000	500 GB
			mssql.x8.xlarge.2	8-core 64 GB	10,000	18,000	1,000 GB
			mssql.x8.2xlarge.2	16-core 128 GB	20,000	36,000	2,000 GB
		Dedicated-host instances	rds.mssql.st.d13	30-core 220 GB	64,000	20,000	2,000 GB
			rds.mssql.st.h43	60-core 470 GB	100,000	50,000	2,000 GB

RDS for PostgreSQL

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
High-availability Edition	9.4	Common instances	rds.pg.t1.small	1-core 1 GB	100	600	5 GB - 2,000 GB
			rds.pg.s1.small	1-core 2 GB	200	1,000	
			rds.pg.s2.large	2-core 4 GB	400	2,000	

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
			rds.pg.s3.large	4-core, 8 GB	800	5,000	
			rds.pg.c1.large	8-core 16 GB	1,500	8,000	
			rds.pg.c1.xlarge	8-core 32 GB	2,000	12,000	
			rds.pg.c2.xlarge	16-core 64 GB	2,000	14,000	
		Dedicated instances (with large memory)	pg.x8.medium. 2	2-core 16 GB	2,500	4,500	250 GB
			pg.x8.large.2	4-core 32 GB	5,000	9,000	500 GB
			pg.x8.xlarge.2	8-core 64 GB	10,000	18,000	1,000 GB
			pg.x8.2xlarge.2	16-core 128 GB	12,000	36,000	2,000 GB
		Dedicated instances (with high CPU)	pg.x4.large.2	4-core 16 GB	2,500	4,500	250 GB or 500 GB
			pg.x4.xlarge.2	8-core 32 GB	5,000	9,000	500 GB or 1,000 GB
			pg.x4.2xlarge.2	16-core 64 GB	10,000	18,000	1,000 GB or 2,000 GB
			pg.x4.4xlarge.2	32-core 128 GB	12,000	36,000	2,000 GB or 3,000 GB
		Dedicated-host instances	rds.pg.st.d13	30-core 220 GB	4,000	20,000	3,000 GB
			rds.pg.st.h43	60-core 470 GB	4,000	50,000	3,000 GB

RDS for PPAS

Series	Version	Type	Type code	CPU/ Memory	Maximum number of connections	Maximum IOPS	Storage capacity
High-availability Edition	9.3 / 10	Common instances	rds.ppas.t1.small	1-core 1 GB (for compatibility testing)	100	1,200	150 GB
		Dedicated instances	ppas.x4.small.2	1-core 4 GB	200	5,000	250 GB
			ppas.x4.medium.2	2-core 8 GB	400	10,000	
			ppas.x8.medium.2	2-core 16 GB	2,500	15,000	
			ppas.x4.large.2	4-core 16 GB	2,500	20,000	250 GB or 500 GB
			ppas.x8.large.2	4-core 32 GB	5,000	30,000	
			ppas.x4.xlarge.2	8-core 32GB	5,000	40,000	250 GB or 500 GB
			ppas.x8.xlarge.2	8-core 64 GB	10,000	60,000	
			ppas.x4.2xlarge.2	16-core 64 GB	10,000	80,000	250 GB or 500 GB
			ppas.x8.2xlarge.2	16-core 128 GB	12,000	120,000	
			ppas.x4.4xlarge.2	32-core 128 GB	12,000	160,000	2000 GB or 3000 GB
			ppas.x8.4xlarge.2	32-core 256 GB	12,000	240,000	
		Dedicated-host instances	rds.ppas.st.h43	60-core 470 GB	12,000	450,000	

Historical instance types of RDS for MySQL

The following table lists the historical instance types of RDS for MySQL. They are no longer available when you create a new instance.

Type code	Number of CPU cores	Memory	Maximum number of connections	Maximum IOPS
rds.mys2.small	2	240 MB	60	150
rds.mys2.mid	4	600 MB	150	300
rds.mys2.standard	6	1,200 MB	300	600
rds.mys2.large	8	2,400 MB	600	1,200
rds.mys2.xlarge	9	6,000 MB	1,500	3,000
rds.mys2.2xlarge	10	12,000 MB	2,000	6,000
rds.mys2.4xlarge	11	24,000MB	2,000	12,000
rds.mys2.8xlarge	13	48,000 MB	2,000	14,000

Historical instance types of RDS for SQL Server

The following table lists the historical instance types of RDS for SQL Server. They are no longer available when you create a new instance.

