

分布式关系型数据库 DRDS

快速入门

快速入门

上手指南

概述

DRDS 快速接入流程图：



购买 DRDS 实例

- 登录 阿里云主页，将鼠标依次移动到 产品>云计算基础服务>数据库，单击 分布式关系型数据库服务 DRDS 进入 DRDS 产品主页。

分布式关系型数据库服务 DRDS

DRDS 是阿里巴巴集团自主研发的分布式数据库中间件产品，专注于解决单机关系型数据库扩展性问题。具备轻量(无状态)、灵活、稳定、高效等特性，稳定运行11年，经历历届双十一核心交易业务和各类行业业务的考验，是您值得信赖的选择。[新用户首购99元，点击了解详情。](#)

新用户首购99元 | 管理控制台 | 文档&API | 产品价格 | 解决方案 | 预约数据库专家服务

- 持续运维**
- 安全可控**
- 分库分表**
依托水平/垂直拆分，支持并行计算，解决高并发、存储容量、在线计算的扩展性难题
- 稳定可靠**
基于稳定的 RDS for MySQL 构建，通过分库分表让 DRDS 数据库整体处于最佳稳定状态
- 持续运维**
以业务 7*24 小时运行标准设计，提供弹性升降配、在线扩容、SQL 审计分析、只读实例等能力
- 安全可控**
支持可用区实例、账号权限、VPC、备份恢复、SQL 闪回等能力，极尽所能，全方位保障业务安全

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。

在购买页面，根据需要选择付费方式以及对应的 实例规格，选择存储类型为 RDS MySQL，单击**立即购买**。规格、付费方式等信息可参考 实例计费文档

分布式关系型数据库服务DRDS(包月)

包年包月 | 按量付费

地域: 华南1 (深圳) 华北1 (青岛) 华北2 (北京) 华东2 (上海) 中国 (香港) 新加坡 **华北5 (呼和浩特)** 美国 (弗吉尼亚) 华北3 (张家口) 华东1 (杭州)

可用区: 可用区A

实例类型: 专享实例

实例网络类型: 专有网络

VPC: test

虚拟交换机: test

MySQL版本: **MySQL 8** MySQL 5
与 MySQL 8.0 完全兼容

存储类型: **RDS MySQL** POLARDB MySQL
DRDS 提供私有 RDS MySQL 资源, 性能表现更优秀, 服务链路更稳定, 售卖价格更优惠, 详情参考 [文档说明](#)

实例系列: 入门版 **标准版** 企业版
面向高并发、复杂查询及轻量分析的业务场景, 详情参考 [文档说明](#)

实例规格: **16核64GB** 32核128GB 48核192GB 64核256GB 96核384GB
128核512GB
所有 DRDS 实例均为专享实例, 每个实例最少提供 2 个节点保证高可用, 各规格差异详见 [实例规格详情](#)

购买数量: 1

购买时长: 1 个月 2 3 4 5 6 7 8 9 1年 2年 3年 自动续费

当前配置

地址: 华北5 (呼和浩特)

可用区: 可用区A

实例类型: 专享实例

实例网络类型: 专有网络

VPC: test

虚拟交换机: test

MySQL版本: MySQL 8

存储类型: RDS MySQL

实例系列: 标准版

实例规格: 16核64GB

购买数量: 1

购买时长: 1个月

配置费用:

[立即购买](#) [加入购物车](#)

确认订单无误后, 单击 **去支付** 进行支付和实例创建。

确认订单

确认订单

支付

支付成功

商品信息	支付方式	数量	折扣	资费
分布式关系型数据库服务DRDS(包月) 实例网络类型: 专有网络 MySQL版本: MySQL 8 存储类型: RDS MySQL 实例系列: 标准版 实例规格: 16核64GB 实例类型: 专享实例 地域: cn-huhehaote VPC: test 购买时长: 1个月 虚拟交换机: test 可用区: cn-huhehaote-a	预付费	1	¥4,654.99	¥0.01

发票: 订单对应可开发票的类型和抬头为您在用户中心-发票信息管理中设置的信息

分布式关系型数据库服务DRDS(包月) 服务协议

[去支付](#)

支付完成并成功创建 DRDS 实例后, 在 **DRDS 控制台 > 实例列表** 可看到已购买到的 DRDS 实例。

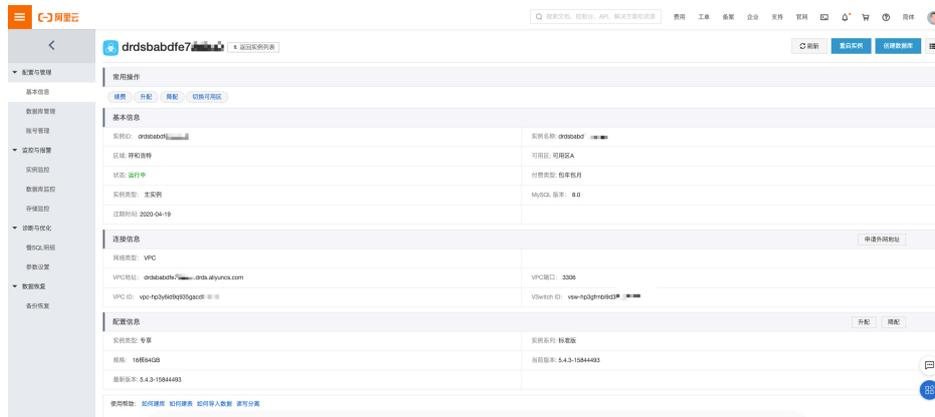
The screenshot shows the '实例列表' (Instance List) page in the DRDS console. The table contains one instance:

