Alibaba Cloud Table Store

Authorization

Issue: 20181115

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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, 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 content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, 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 the Alibaba Cloud website, product programs, or content 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 contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information 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.
	This warning information 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 restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 **Note:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
`Italics`	It is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *`[-all|-t]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 RAM and STS

Alibaba Cloud's permission management function include Resource Access Management (RAM) and Security Token Service (STS). This function enable users to access Table Store through RAM user accounts with different permissions, and grant users temporary access authorization.

RAM is primarily used to control account system permissions over a long-term period. It allows you to assign different permissions to different RAM users created under your primary account to implement authorization management. For more information, see *RAM*.

STS is a security credential (token) management system that grants temporary access permissions.

Background

RAM and STS are designed to securely grant access to users without disclosing the primary account's AccessKey. Unintentional AccessKey disclosure poses serious account security risks as unauthorized users may freely operate the affected primary account, including malicious use of resources and theft of account information.

RAM provides permission control function used to allocate RAM users with different permissions to different entities, minimizing impact to a primary account if a RAM user's AccessKey is disclosed. Generally, RAM users are created for long-term account operations. Therefore, the AccessKeys of RAM users must not be disclosed.

In contrast to RAM's long-term control function, STS provides temporary access authorization by returning a temporary AccessKey and token, which can be used directly by temporary users to access Table Store. Generally, the permissions obtained from STS are more restrictive and only valid for a limited period of time.

Basic concepts

Basic concepts related to RAM and STS are described as follows:

Concept	Description
RAM user	RAM users are created under an Alibaba Cloud primary account and assigned independent passwords and permissions, with each RAM user having its own AccessKey. RAM users can be used to perform authorized operations in the same way as the primary account. Generally, RAM users can be understood as

Concept	Description
	users with certain permissions or operators with permissions for specified operations.
Role	A virtual concept indicating certain operation permissions, roles do not have independent logon passwords or AccessKeys. RAM users can assume roles, and the permissions that are granted when a role is assumed belong to this role. A role may be assumed by multiple users at the same time.
Policy	Policies are rules used to define permissions, such as the permissions to read or write certain resources.
Resource	Resources are the cloud resources that users can access, such as one or all instances of Table Store, or a certain table in an instance.

The relationship between a RAM user and its roles is similar to a relationship between an individual and their social identities in different scenarios. For example, a person can assume a role of employee in a company and assume a role of parent at home. Different roles are assigned corresponding permissions. The concept of employee or parent is not an actual entity able to take actions. Roles are complete only when being assumed by RAM users. Furthermore, a role may be assumed by multiple users at the same time. The user who assumes a role is automatically assigned all permissions of the role.

The following example provides more detailed information:

Assume that an Alibaba Cloud primary account named Alice has two Table Store instances named alice_a and alice_b. Alice has full permissions on both instances.

To maintain security of the primary account, Alice uses RAM to create two RAM users: Bob and Carol. Bob has the read and write permissions on alice_a, and Carol has the read and write permissions on alice_b. Bob and Carol both have their own AccessKeys. If the AccessKey of Bob or Carol is disclosed, only the corresponding instance is affected. Alice can then cancel the permissions of the compromised RAM user through the console.

If Alice needs to authorize another RAM user to read the tables in alice_a, instead of disclosing Bob’s AccessKey to the user, Alice can create a role (for example, AliceAReader), and assign

the role the read permission on alice_a. However, AliceAReader cannot be used directly as no AccessKey corresponds to this role.

To obtain temporary authorization, Alice can call STS's AssumeRole interface to inform STS that RAM user Bob wants to assume the role AliceAReader. If the interface is successfully called, STS returns a temporary AccessKeyID, AccessKeySecret, and SecurityToken as the access credentials. A temporary user assigned with these credentials obtains the temporary permission to access alice_a. The credentials' expiration time is specified when the AssumeRole interface is called.

RAM and STS best practices

RAM and STS are designed with complexity to achieve flexible permission control at the cost of simplicity.

RAM users and roles are two concepts used to separate the entity that performs operations from the virtual entity that represents a permission set. If a RAM user requires many permissions (including read and write permissions) but each operation only requires part of the total permission set, you can create two roles: one with the read permission and one with the write permission. Then create a user who does not have any permissions but can assume these two roles. When a RAM user needs to read or write data, the RAM user can temporarily assume the role with the required permission. In addition, roles can be used to grant permissions to other Alibaba Cloud users, making collaborations easier while maintaining strict account security.