Type code	Number of CPU cores	Memory	Maximum number of connections	Maximum IOPS
rds.mss1.small	6	1,000 MB	100	500
rds.mss1.mid	8	2,000 MB	200	1,000
rds.mss1.standard	9	4,000 MB	400	2,000
rds.mss1.large	10	6,000 MB	600	3,000
rds.mss1.xlarge	11	8,000 MB	800	4,000
rds.mss1.2xlarge	12	12,000 MB	1,200	6,000
rds.mss1.4xlarge	13	24,000MB	2,000	12,000
rds.mss1.8xlarge	13	48,000 MB	2,000	14,000

Historical instance types of RDS for PPAS

The following table lists the historical instance types of RDS for PPAS. They are no longer available when you create a new instance.

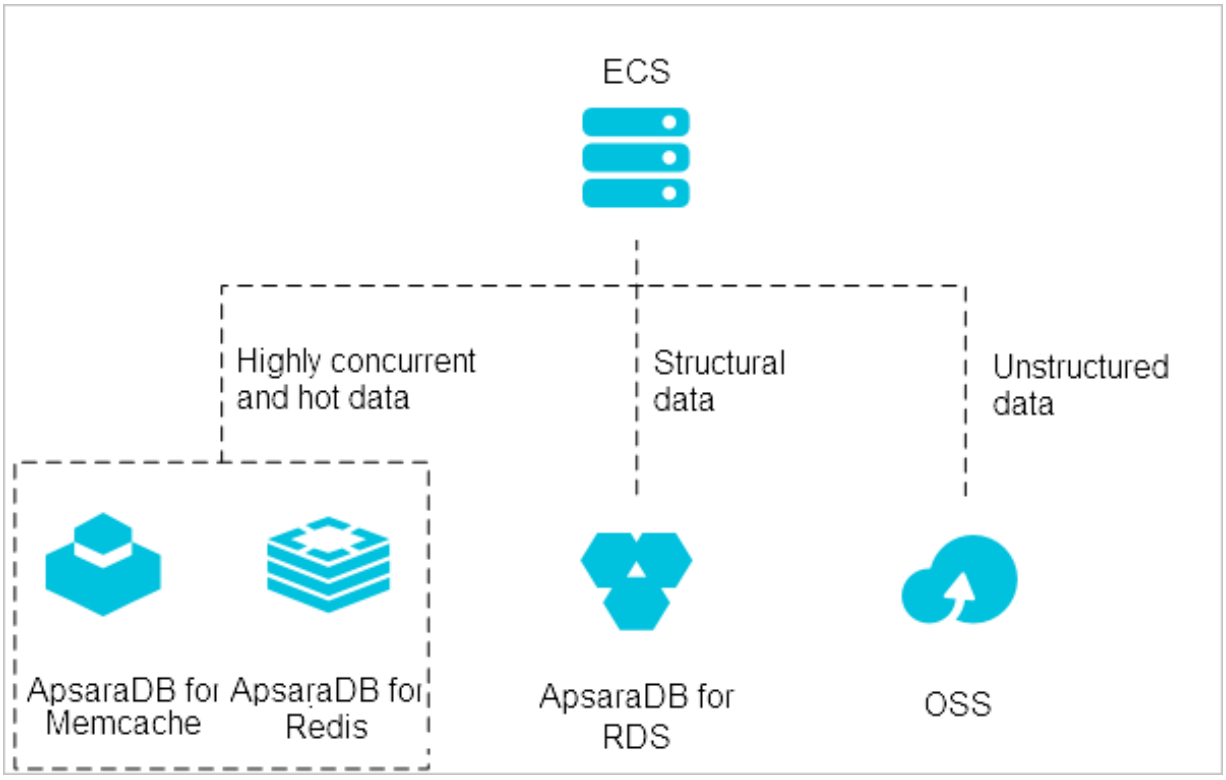
Type code	Number of CPU cores	Memory	Maximum number of connections	Maximum IOPS
rds.ppas.s1.small	1	2 GB	200	1,000
rds.ppas.s2.large	2	4 GB	400	2,000
rds.ppas.s3.large	4	8 GB	800	5,000
rds.ppas.m1.medium	4	16 GB	1,500	8,000
rds.ppas.c1.xlarge	8	32 GB	2,000	12,000
rds.ppas.c2.xlarge	16	64 GB	2,000	14,000

