Alibaba Cloud

云原生分布式数据库 PolarDB-X 快速入门

文档版本: 20220506



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。

- 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。
- 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。
- 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
⚠ 危险	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。	
▲ 警告	该类警示信息可能会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 时间约十分钟。
〔) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等,不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面,单击 确定 。
Courier字体	命令或代码。	执行 cd /d C:/window 命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	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。	switch {act ive st and}

目录

1.创建实例	05
2.创建数据库	06
3.创建表	09
4.连接到数据库	10
5.执行SQL基本操作	12

1.创建实例

本文介绍如何创建PolarDB-X 1.0实例及相关注意事项。

操作步骤

- 1. 您可以通过以下任意一种方式进入PolarDB-X 1.0实例购买入口。
 - 购买入口一
 - a. 登录阿里云主页。
 - b. 将鼠标依次移动到产品 > 数据库 > 关系数据库,单击云原生分布式数据库PolarDB-X,进入PolarDB-X 1.0产品主页。
 - c. 单击**立即购买**,进入PolarDB-X 1.0实例购买页面。
 - 。 购买入口一
 - a. 登录云原生分布式数据库控制台。
 - b. 在控制台页面右上角,单击创建实例,进入PolarDB-X 1.0实例购买页面。
- 2. 商品类型选择PolarDB-X 1.0 包年包月。
- 3. 在购买页面,设置如下选项。

选项	说明
地域和可用区	通过选择地域和可用区配置实例所在的物理位置。
	⑦ 说明 目前PolarDB-X 1.0实例仅支持新加坡和中国(香港)。
实例类型	固定为 专享实例 ,不可变更。
实例网络类型	固定为 专有网络 ,不可变更。
	⑦ 说明 配置专有网络前,您需要提前准备相应地域和可用区内的VPC和虚拟交换机,详情请参见创建和管理专有网络。
实例系列	您可以针对不同业务场景选择不同系列,您可以选择 入门版、标准版、企业版 或 至 尊版。
实例规格	通过选择规格配置实例物理规格CPU核数和内存大小。
购买时长	请选择实例购买的时长。
购买数量	选择需要购买的实例数量。

4. 单击立即购买。

- 5. 在确认订单页,确认订单详情并勾选服务协议,单击去支付。
- 6. 在**支付**页,确认待支付订单和支付方式,单击**支付**。
- 7. 支付成功后,一般需要1~5分钟开通服务。之后您可以返回控制台的实例列表页查看新创建的实例。

2.创建数据库

PolarDB-X 1.0创建的数据库构建于RDS实例之上,出于OLTP业务稳定性的考虑,建议您选择新的RDS实例用于PolarDB-X 1.0数据库的创建。

使用限制

目前只支持在控制台上创建PolarDB-X 1.0数据库,不支持SQL创建PolarDB-X 1.0数据库。

操作步骤

- 1. 登录云原生分布式数据库控制台。
- 2. 在页面左上角选择目标实例所在地域。
- 3. 找到目标实例,单击实例ID,进入目标实例基本信息页。
- 4. 在实例基本信息页的右上角,单击创建数据库。
- 5. 在弹出的对话框中,根据您的需要选择拆分模式并设置相应参数。
 - 若您选择拆分模式为水平拆分,请参见如下步骤。

a. 在填**写基本信息**页,设置如下参数后,单击**下一步**。

参数	说明	
拆分模式	选择水平拆分。	
	⑦ 说明 PolarDB-X 1.0可以将数据按照拆分规则分拆到多个库表中。 通过水平拆分,实现数据库的横向扩展,可线性提升数据库整体的存储容 量以及并发吞吐。	
存储类型	您可以选择存储类型为 POLARDB for MySQL(推荐) 或 RDS for MySQL (本文以RDS for MySQL为例)。	
	输入数据库名称。	
数据库名	 ⑦ 说明 数据库名称需满足如下要求: ● 由小写字母、数字和下划线(_)组成。 ● 以字母开头,且长度不超过24字符。 	
字符集	您可以选择字符集为utf8、gbk、latin1或utf8mb4。	
链接密码	输入数据库链接密码。	
	 ⑦说明 链接密码需满足如下要求: 密码长度需为8~30个字符。 需至少包含大写字母、小写字母、数字、下划线(_)中的任意 三种。 	
确认密码	再次输入数据库链接密码进行确认。	

b. 在选择RDS页,选中目标RDS实例,单击 >图标将其从左侧待选框移动到右侧已选框中,单

击下一步。

○ 若您选择拆分模式为垂直拆分,请设置如下参数后,单击下一步。

参数	说明
	选择 垂直拆分 。
拆分模式	② 说明 您可以批量导入不同PolarDB MySQL集群或RDS MySQL实例上已有的多个数据库,利用PolarDB-X 1.0实例级账号进行统一授权管理,通过一个PolarDB-X 1.0链接即可完成横跨多个RDS实例数据库间的联合查询及事务写入操作。

参数	说明
存储类型	您可以选择存储类型为POLARDB for MySQL(推荐)或 RDS for MySQL(本 文以RDS MySQL为例)。
添加数据库	 i. 在RDS实例待选框中选中目标RDS实例。 ii. 从目标RDS实例已有的数据库中选中目标数据库,单击 > 将其移动到右边已选 框。
字符集	您可以选择字符集为utf8、gbk、latin1或utf8mb4。
账号类型	您可以创建新账号或选择已有账号。
数据库账号	输入需要创建的新账号名称。
链接密码	 输入数据库链接密码。 ⑦ 说明 链接密码需满足如下要求: 密码长度需为8~30个字符。 需至少包含大写字母、小写字母、数字、下划线(_)中的任意三种。
确认密码	再次输入数据库链接密码进行确认。仅当选择水平拆分模式或为垂直拆分模式创建 新账号时支持该参数。