实例名称	类型	网络类型	付费类型	状态	可用区	实例类型	实例系列	实例规格	创建时间	操作
drds-ali-hn drds-ali-hn	专享	VPC	包年包月 (1天起计费)	运行中	呼和浩特 可用区A	主实例	标准版	16核64GB	2020-03-18 14:28:17	刷新 续费 更多

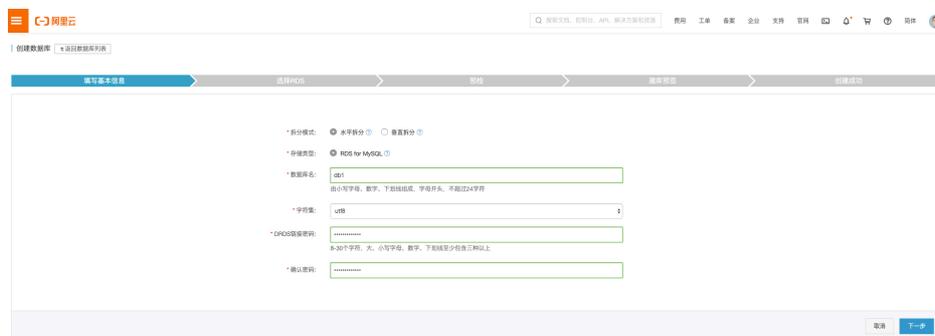
共有 1 条, 每页显示: 20 条

创建 DRDS 数据库

- 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，填写数据库基本信息，选择 **水平拆分** 模式



点击 **下一步**，创建 DRDS 专属私有 RDS MySQL 资源，或选择已创建的 私有 RDS MySQL 资源。

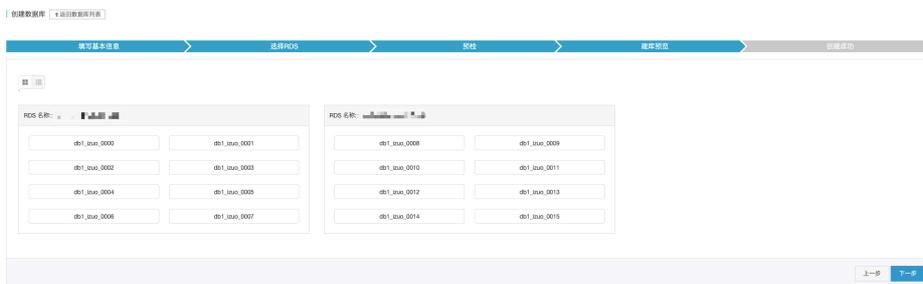
- DRDS 提供专属私有 RDS MySQL 资源，相对于用户自行购买的 RDS 资源，DRDS 私有 RDS MySQL 性能表现更优秀，服务链路更稳定，售卖价格更优惠。
- 为提升服务稳定性，避免用户自行修改 RDS MySQL 参数等运维操作，DRDS 私有 RDS MySQL 的常规运维操作需在 DRDS 控制台内完成。

[购买新RDS](#) 使用已有RDS

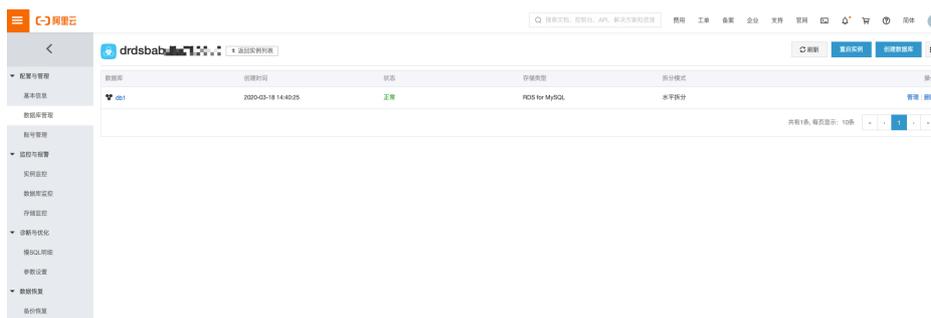
| DRDS 私有 RDS 按量后付费



建库预览，DRDS 默认会在 1 个 RDS 实例上创建 8 个物理库，总物理分库数量为选择的 MySQL 实例数量 * 8。



- 单击 **下一步** 进行 DRDS 数据库创建，DRDS 数据库列表可查看创建状态。请耐心等待创建完成。



连接 DRDS 进行 SQL 操作

- 进入 **DRDS 数据库详情页** 找到连接信息，由 “ip(domain)”、“port”、“dbname”、“user” 组成。您也可以参考 [外网访问 文档](#) 临时开通公网地址，从您本地机器连接进行试用，正式使用请记得关闭，确保安全。



