ALIBABA CLOUD

阿里云

可信账本数据库
用户指南

文档版本：20220706

阿里云

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
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。	危险 重置操作将丢失用户配置数据。
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。	注意 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。	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 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。	`switch {active|stand}`

目录

1.公测新用户使用引导

本文主要帮助新用户快速创建LedgerDB实例和完成实例访问的相关配置。

步骤

1. 完成公测申请，申请地址。

2. 创建LedgerDB实例。

   1. 首先登录控制台。

   2. 按照控制台引导，通过实例购买页面创建LedgerDB实例。

3. 配置身份公钥。

配置写入数据时用于数据验签的公钥。目前支持的公钥算法 SECPK1（ECCK1），单击此处查看参考文档。

4. 配置实例访问方式。

LedgerDB提供两种方式访问LedgerDB实例，单击此处查看参考文档。

- 配置VPC访问LedgerDB实例。此方式访问速度更快，更安全。
- 获取公网访问地址，此方式允许用户通过公网访问LedgerDB实例。

5. 写入数据。

Ledger通过POP API/SDK/Ledger Client来操作账本实例。为了保证数据安全，目前向Ledger实例写入数据，仅可通过Ledger Client内提供的 appendTransaction API。

- 参考文档
- SDK下载地址

2.身份公钥设置

本文介绍在可信账本数据库LedgerDB内进行身份公钥设置。

概述

对于LedgerDB中的每一个Ledger实例，公钥是每个成员的唯一身份标示，用于执行数据写入、数据读取、账本成员管理等操作。

LedgerDB目前支持两种方式的身份密钥管理：

- KMS管理（推荐）
- 本地管理

重要提醒

可信账本数据库LedgerDB不接触用户私钥，更不会保存用户私钥，请自行妥善保管私钥。安全起见，公钥更新后历史公钥将永久失效。

KMS管理

LedgerDB已经接入了阿里云密钥管理服务KMS。用户可在创建LedgerDB实例过程中，通过开通/授权LedgerDB服务访问您的KMS服务，来指定用于LedgerDB实例的用户身份密钥。

KMS内EC_P256K和EC_SM2两种类型的密钥可用于LedgerDB。在LedgerDB内使用KMS服务所产生的费用，单独由KMS收取。

使用流程

1. 开通KMS服务（如果已开通，请忽略）。
2. 授权LedgerDB访问您的KMS服务。
3. 创建/选择密钥。
4. 完成。

本地管理

1. 登录LedgerDB控制台。

2. 在**实例详情**页面内的**身份公钥**标签页内，进行公钥的首次上传和更新。目前支持的公钥算法为SECPK1(ECCK1)。 密钥对生成算法可参考 ECCK1KeyPair 密钥对生成。

重要提醒

- 可信账本数据库LedgerDB不接触用户私钥，更不会保存用户私钥，请自行妥善保管私钥。
- 安全起见，公钥更新后历史公钥将永久失效。

3.访问设置

本文介绍在可信账本数据库LedgerDB内如何进行Ledger实例的VPC访问和公网访问设置。

VPC访问设置

专有网络（VPC）：一个VPC就是一个隔离的网络环境。VPC的安全性较高，推荐您使用VPC网络访问Ledger实例。什么是VPC?

