ALIBABA CLOUD

Alibaba Cloud

LedgerDB

User Guide

Document Version: 20220425

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.The public beta user guide

This article introduces users to quickly create and access LedgerDB instances.

Step

1. Complete the application for public preview. Application address.

2. Create a LedgerDB instance.

- Login LedgerDB console.
- Follow the instructions in the console to create a LedgerDB instance on the instance purchase page.

3. Configure the identity public key.

The public key used for data verification. Currently, the public key algorithm SECPK1(ECCK1) is supported. Document.

4. Configure the instance access method.

LedgerDB provides two ways to access LedgerDB instances. Document.

- Configure a VPC instance to access LedgerDB. This access method is faster and safer.
- Get the public network access address, this method allows the user to access LedgerDB instance through the public network

5. Write data.

Users use the POP API, SDK, or Ledger Client to manage the LedgerDB. To ensure data security, only the appendTransaction API.

- Document
- SDK download

2.Identity public key settings

This article introduces how to set the public identity key in LedgerDB.

Overview

For each Ledger instance in LedgerDB, the public key is the unique identification of each member, and is used to perform operations such as data writing, data reading, and Ledger member management.

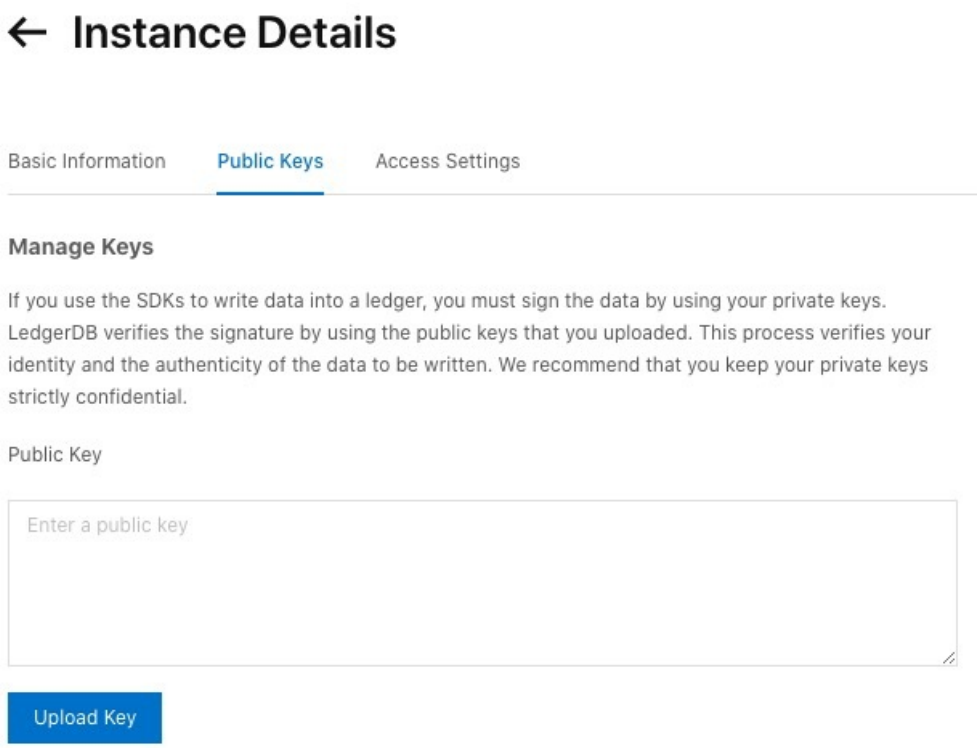
Important note

- The LedgerDB does not know user private keys and does not save user private keys. You can manually keep the private keys confidential.
- For security reasons, the historical public key will be permanently invalid after the new public key is updated

Upload and update the public key

1. Login LedgerDB console

2. In **instance details** the in the page **identity public key** tab Page, perform the first upload and update of the public key. The currently supported public key algorithm is SECPK1(ECCK1). For key pair generation algorithms, see ECCK1KeyPair key pair generation.