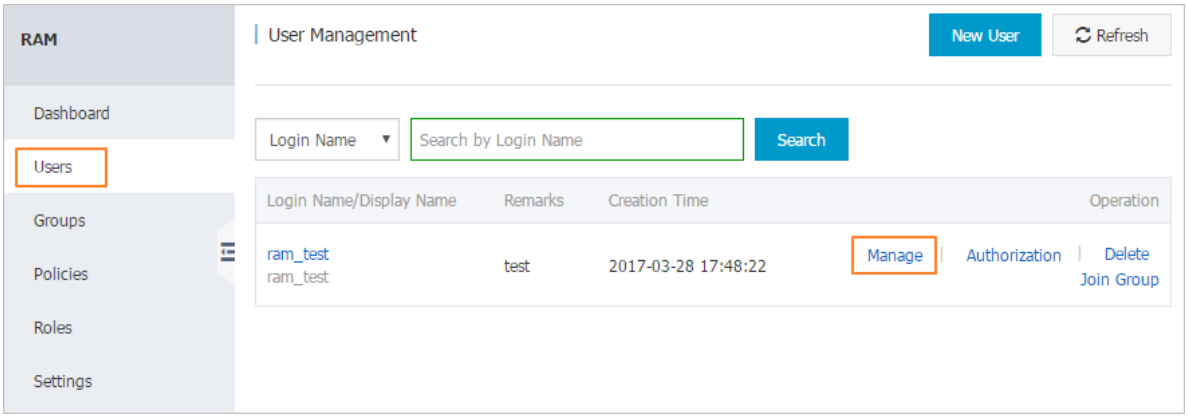
Implementing RAM or STS through the console and command line operations are strongly recommended to reduce the actual amount of codes that must be used. If code must be used to perform such operations, see the *RAM API Reference* and *STS API Reference*.

2 RAM user access

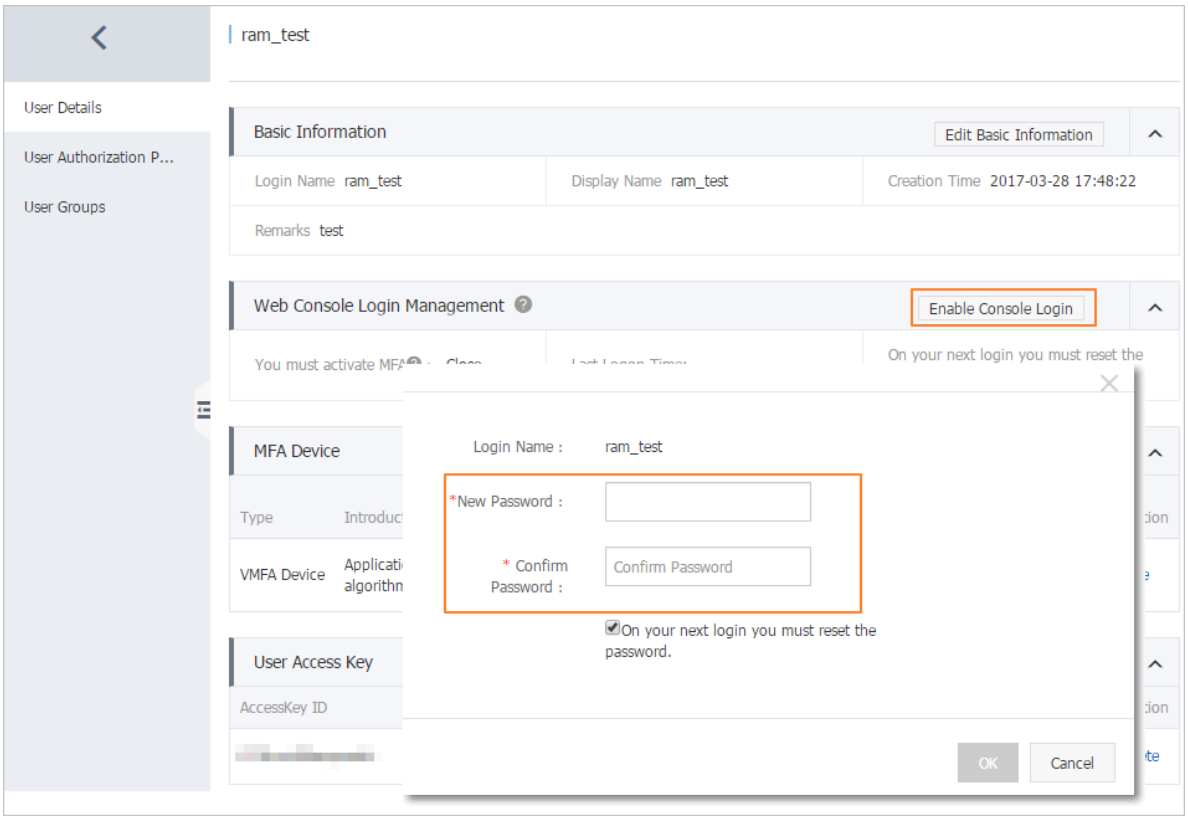
To maintain strict account security, use a RAM user to log on to the Table Store console for instance and table management. A RAM user can also be used to call the SDK interface and access data APIs.