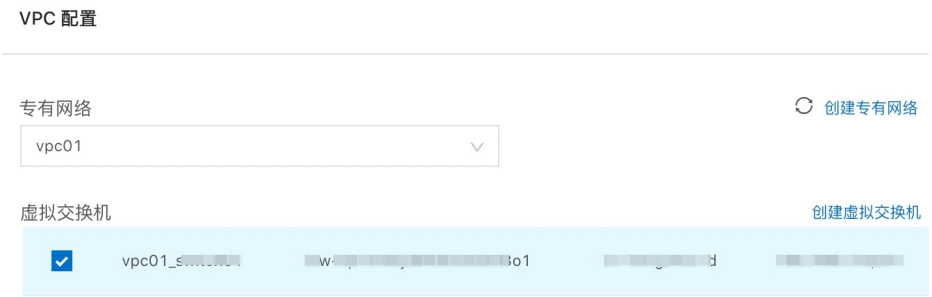
操作步骤

1. 登录LedgerDB控制台。
2. 在**实例详情**页面内的**访问设置**标签页内。点击“VPC Endpoint”旁边的“点击设置”按钮。

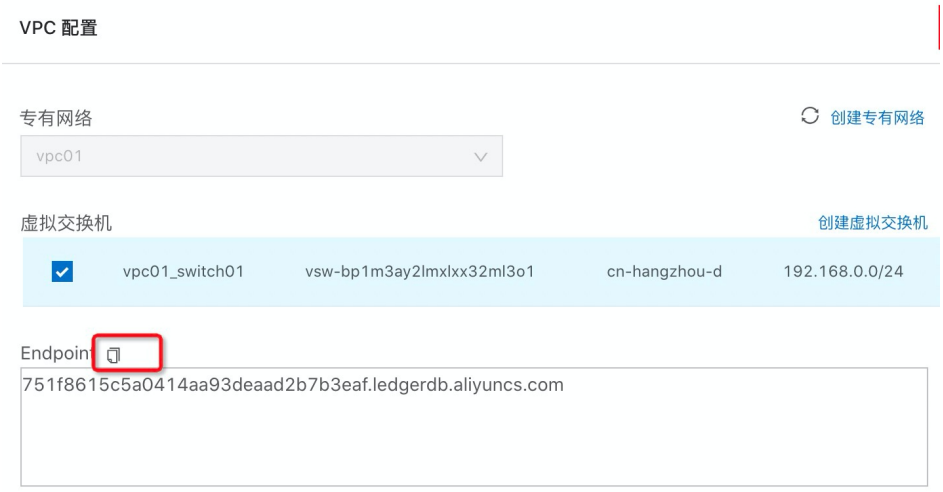
3. 创建/选择专有网络（VPC）。

- 从未在阿里云创建过VPC的用户，可通过**创建专有网络**按钮跳转至VPC控制台进行专有网络的创建。
- 从未在阿里云创建过虚拟交换机的用户，可通过**创建虚拟交换机**按钮跳转至VPC控制台进行虚拟交换机的创建。

4. 在列表中选择已有的VPC和VPC下的虚拟交换机，并提交。



5. 复制系统生成的Endpoint访问链接，用于您的业务系统通过VPC的方式访问具体的Ledger实例。



公网访问设置

操作步骤

1. 登录LedgerDB控制台。
2. 在**实例详情**页面内的**访问设置**标签页内。点击“公网地址”旁边的“点击获取”按钮。

3. 复制系统返回的Ledger实例的公网访问地址，用于您的业务系统通过公网访问具体的Ledger实例。

访问白名单设置

只有在白名单内的IP地址才可以通过Ledger实例的公网地址访问Ledger实例。仅Ledger实例的创建者可配置白名单列表。

1. 登录LedgerDB控制台。
2. 在**实例详情**页面内的**访问设置**标签页内，进行白名单设置。

> **注意**
>
> 支持IPv4地址段。填写CIDR网段格式，在IP地址后加上斜杠（/）以及1~32的掩码范围，其中1~32表示子网掩码中网络标识位的长度。例如，192.168.0.3/24。关于CIDR格式介绍，请参见网络FAQ。如果填写0.0.0.0/0表示允许所有IP地址的访问，设置时请务必谨慎。

4.成员管理

本文介绍在可信账本数据库LedgerDB内如何管理账本成员。

- 什么是账本成员？请参见功能介绍
- 只有账本管理员可对账本成员进行管理
- 账本的创建者默认为账本的管理员

功能入口

1. 登录LedgerDB控制台。
2. 在实例详情页面内的基本详情标签页内。

邀请成员

- 可一次性输入多个阿里云UID
- 邀请通知会以阿里云站内信的形式发送给被邀请人，请被邀请人查看阿里云通知
- 被邀请人点击通知内邀请链接来接受邀请

权限管理

目前的权限列表：

- 管理员，拥有对Ledger的所有操作权限
- 写入，可向LedgerDB写入读取数据
- 只读，可读取LedgerDB数据

禁用和启用

- 禁用：成员无法再访问对应Ledger
- 启用：恢复成员对应Ledger的访问权限

移除

将指定成员移除对应账本。可通过**邀请成员**的方式，再次邀请其加入。

5.时间账本

本文介绍在可信账本数据库LedgerDB内的时间账本功能。

什么是时间账本？

时间证据编号查询

时间证据编号是时间锚点（TimeAnchor）和TSA在时间账本内的记录编号。用户可通过自己某个账本内的时间锚点对应的时间证据编号来这个公共的时间账本内查询。

TSA验证

TSA验证，指的是在可信第三方网站，验证LedgerDB内记录的时间信息的可信性。用户可在TSA详情页中，通过页面提供的验证功能，跳转至第三方验证网站进行时间戳验证。验证所需数据，均已在TSA详情页提供。