3.Access settings

This article introduces how to configure VPC access and public network access for Ledger instances in LedgerDB.

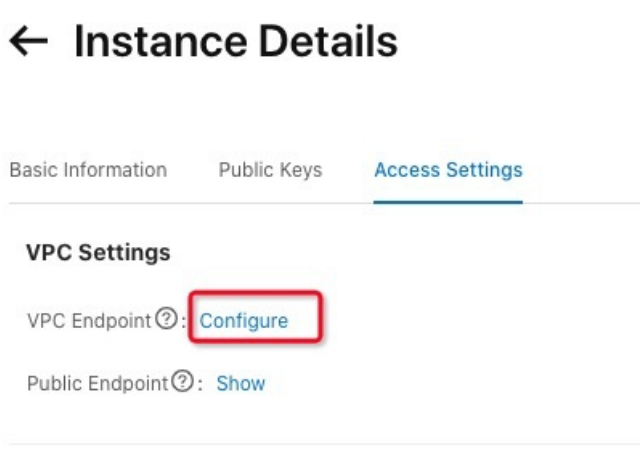
VPC access settings

VPC: Each Virtual Private Cloud (VPC) is an isolated network. We recommend that you use VPC to access Ledger instances because VPC is more secure. What is a VPC?

Procedure

1. Login LedgerDB console.

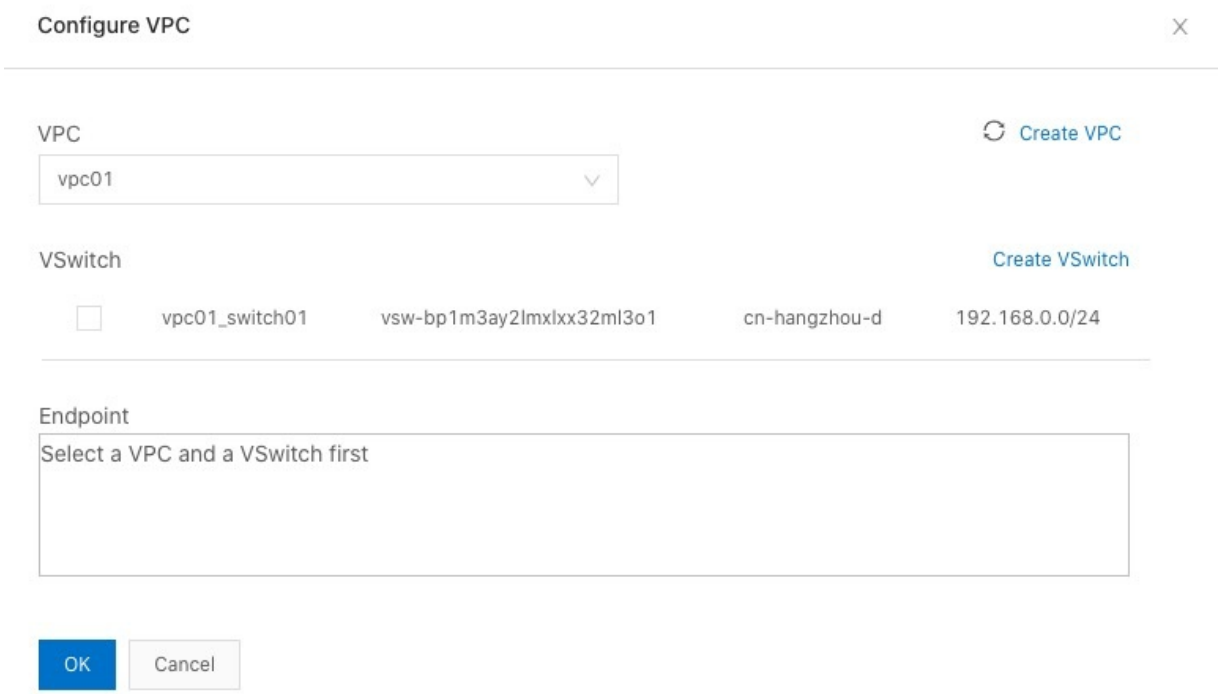
2. In **instance details** page **access settings** inside the tab. Click the click Configure button next to VPC Endpoint.



