

分布式关系型数据库 DRDS

快速入门

快速入门

快速入门概述

快速入门文档主要介绍购买 DRDS 实例、创建 DRDS 数据库和表、使用客户端连接 DRDS 进行数据库操作的完整流程。

DRDS 快速接入流程图：



步骤一：购买 DRDS 实例

本文档主要介绍如何购买 DRDS 实例及相关注意事项。

购买入口

第一种

登录阿里云主页，将鼠标依次移动到**产品>互联网中间件**，单击**分布式关系型数据库服务 DRDS** 进入 DRDS 产品主页。

在 DRDS 产品主页上，单击**立即购买**进入 DRDS 实例购买页面，根据提示完成DRDS实例购买。



第二种

登录 DRDS 控制台(<https://drds.console.aliyun.com>)。

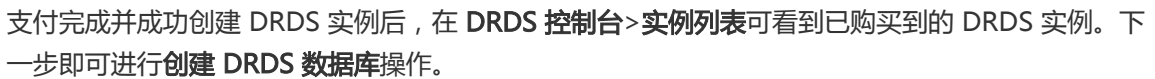
单击页面右上角创建实例按钮进入 DRDS 实例购买页面，根据提示完成实例购买。



购买步骤

在购买页面，根据需要选择付费方式以及对应的实例规格，单击**立即购买**。

实例规格、付费方式等详细信息也可参考实例计费文档。



3

创建 DRDS 数据库和创建普通 MySQL 数据库有两点不同：

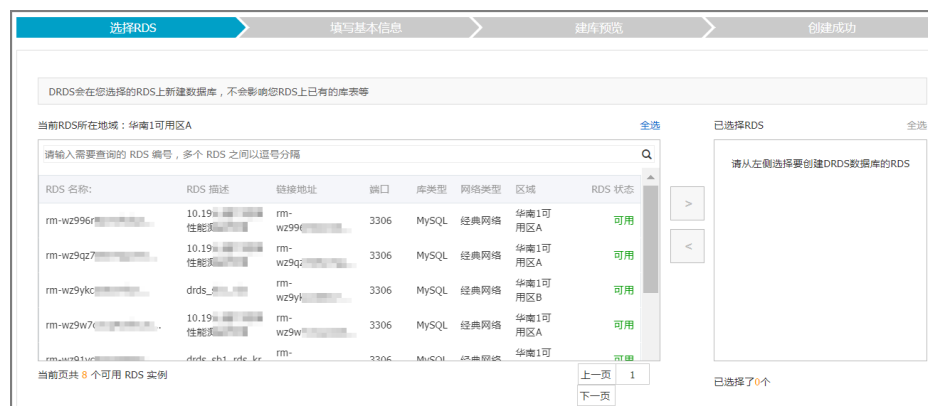
- DRDS 创建数据库操作只能在控制台上进行；
- DRDS 创建数据库需要选择若干（包含一个）RDS 作为数据存储节点（如果没有 RDS，请先购买）。

创建数据库步骤

在 DRDS 控制台左侧菜单栏选择**实例列表**，单击需要建数据库的 DRDS 实例进入实例基本信息页。

单击页面右上角**创建数据库**按钮。

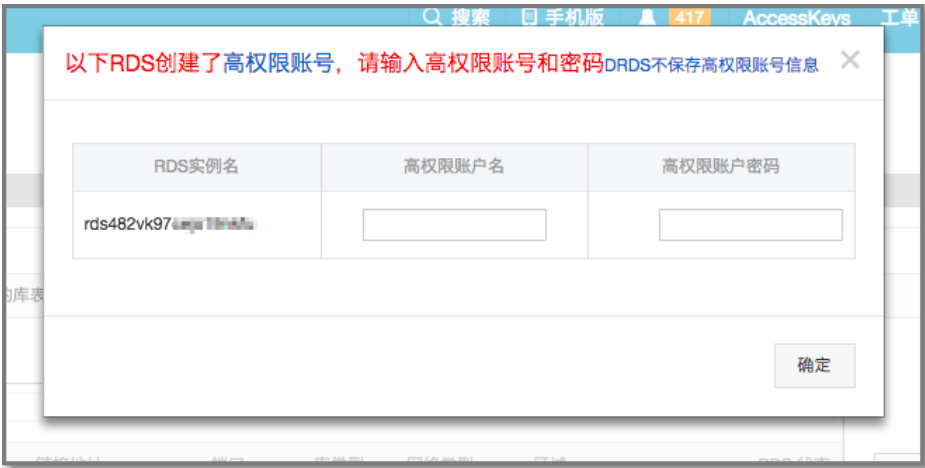
根据需要选择 RDS 作为 DRDS 的存储节点。



说明：

可选择作为 DRDS 数据存储节点的 RDS 需要同时满足以下几个条件：类型为 MySQL、状态为运行中、所在区域和 DRDS 所在区域匹配。

如果选择的 RDS 创建了高权限账号，则单击**下一步**会要求填写高权限账号和密码。DRDS 无法也不能直接获取到相应 RDS 的账号密码，填写的高权限账号和密码也不会被存储，只会在接下来的操作中临时使用。