8 Typical applications

8.1 国际站没有

Remote disaster tolerance

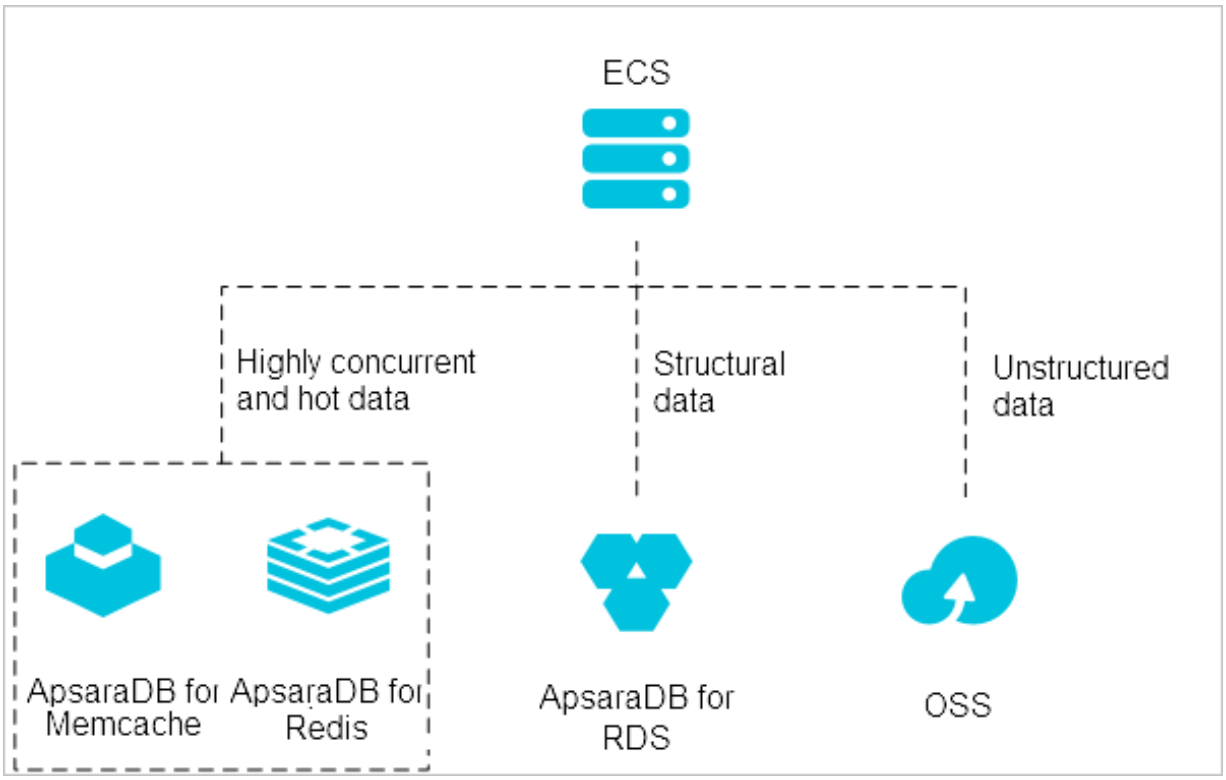
Alibaba RDS enables a user to create remote disaster recovery instances, and you can create remote disaster recovery instances to resist faults at the multi-zone level. In addition, the user passes through the DTs In addition, you can use the DTS to synchronize the database in the user -built equipment room to the RDS instance in any region on Alibaba RDS in real time. Even if the IDC is damaged, a data backup always exists on RDS.



- - For details on how to create local data together with RDS remote disaster recovery, please refer to [Creating a Real-time Synchronization Job].
- Alibaba Cloud database MySQL edition disaster recovery creation methods can be found in.

8.2 Diversified data storage

RDS supports diversified storage extension through storage products, such as Memcache, Redis, and OSS.