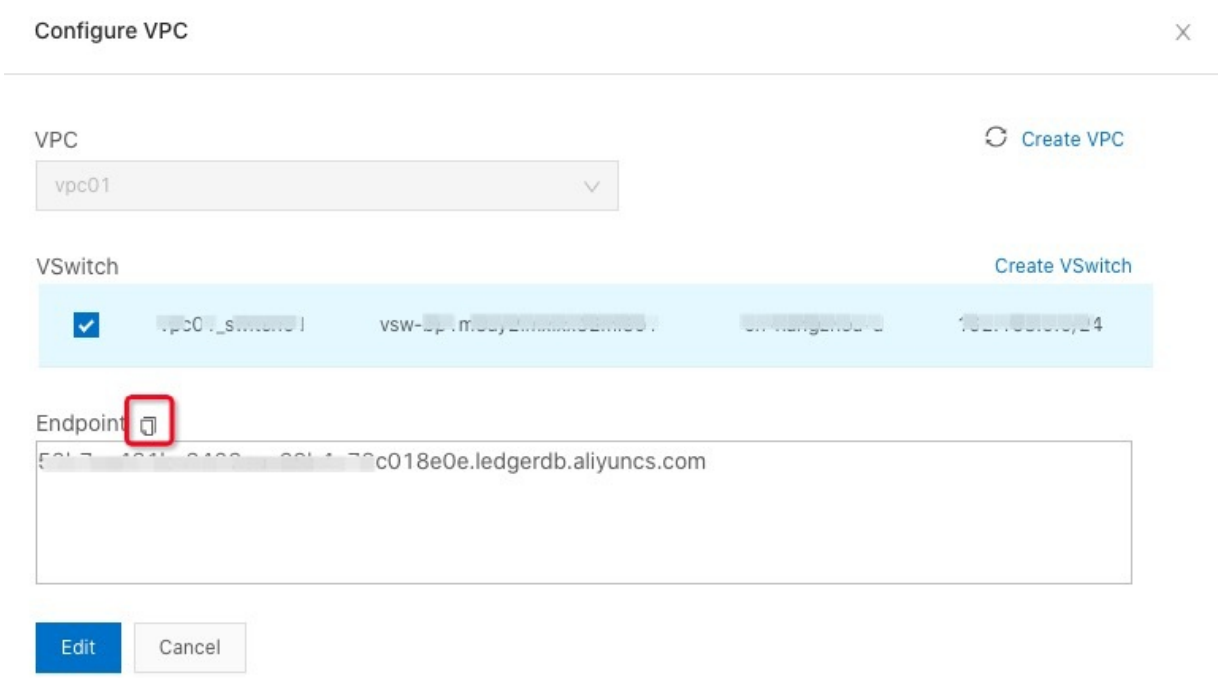
3. Create /Select a VPC

- If you have never created a VPC on Alibaba Cloud, use the **"Create a VPC"** button to go to the VPC console to create a VPC.
- If you have not created a vSwitch in the Alibaba Cloud Console, click **"Create vSwitch"** and create a vSwitch in the VPC console.