Log on to the console

1. Log on to the *RAM console*.
2. Click **Users**.



3. Select the required RAM user (this example uses ram_test), and click **Manage**.
4. Click **Enable Console Logon** and reset the password.



5. Return to the User Management page and click **Dashboard**.
6. Click the **RAM User Login Link**.
7. Enter the RAM user's user name and password set in step 4 to log on to the console.
8. Click **Table Store** to access the console..

Enable MFA

MFA (Multi-Factor Authentication) is a simple, effective method that provides additional security protection. When MFA is enabled, a 2-step security verification process is required through use of both a username and password to log on to the Alibaba Cloud console (first security factor), and a variable verification code provided by the MFA device (second security factor).

1. Use the primary account to log on to the *RAM console*.
2. In the left-side navigation pane, click **Users**.
3. Locate the required RAM user, and click **Manage**.
4. On the **User Details** page, click **Enable VMFA Device** and follow the guide to enable the VMFA device.
5. Enable MFA.

After MFA is enabled, a MFA device verification code is required when logging on to the console.

Call an API

Create an AccessKey for the required RAM user and pass in this AccessKey when calling the SDK interface. The RAM user can then operate the same as the primary account.

3 STS temporary access authorization

This topic describes how to use STS to grant users temporary permissions for accessing Table Store to better restrict access permissions for short-term RAM user requirements.

Assume that a developer's app is used by multiple users, and each user is allowed to write data to instance ram_test_app. Data upload permissions must therefore be granted to these users, and the data stored by multiple users need to be separated.

To meet these requirements, use STS to grant users temporary access permission. The process is as follows.

Create roles

1. Create a new RAM user named ram_test_app and do not grant it permissions. For RAM user creation details, see step 1 to step 7 in Use case.
2. Create two roles, RamTestAppReadOnly and RamTestAppWrite, to perform read operations and to upload files respectively.
   a. Log on to the *RAM console*.
   b. Select **Roles** > **Create Role**.
   c. Select **User Role**.
   d. Enter the role information. Retain the default parameters and click **Next**.
   e. Enter the role name RamTestAppReadOnly and click **Create**.
   f. Click **Close**.
3. When created, the role does not have any permissions. To enable permissions, create a custom authorization policy.
   a. Select **Policies** > **Create Authorization Policy**.
   b. Select **Blank Template**.
   c. Enter the **Authorization Policy Name**. For example: ram-test-app-readonly, and enter Policy Content as follows.

```
{
"Statement": [
 {
   "Effect": "Allow",
   "Action": [
  "ots:BatchGet*",
  "ots:Describe*",
  "ots:Get*",
  "ots:List*"
   ],
```

```
   "Resource": [
  "acs:ots:*:*:instance/ram-test-app",
  "acs:ots:*:*:instance/ram-test-app/table/*"
   ]
 }
],
"Version": "1"
}
```