Cache data persistence

ApsaraDB for Memcache and ApsaraDB for Redis can be used together to form a high-throughput and low-latency storage solution.

These cache products have two characteristics:

- Fast response with a request delay of only a few milliseconds
- Higher QPS (queries per second) than RDS

For more information, see *Cached data persistence*.

Multi-structure data storage

OSS is an Alibaba Cloud storage service that features massive capacity, robust security, low cost, and high reliability. RDS and OSS can work together to form multiple data storage solutions.

For example, when RDS and OSS are used in a forum, resources such as the images of registered users or those posted on the forum can be stored in OSS to reduce the storage pressure on RDS.

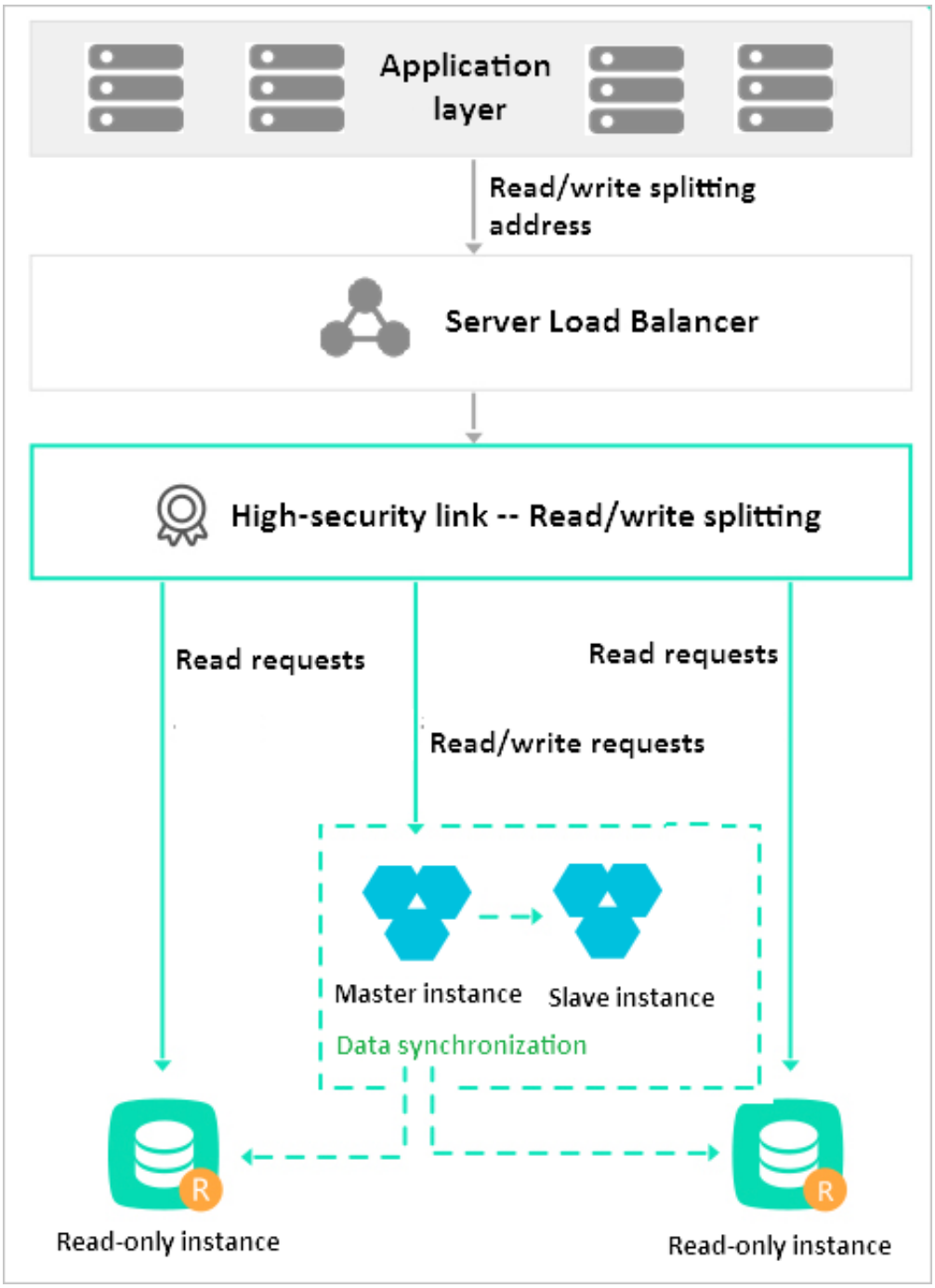
For more information, see *Multi-structure data storage*.