4. Select an existing VPC and a vSwitch under the VPC from the drop-down list and click submit.



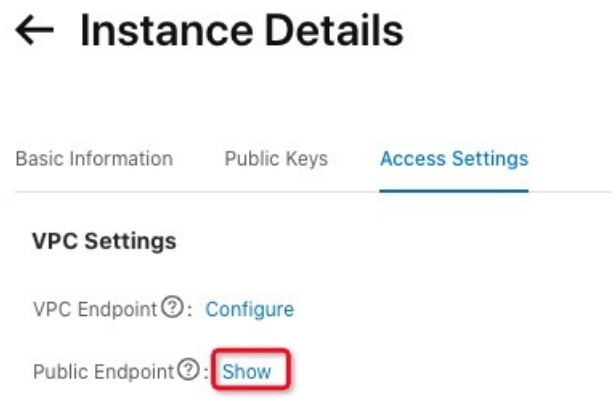
5. Copy the Endpoint generated by LedgerDB for you to access a specific Ledger instance in a VPC.



Public network access settings

Procedure

1. Login LedgerDB console
2. In **instance details** page **access settings** inside the tab. Next to public endpoint, click **show**.



3. Copy the public IP address of the Ledger instance that is returned by the system for you to access a specific Ledger instance through the public network.

Whitelist settings

Only IP addresses that are added to the whitelist can access the LedgerDB instance through the public endpoint.

Only the creator of a LedgerDB instance can configure the whitelist:

1. Login LedgerDB console

2. In **instance details** page **access settings** inside the tab. Configure a whitelist.

> **Notice**
>
> IPv4 CIDR blocks are supported. Enter a CIDR block, and add a forward slash (/) and a mask ranging from 1 to 32. The mask indicates the length of the network identification bit in the subnet mask. Example: 192.168.0.3/24. For more information about the CIDR format, see Network FAQ. If 0.0.0.0/0 indicates that access from all IP addresses is allowed, proceed with caution.

4.Manage members

This article introduces how to manage the members of the ledger in LedgerDB.

- What is a ledger member? See features.
- Only the ledger administrator can manage the ledger members
- By default, the creator of a ledger is the administrator of the ledger.

Entry

1. Login LedgerDB console
2. On the instance details page. Click "Manage" link beside "Ledger Members" label.

Invite member accounts

- You can enter multiple Alibaba Cloud UIDs at a time.
- The invitation notification will be sent to each invitee by sending an Alibaba Cloud internal message. Check the Alibaba Cloud notification
- The invitee clicks the invitation link in the notification to accept the invitation.

Manage permissions

Current permissions:

- Administrator, with all the operation permissions on Ledger
- Write, write and read data to LedgerDB
- Read-only, able to read LedgerDB data

Disable and enable

- Disabled: the member can no longer access the corresponding Ledger.
- Enabled: restores the access permissions of a Ledger that is associated with a member.

Remove

Removes the specified member from the corresponding ledger. Can be passed **invite members** the way to invite it again.