按提示填写基本信息。

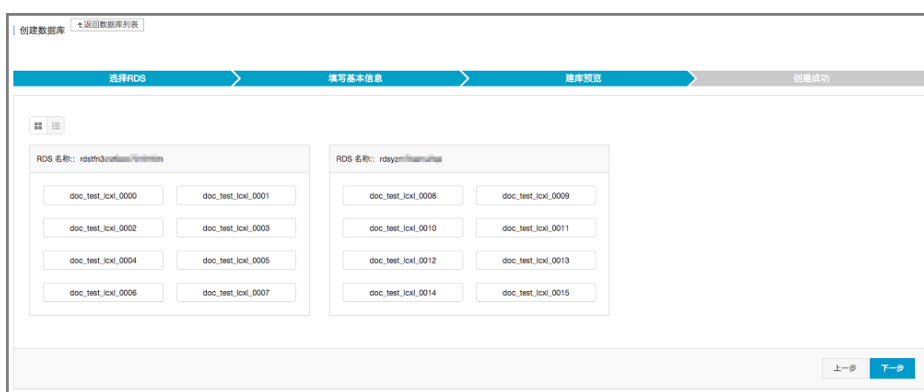


创建类型对应使用 DRDS 的不同方式。

- **拆分**：即分库分表，将数据按照拆分规则分拆到多个库表中，由 DRDS 代理 SQL 执行。拆分涉及到数据导入导出、SQL 功能/性能测试和改造，对应用功能和性能会有一些的影响。
- **非拆分**：将已有的 RDS 数据库交由 DRDS 进行代理访问，实现读写分离的功能。无需进行数据导入，无需修改程序代码，修改数据库连接串和用户名密码即可。

建库预览，确认信息正确。

DRDS 默认会在 1 个 RDS 上创建 8 个物理库，总的物理库数量为选择的 RDS 数量乘以 8。



预览确认没有问题后，单击**下一步**进行 DRDS 数据库的实际创建。



DRDS 会通过 RDS API 的方式或者使用高权限账号进行库和对应账号、DRDS 系统表和配置的创建。请耐心等待 DRDS 数据库创建完成。

您也可以在 DRDS 数据库列表查看 DRDS 数据库的创建状态。



步骤三：创建 DRDS 表

创建完 DRDS 数据库后，如同普通单机数据库，DRDS 也需要创建表。但是语法有所差异，主要表现在 DRDS 建表语句对数据拆分信息的表达上。

DRDS 建表步骤

登录 DRDS 控制台，在左侧菜单栏单击**实例列表**。

在实例列表页单击需要操作的实例名称，并单击左侧菜单栏**数据库列表**。

单击需要建表的数据库名称进入数据库**基本信息**页，复制 DRDS 连接串。

doc_test 返回数据库列表

快捷操作

创建表/修改表结构 修复分库链接

基本信息

数据库名: doc_test 工作模式: 拆分

实例ID: drds-xxxxxx 状态: 运行中

内网地址

网络类型: 经典网络 用户名: doc_test

内网地址: drds-xxxxxx.drds.aliyuncs.com 内网端口: 3306

命令行连接地址: mysql -h{DRDS_IP_ADDRESS} -P{DRDS_PORT} -u{user} -p{password} -D{DRDS_DBNAME}

通过 MySQL 客户端进行服务连接。

例如使用 MySQL 命令行连接：

```
mysql -h${DRDS_IP_ADDRESS} -P${DRDS_PORT} -u${user} -p${password} -D${DRDS_DBNAME}
```

注意：

- 如果 DRDS 实例为专享实例，默认只提供内网地址，建议使用相同地域的 ECS 安装 MySQL 命令行连接。
- 如果 DRDS 实例为共享实例，可以使用其公网地址，从外网机器或者办公机器进行连接使用。

执行 DRDS DDL 进行建表。DDL 详细语法请参考 DRDS DDL 语法说明文档。

```
//DRDS DDL
CREATE TABLE shard_table(
id int,
name varchar(30),
primary key(id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(id) tpartition by hash(id) tpartitions 3;
```