6. 在**预检**页面,等待预检结果显示为**成功**,单击下一步。

7. 在建库预览页面确认建库信息,单击下一步完成数据库创建。

3.创建表

创建完PolarDB-X 1.0数据库后,您还需要为数据库创建表。本文将介绍如何创建PolarDB-X 1.0表。

- 1. 登录云原生分布式数据库控制台。
- 2. 在页面左上角选择目标实例所在地域。
- 3. 找到目标实例,单击实例ID,进入目标实例基本信息页。
- 4. 在左侧导航栏中,单击数据库管理。
- 5. 在数据库列表页,找到目标数据库,单击数据库ID,进入数据库基本信息页。
- 6. 在VPC地址区域,找到命令行链接地址获取连接信息。
- 7. 获取连接信息后,您可以使用如下MySQL命令行连接DRDS数据库:

mysql -h\${DRDS_ENDPOINT} -P\${DRDS_PORT} -u\${user} -p\${password} -D\${DRDS_DBNAME}

⑦ 说明 PolarDB-X 1.0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进行命令行连接。

8. 执行如下语句创建表。

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

4.连接到数据库

本文将介绍如何连接PolarDB-X 1.0数据库。

操作步骤

- 1. 登录云原生分布式数据库控制台。
- 2. 在页面左上角选择目标实例所在地域。
- 3. 找到目标实例,单击实例ID,进入目标实例基本信息页。
- 4. 在左侧导航栏中,单击数据库管理。
- 5. 在数据库列表页,找到目标数据库,单击数据库ID,进入数据库基本信息页。
- 6. 在VPC地址区域,找到命令行链接地址,获取连接信息。
- 7. 获取连接信息后,您可以通过如下方式进行连接:
 - 如果您的服务器安装了MySQL,您可以通过如下MySQL命令行连接PolarDB-X 1.0数据库。

```
//MySQL 命令行连接
mysql -h${DRDS IP ADDRESS} -P${DRDS PORT} -u${user} -p${password} -D${DRDS DBNAME}
```

- PolarDB-X 1.0支持通过如下第三方客户端进行连接,您可以去对应的官方网站下载客户端。
 - MySQLWorkbench(推荐)
 - SQLyog
 - Sequel Pro
 - Navicat for MySQL

⑦ 说明 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,对于工具高级特性,PolarDB-X 1.0可能并不支持。

- PolarDB-X 1.0支持通过如下符合MySQL官方交互协议的第三方程序代码进行连接:
 - 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)

连接示例

• JDBC Driver for MySQL (Connector/J) 程序代码示例。

//JDBC

```
Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection("jdbc:mysql://drdsxxxx.drds.aliyuncs.com:3
306/doc_test","doc_test","doc_test_password");
//...
conn.close();
```

• 应用端连接池配置示例。

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" d</pre>
estroy-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>
```

```
    ⑦ 说明 推荐使用Druid连接池连接PolarDB-X 1.0,关于Druid的详细信息请参见Druid Git hub资源。
```

5.执行SQL基本操作

本文介绍如何在PolarDB-X 1.0中执行常用的SQL语句。

- 查看数据库
- 创建、查看和删除表
- 增加列、删除列、修改列的数据类型
- 创建、查看和删除本地索引
- 增删改查表数据

查看数据库

查看数据库:

SHOW DATABASES;

创建、查看和删除表

- 创建表。
 - 创建单表:

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

○ 创建一个对目标列(如ID列)使用HASH函数进行拆分的表:

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

- 查看建表语句。
 - 查看单表建表语句:

SHOW CREATE TABLE single_tbl;

○ 查看通过HASH函数进行拆分的表的建表语句:

SHOW CREATE TABLE multi_db_single_tbl;

- 删除表。
 - 删除单表:

DROP TABLE single_tbl;

○ 删除通过HASH函数进行拆分的表:

DROP TABLE multi_db_single_tbl;

• 查看数据库中的所有表:

SHOW TABLES;

增加列、删除列、修改列的数据类型

● 增加列:

ALTER TABLE multi_db_single_tbl ADD COLUMN textcol text;

• 修改列的数据类型:

ALTER TABLE multi_db_single_tbl MODIFY COLUMN textcol varchar(40);

删除列:

ALTER TABLE multi db single tbl DROP COLUMN textcol;

创建、查看和删除本地索引

• 您可以使用如下两种语句中的任意一种创建本地索引:

CREATE INDEX idx_name ON multi_db_single_tbl (name);

ALTER TABLE multi_db_single_tbl ADD INDEX idx_name(name);

• 查看本地索引:

SHOW INDEX FROM multi db single tbl;

您可以使用如下两种语句中的任意一种删除本地索引:

DROP INDEX idx_name ON multi_db_single_tbl;

ALTER TABLE multi db single tbl DROP INDEX idx name;

增删改查表数据

• 插入表数据:

```
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');
```

• 查询表数据:

SELECT * FROM multi db single tbl;

• 修改表数据:

UPDATE multi db single tbl set name='zzz' WHERE id in (100001,100002,100003,100004);

• 删除表数据:

DELETE FROM multi db single tbl WHERE id = 100002;