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。

MySQL 命令行

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

注意：

DRDS 兼容 MySQL 官方命令行客户端，支持 5.X 系列版本，暂不支持 8.0 客户端，如遇密码错误等提示信息，请尝试安装使用 5.X 版本客户端。

第三方工具

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

注意：

第三方 GUI 客户端可执行基础的数据操作，包括数据的增删改查和 DDL 操作，对于工具高级特性，DRDS 可能并不支持。推荐使用云上 DMS 产品。

通过 MySQL 官方驱动或者符合 MySQL 官方交互协议的程序进行连接

- JDBC Driver for MySQL (Connector/J)
- 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)

以 JAVA 为例

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

应用连接池配置示例 (JAVA)

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](#)。

Spring 配置

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
<property name="driverClassName" value="com.mysql.jdbc.Driver" />
<!-- 基本属性 URL、user、password -->
<property name="url"
value="jdbc:mysql://drdsxxxxx.drds.aliyuncs.com:port/db?autoReconnect=true&rewriteBatchedStatements=true&socketTimeout=30000&connectTimeout=3000" />
<property name="username" value="abc" />
<property name="password" value="123456" />
<!-- 配置初始化大小、最小、最大 -->
<property name="maxActive" value="20" />
<property name="initialSize" value="3" />
<property name="minIdle" value="3" />
<!-- maxWait 获取连接等待超时的时间 -->
<property name="maxWait" value="60000" />
<!-- timeBetweenEvictionRunsMillis 间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒 -->
<property name="timeBetweenEvictionRunsMillis" value="60000" />
<!-- minEvictableIdleTimeMillis 一个连接在池中最小空闲的时间，单位是毫秒-->
<property name="minEvictableIdleTimeMillis" value="300000" />
<!-- 检测连接是否可用的 SQL -->
<property name="validationQuery" value="select 'z' from dual" />
<!-- 是否开启空闲连接检查 -->
<property name="testWhileIdle" value="true" />
<!-- 是否在获取连接前检查连接状态 -->
<property name="testOnBorrow" value="false" />
<!-- 是否在归还连接时检查连接状态 -->
<property name="testOnReturn" value="false" />
<!-- 是否在固定时间关闭连接。此参数默认可以不加，但是增加此参数可以均衡后端服务节点参数 -->
<property name="phyTimeoutMillis" value="1800000" />
```

```
</bean>
```

SQL 基本操作

查看数据库

使用 SHOW DATABASES 语句查看数据库

```
show databases;
```

创建、查看和删除表

使用 CREATE TABLE 语句创建表

```
-- 拆分表
CREATE TABLE multi_db_single_tbl(
  id int auto_increment,
  name varchar(30),
  primary key(id)
) dbpartition by hash(id);

-- 单表
CREATE TABLE single_tbl(
  id int,
  name varchar(30),
  primary key(id)
);
```

使用 SHOW CREATE 语句查看建表语句

```
show create table multi_db_single_tbl;

show create table single_tbl;
```

使用 DROP TABLE 语句删除表

```
drop table multi_db_single_tbl;

drop table single_tbl;
```

使用 SHOW TABLES 语句查看数据库中的所有表

```
show tables;
```

加列、减列、修改数据类型

使用 ALTER TABLE 语句加列、修改列、减列

```
alter table multi_db_single_tbl add column textcol text;  
alter table multi_db_single_tbl modify column textcol varchar(40);  
alter table multi_db_single_tbl drop column textcol;
```

创建、查看和删除本地索引

创建本地索引

```
create index idx_name on multi_db_single_tbl (name);
```

or

```
alter table multi_db_single_tbl add index idx_name(name);
```

查看本地索引

```
show index from multi_db_single_tbl;
```

删除本地索引

```
drop index idx_name on multi_db_single_tbl;
```

or

```
alter table multi_db_single_tbl drop index idx_name;
```

增删改查数据

使用 INSERT 语句向表内插入数据

```
insert into multi_db_single_tbl (name) values ('test_name');  
insert into multi_db_single_tbl (name) values ('test_name');  
insert into multi_db_single_tbl (name) values ('test_name'),('test_namexx');
```

使用 SELECT 语句检索表内数据。例如：

```
select * from multi_db_single_tbl;
```

使用 UPDATE 语句修改表内数据。例如：

```
update multi_db_single_tbl set name='zzz' where id in (100001,100002,100003,100004);
```

使用 DELETE 语句删除表内数据：

```
delete from multi_db_single_tbl where id = 100002;
```

创建、授权和删除用户

使用 CREATE USER 语句创建一个用户 drdsuser，密码为 123456

```
create user drdsuser@'%' identified by '123456';
```

授权用户 drdsuser 可检索数据库 sample_db 内的表

```
grant insert,update,delete,select on sample_db.* to drdsuser@'%';
```

查询用户 drdsuser 的权限

```
show grants for drdsuser@'%'
```

删除用户 drdsuser

```
drop user drdsuser@'%'
```