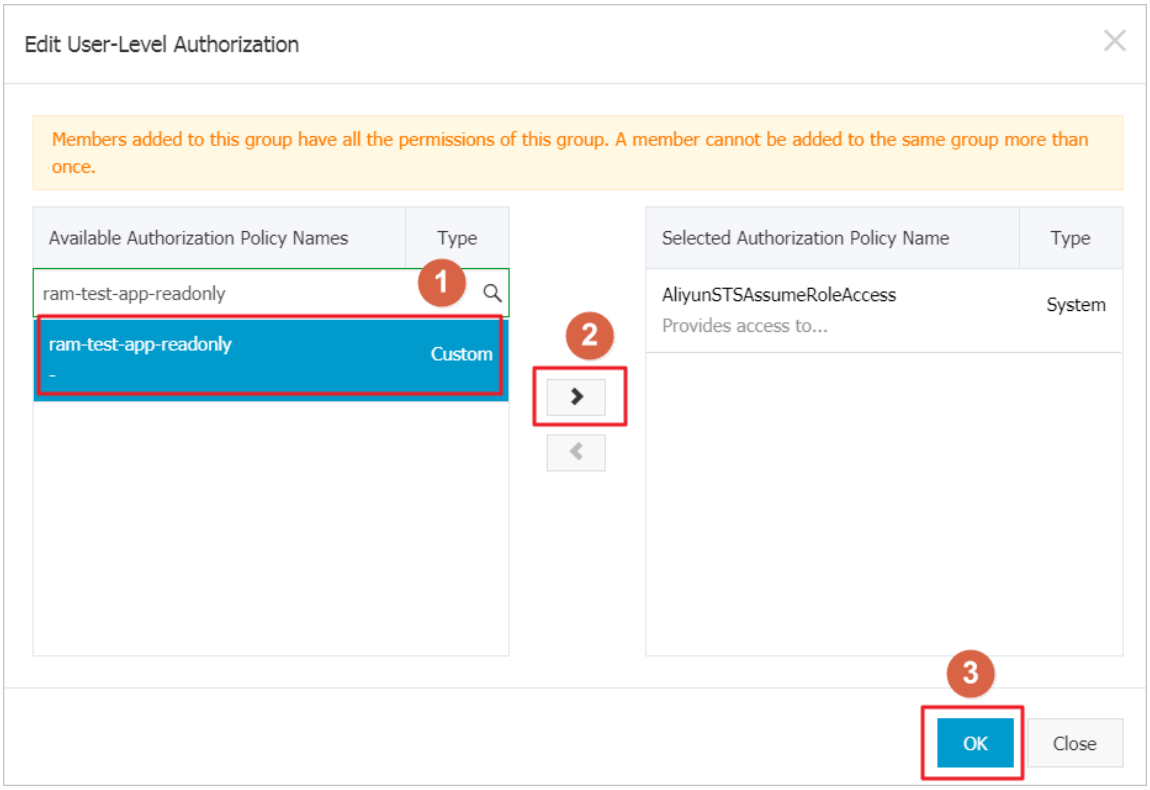
In the policy, the role is granted the read-only permission for ram-test-app.

d. Click **Create Authorization Policy** and then click **Close**.

4. Grant the role RamTestAppReadOnly the ram-test-app read-only permission.

a. Click **Roles** and click **Authorize** on the right side of role RamTestAppReadOnly.

b. Select ram-test-app-readonly and click  .



c. Click **OK**.

Role RamTestAppReadOnly has been granted the read-only permission for ram-test-app.

5. Follow the preceding steps to create another role called RamTestAppWrite, and grant this role the write permission for ram-test-app. The authorization policy is as follows.

```
{
"Statement": [
 {
   "Effect": "Allow",
   "Action": [
```

```
   "ots:Create*",
   "ots:Insert*",
   "ots:Put*",
   "ots:Update*",
   "ots:Delete*",
   "ots:BatchWrite*"
    ],
    "Resource": [
   "acs:ots:*:*:instance/ram-test-app",
   "acs:ots:*:*:instance/ram-test-app/table/*"
    ]
  }
],
"Version": "1"
}
```

Temporary access authorization

- Preparations

  Authorization is required for assuming roles, otherwise, any RAM users could assume these roles. To assume corresponding roles, a RAM user needs to have designated permissions explicitly configured. To create two policies and grant them to a RAM user (in this example, ram_test_app), follow these steps:

  1. In the left-side navigation pane, click **Policies**.
  2. Click **Create Authorization Policy**.
  3. Select **Blank Template**.
  4. Enter the **Authorization Policy Name** and **Policy Content** as follows.

     - AliyunSTSAssumeRolePolicy2016011401

     ```
     {
     "Version": "1",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": "sts:AssumeRole",
             "Resource": "acs:ram:1983407596944237:role/ramtestapp
     readonly"
         }
     ]
     }
     ```

     - AliyunSTSAssumeRolePolicy2016011402

     ```
     {
     "Version": "1",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": "sts:AssumeRole",
             "Resource": "acs:ram:1983407596944237:role/ramtestappwrite
     "
     ```

```
    }
]
}
```

Note:
The content entered after Resource is a role ID. To find the role ID, click **Manage** on the **Roles** page.

5. Click **Users**.
6. Find `AliyunSTSAssumeRolePolicy2016011401` and `AliyunSTSAssumeRolePolicy2016011402`, and click to add them.

- Use STS

  Download STS command line tool Python.