6.数据可信性的验证

本文主要介绍LedgerDB内的数据可信性验证。同时提供可信性验证的方法，用户可根据业务需求，自行对LedgerDB内的数据进行可信性验证。

什么是数据可信性？

指写入Ledger实例的数据，具有不可篡改和不可抵赖性，同时可追溯和溯源，并且支持可信性的验证。

- 不可篡改：任何数据写入Ledger实例后，任何人都无能力修改和删除，包括LedgerDB服务提供方。
- 数据不可抵赖：任何数据的写入动作，都需要数据的写入方使用自己的私钥为待写入的数据进行签名，配合“不可篡改”能力，任何写入Ledger实例的数据都不能被抵赖。
- 数据溯源（追溯）：由于LedgerDB在数据记录时采用类似Journal模式，所有数据的非读操作都会被写入对应的Ledger实例中，因此LedgerDB在数据库层面提供原生的数据溯源能力。

什么是数据可信性验证？

数据可信性验证，是指通过密码学算法来验证存入LedgerDB内的数据是否被篡改。

控制台可信性验证

LedgerDB底层基于新型的默克尔累加器进行研发，所以在数据可信性验证时，会涉及到默克尔树的相关概念。

用户可对指定账本的指定数据进行可信性验证。

JSON字段解释

result	验证结果。成功/失败（成功表示数据可信性验证通过）
MemberId	被验证数据的写入方ID
LedgerId	被验证数据所属LedgerID
JournalSequence	被验证数据在所属Ledger中的记录编号
RootHash	被验证数据的父节点哈希值
WriterPubKey	被验证数据的写入方公钥
ProofPath	进行可信性验证的辅助数据。用户可利用此数据和RootHash，通过本文提供的验证方法自行进行验证
Timestamp	被验证数据的写入时间戳

数据可信性验证方法

用户可根据业务需求，通过此方法自行验证数据的可信性。

```
import com.alibaba.fastjson.JSON;
import com.alibaba.fastjson.JSONArray;
import com.google.common.hash.Hashing;
import org.bouncycastle.util.Arrays;
import org.bouncycastle.util.encoders.Hex;
import org.junit.Assert;
import org.junit.Test;
import org.springframework.util.StringUtils;
public class ProofDemo {
  @Test
  public void test() {
    // 来自于从LedgerDB控制台>数据可信性验证>JSON复制功能。同时也可以自行拼装
    // 对应ProofPath字段
    String proofPath = "******";
    // 来自于从LedgerDB控制台>数据可信性验证>JSON复制功能
    // 对应RootHash字段
    String rootHash = "来自于从LedgerDB控制台>数据可信性验证>JSON复制功能";
    Assert.assertTrue(verifyProofPathV1(rootHash, proofPath));
  }
  public static byte[] calculateRoot(JSONArray proofPath) {
    byte[][] childHashes = new byte[proofPath.size()][];
    for (int i = 0; i < proofPath.size(); i++) {
      Object child = proofPath.get(i);
      byte[] childHash = null;
      if (child instanceof String) {
        // 叶子节点
        childHash = Hex.decode((String) child);
      } else {
        // 分支节点
        childHash = calculateRoot(proofPath.getJSONArray(i));
      }
      childHashes[i] = childHash;
    }
    // 父节点hash为子节点的hash的拼接作为输入计算而得
    byte[] input = Arrays.concatenate(childHashes);
    return Hashing.sha256().hashBytes(input).asBytes();
  }
  public static boolean verifyProofPathV1(String rootHash, String proofPath) {
    JSONArray path = JSON.parseArray(proofPath);
    if (StringUtils.isEmpty(rootHash)) {
      if (path.size() == 0) {
        return true;
      } else {
        return false;
      }
    }
    if (path.size() == 1) {
      return rootHash.equals(path.getString(0));
    }
    byte[] rootCalculated = calculateRoot(path);
    return rootHash.equalsIgnoreCase(Hex.toHexString(rootCalculated));
  }
}
```