步骤四：连接 DRDS

在控制台创建 DRDS 实例、创建 DRDS 数据库、创建表之后，需要通过连接 DRDS 进行后续的数据库操作。

连接 DRDS

进入 **DRDS 数据库详情页**找到 DRDS 的连接信息，由 “ip(domain)”、“port”、“dbname”、“user”、“password” 组成。



The screenshot shows the 'doc_test' database details page in the DRDS console. It includes a '快捷操作' (Quick Actions) section with buttons for '创建表/修改表结构' and '修复分库链接'. The '基本信息' (Basic Information) section displays the database name 'doc_test', instance ID, and status '运行中'. The '内网地址' (Intranet Address) section shows the network type '经典网络', IP address 'drds-xxxxx.aliyuncs.com', and port '3306'. The '命令行连接地址' (Command Line Connection Address) section provides the MySQL connection string: 'mysql -h[drds-xxxxx.aliyuncs.com] -P3306 -u[doc_test] -p[]'. Red boxes highlight the IP address, port, user, and password fields in the connection string.

doc_test 返回数据库列表	
快捷操作	
创建表/修改表结构 修复分库链接	
基本信息	
数据库名: doc_test	工作模式: 拆分
实例ID: drds-xxxxx	状态: 运行中
内网地址	
网络类型: 经典网络	用户名: doc_test
内网地址: drds-xxxxx.aliyuncs.com	内网端口: 3306
命令行连接地址: mysql -h[drds-xxxxx.aliyuncs.com] -P3306 -u[doc_test] -p[]	

获取到连接信息后，可通过**第三方工具**或者**程序代码**两种方式进行连接。

第三方工具

DRDS 遵循 MySQL 官方交互协议，所以支持第三方工具的连接和使用。

```
//MySQL 命令行连接
mysql -h${DRDS_IP_ADDRESS} -P${DRDS_PORT} -u${user} -p${password} -
D${DRDS_DBNAME}
```

注意：DRDS 完全兼容 MySQL 官方命令行客户端(版本5.1+)。DRDS 不支持 MySQL 的历史版本(如3.x、4.x 等版本)的指令和不常用指令，因此 DRDS 只承诺第三方 GUI 客户端可执行基础的数据库操作，包括数据的增删改查和 DDL 操作。

程序代码

主要通过 MySQL 官方驱动或者第三方符合 MySQL 官方交互协议的程序进行连接。

DRDS 支持的客户端工具

- MySQL 命令行 (推荐)
- MySQL Workbench (推荐)
- SQLyog
- Sequel Pro
- Navicat for MySQL

DRDS 支持的程序驱动

JDBC Driver for MySQL (Connector/J)

```
//JDBC
Class.forName("com.mysql.jdbc.Driver");
Connection conn =
DriverManager.getConnection("jdbc:mysql://drdsxxxxx.drds.aliyuncs.com:3306/doc_test","doc_te
st","doc_test_password");
//...
conn.close();
```

- Python Driver for MySQL (Connector/Python)
- C++ Driver for MySQL (Connector/C++)
- C Driver for MySQL (Connector/C)
- ADO.NET Driver for MySQL (Connector/NET)
- ODBC Driver for MySQL (Connector/ODBC)
- PHP Drivers for MySQL (mysqli, ext/mysqli, PDO_MYSQL, PHP_MYSQLND)
- Perl Driver for MySQL (DBD::mysql)
- Ruby Driver for MySQL (ruby-mysql)

应用端连接池配置示例

推荐使用 Druid 连接池连接 DRDS , 关于 Druid 的详细信息请参考 [Druid Github 资源](#)。

配置示例：

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-
method="close">
<property name="url" value="jdbc:mysql://drdsxxxxx.drds.aliyuncs.com:3306/doc_test" />
<property name="username" value="doc_test" />
<property name="password" value="doc_test_password" />
```

```
<property name="filters" value="stat" />

<property name="maxActive" value="100" />
<property name="initialSize" value="20" />
<property name="maxWait" value="60000" />
<property name="minIdle" value="1" />

<property name="timeBetweenEvictionRunsMillis" value="60000" />
<property name="minEvictableIdleTimeMillis" value="300000" />

<property name="testWhileIdle" value="true" />
<property name="testOnBorrow" value="false" />
<property name="testOnReturn" value="false" />

<property name="poolPreparedStatements" value="true" />
<property name="maxOpenPreparedStatements" value="20" />

<property name="asyncInit" value="true" />
</bean>
```