8.3 Open search

OpenSearch is a structured data search hosting service, which provides mobile application developers and Website owners with simple, efficient, stable, low-cost, and scalable search solutions. OpenSearch automatically synchronizes the data in RDS to realize various complex searches.

8.4 Read/Write splitting function

ApsaraDB for MySQL allows read-only instances to be directly attached to RDS in order to distribute the read pressure on the master instance. Each read-only instance has an independent connection string, and the read pressure can be automatically distributed on the application side.



For more information on creating read-only instances on ApsaraDB for MySQL, see *Create a read-only instance*. To enable read/write splitting, see *Enable read/write splitting*.

8.5 Big data analysis

The Open Data Processing Service (MaxCompute) is formerly known as the Big Data Computing Service (ODPS). It stores and computes structured data in batches, providing solutions for massive data warehouses as well as big data analysis and modeling.

With the data integration service, RDS data can be imported into MaxCompute to achieve large-scale data computing.

9 RDS usage instructions

- Instructions for RDS instance upgrade

  There is a transient disconnection of up to 30 seconds during RDS instance upgrade. To avoid service unavailability caused by interruptions, you must make preparations in advance and set auto-reconnection for the program and to RDS.

- Risks of intranet/Internet switchover

  During the switchover between the intranet and Internet, RDS may be disconnected with the server and the RDS IP address may change. After the switchover, update the connection address in the program in a timely manner.

- Rollback risks

  Currently, RDS only supports data rollback for the entire instance but not for a single table or database. It is important to make backups of crucial data before rollback to prevent data loss. If you need to roll back only partial tables or partial data, we recommend that you recover the data by creating a temporary instance or a clone instance. After a temporary instance or a clone instance is created successfully, the required data is directed returned to the production database. For more information, see *Recover data to the master instance through a temporary instance*.

- Instructions on RDS locking policies

  Currently, RDS only supports data rollback for the entire instance, not for a single table or database. It is important to make backups of crucial data before rollback to prevent data loss. We recommend that you create a temporary instance for data recovery if you only want to roll back part of the tables or part of the data. You must import the required data back to the production databases from the temporary instance. For more information, see *Recover data to the master instance through a temporary instance*.

- Instructions on RDS failover

  RDS adopts a master-slave mode to guarantee high availability. When the master node fails, RDS fails services over to the slave node within 30 seconds. During the failover, there may be an inaccessible period of (less than or equal to) 30 seconds. You must set auto-reconnection for your program to RDS to avoid service unavailability.

- Instructions on data synchronization mode for MySQL instances

- For MySQL 5.1, data synchronization between the master and slave nodes is in asynchronous mode. This mode guarantees high performance of instances, but it may lead to probabilistic data inconsistency between the master and slave nodes.
- For MySQL 5.5, data synchronization between the master and slave nodes is in semi-synchronous mode. This mode compromises write performance to some extent, but it can greatly reduce the probability of data inconsistency between the master and slave databases. If you have a high requirement on data reliability (for example, financial systems), we recommend that you purchase instances of MySQL 5.5 or later versions.
- For MySQL 5.6, data synchronization between the master and slave nodes uses GTID (a new feature in MySQL 5.6). This feature guarantees both the performance and data consistency.

- Notes of attentions for using RDS

  After purchasing an RDS instance, you do not need to conduct basic O&M (such as high availability guarantee, backup, and security patches) for databases. Pay attention to the following matters:

  - Check whether the CPU, IOPS, space, and the number of connections of your RDS instance are adequate. If not, you must optimize or upgrade your instance.
  - Check whether your RDS instance has performance problems, for example, whether there is a large number of slow SQL queries, whether the SQL statements need optimization, and whether there are extra or missing indexes.
  - Check whether your RDS instance has any web SQL injection warnings. If so, your databases may be vulnerable to web SQL injections. In this case, modify the application to prevent web SQL injection attacks.