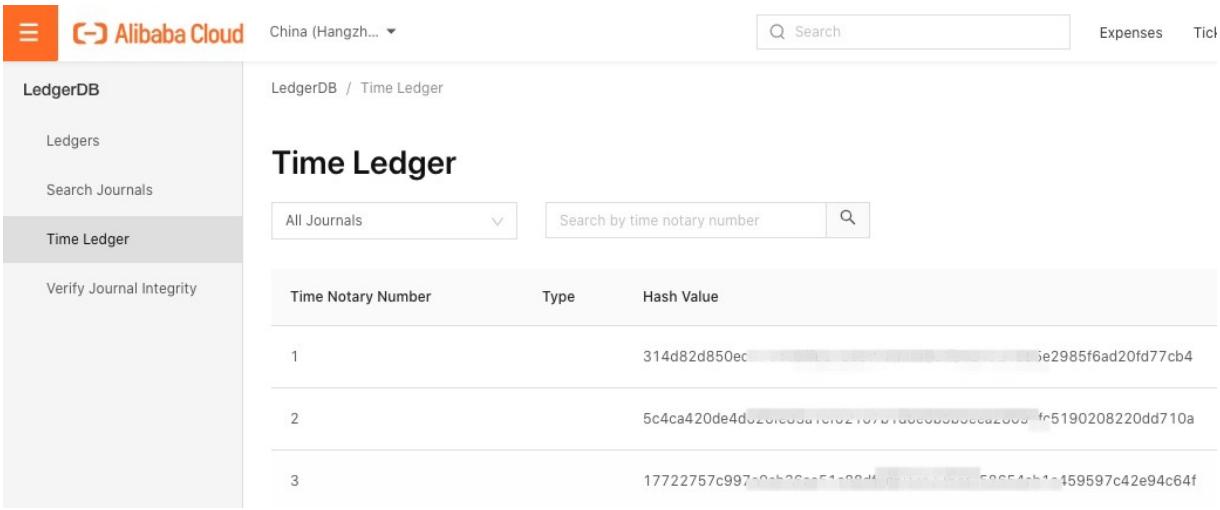
5.Time Ledger

This document introduces the Time Ledger in LedgerDB.

What is Time Ledger?

Time evidence number query

The time evidence number is the record number of the time anchor and the TSA in the Time Ledger. Users can query the time evidence number corresponding to the time anchor in a certain Ledger to the Time Ledger.



TSA validation

TSA verification refers to verifying the credibility of the time information recorded in LedgerDB on a trusted third-party website. You can go to the third-party verification website and perform the timestamp verification on the TSA details page by clicking the verification function. The data required for verification is provided on the TSA details page.

6.Verify data credibility

This article introduces data trustworthiness in LedgerDB. At the same time provide a method for credibility verification, users can perform credibility verification on data in LedgerDB according to business needs.

What is Data credibility?

Refers to the data written to ledger instances. The data is tamper-proof, non-repudiation-proof, traceable, and supported for credibility verification.

- Tamper-proof: no one can modify or delete the data after it is written to the Ledger instance, including the LedgerDB service provider.
- Non-repudiation of data: when any data is written, the data writer must sign the data using its own private key. With the "tamper-resistant" capability, no data written to ledger instances can be denied.
- Data tracing: LedgerDB uses the journal mode for data recording, so all non-read operations of the data are written to the corresponding ledger instance. Therefore, LedgerDB provides native data tracing capabilities at the database level.

What is data trustworthiness verification?

Data credibility verification is a cryptographic algorithm used to verify whether the data stored in LedgerDB has been tampered with.

Verify credibility in the console

LedgerDB is developed based on the new merkel accumulator. When verifying the trustworthiness of data, we need to involve the concepts related to the Merkel tree.

Allows you to verify the credibility of specified data in a specified ledger.

JSON field description

result	Success /Failure (Success indicates that the data credibility verification is passed)
MemberId	The writer ID of the data to be verified.
LedgerId	Ledger instance id to which the verified data belongs
JournalSequence	The record number of the ledger to which the data to be verified belongs.
RootHash	The parent node hash value of the validated data
WriterPubKey	The public key of the writer.
ProofPath	Auxiliary data for credibility verification. You can use the data and RootHash to perform self-verification through the verification method provided in this article
Timestamp	The timestamp when the data is written.

Verify credibility of data

This method is used to verify the credibility of data based on business needs.

```
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() {
    // From the LedgerDB console> data trustworthiness verification> JSON replication. At t
he same time, it can also be assembled by itself.
    // Corresponding profpath field
    String proofPath = "******";
    // From the LedgerDB console> data reliability verification> JSON replication function
    // Corresponds to the RootHash field.
    String rootHash = "from the LedgerDB console> data trustworthiness verification> JSON c
opy function";
    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) {
        // Leaf node
        childHash = Hex.decode((String) child);
      } else {
        // Branch node
        childHash = calculateRoot(proofPath.getJSONArray(i));
      }
      childHashes[i] = childHash;
    }
    // The hash of the parent node is a child node, and the concatenation of the hash is ca
lculated as the input.
    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));
    }
  }
```