  For more information about the call method and parameter description, see *STS Help Documentation*.

```
$python ./sts.py AssumeRole RoleArn=acs:ram::1983407596944237:role
/ramtestappreadonly RoleSessionName=usr001 Policy='{"Version":"1
","Statement":[{"Effect":"Allow","Action":["ots:ListTable","ots:
DescribeTable"],"Resource":["acs:ots:*:*:ram-test-app","acs:ots:*:*:
ram-test-app/*"]}]}' DurationSeconds=1000 --id=id --secret=secret
```

  The parameters are described as follows:

Parameter	Description
RoleArn	The ID of the role to be assumed. A Role ID can be queried in **Roles** > **Manage**.
RoleSessionName	The name of a temporary credential. Generally, we suggest separating RoleSessionNames by different application users.
Policy	An additional permission restriction added when the role is assumed. **Note:** Policy is used to restrict the temporary credential permissions after a role is assumed. Temporary credential permissions are the overlapping permissions of the role and the policy passed in. When a role is assumed, a policy can be passed in

Parameter	Description
	to improve flexibility. For example, when uploading files, you can add different uploading path restrictions for different users.
DurationSeconds	The validity period of the temporary credentials, measured in seconds. The minimum value is 900, and the maximum value is 3600.
id and secret	The AccessKey of the role to be assumed.

- Test STS functions

  Create a table named test_write_read on the *console*. Set the primary key to name and the type to string. Use the Table Store CLI tool to test read and write operations.

  In this example, the RAM user ram_test_app is used to access Table Store. In actual scenarios, replace the following AccessKey with your own AccessKey.

```
python2.7 ots_console --url https://TableStoreTest.cn-hangzhou.
ots.aliyuncs.com --id 6iTlV1uhiY71mlRt --key clkkuDiq69IJWJ7PnA9P
XJxhRWMr3P
You cannot access the instance!
ErrorCode: OTSNoPermissionAccess
ErrorMessage: You have no permission to access the requested
resource, please contact the resource owner.
```

  Without access permission, access attempts using the RAM user ram_test_app fail.

- Use temporary authorization to write data

  In this example, the policy passed in is the same as the role policy. The expiration time is set to 3,600s, and SessionName is assumed as session001.

  1. Use STS to obtain temporary credentials.

```
python2.7 ./sts.py AssumeRole RoleArn=acs:ram::1983407596944237
:role/ramtestappwrite RoleSessionName=session001 Policy='{"
Statement": [{"Effect": "Allow","Action": ["ots:Create*","ots:
BatchWrite*","ots:Put*","ots:Insert*","ots:Update*","ots:Delete
*"],"Resource": ["acs:ots:*:*:instance/ram-test-app","acs:ots:*:*:
instance/ram-test-app/table/*"]}],"Version": "1"}' --id=6iTlV1uhiY
71mlRt --secret=clkkuDiq69IJWJ7PnA9PXJxhRWMr3P
{
"AssumedRoleUser": {
    "Arn": "acs:ram::1983407596944237:role/ramtestappwrite/
session001",
    "AssumedRoleId": "330629052749595885:session001"
},
"Credentials": {
    "AccessKeyId": "STS.x4gG7KMsfHckQe8nPKLO",
```

```
    "AccessKeySecret": "IA6CJh5kE5J5m8mR6aQXWbMemSL63Xh7SIhrEcke
",
    "Expiration": "2016-01-14T07:58:14Z",
    "SecurityToken": "CAESgAQIARKAATDsbhiBSujhVEHoMKm1i17pyZhP
TCe1BnVF5YzdNyRos4WuQjalxLkOE/hNNxg25vTo9bljKg4VCcrfh6GkJNuj
MMcJ4V1i/0RMDLfXwa0/vOHP9W/oSQpwAD5EaWJfqVY/nxwmJ0aKJDHPmSieWssn
lmocaOZAgHkpCqQSSDA8GhhTVFMueDRnRzdLTXNmSGNrUWU4blBLTE8iEjMz
MDYyOTA1Mjc0OTU5NTg4NSoGdXNyMDAxMPnCkfmjKjoGUnNhTUQ1QuIBCgEx
GtwBCgVBbGxvdxJnCgxBY3Rpb25FcXVhbHMSBkFjdGlvbhpPCgtvdHM6Q3Jl
YXRlKgoPb3RzOkJhdGNoV3JpdGUqCghvdHM6UHV0KgoLb3RzOkluc2VydCoK
C290czpVcGRhdGUqCgtvdHM6RGVsZXRlKhJqCg5SZXNvdXJjZUVxdWFscxII
UmVzb3VyY2UaTgohYWNzOm90czoqOio6aW5zdGFuY2UvcmFtLXRlc3QtYXBw
CilhY3M6b3RzOio6KjppbnN0YW5jZS9yYW0tdGVzdC1hcHAvdGFibGUvKkoQ
MTk4MzQwNzU5Njk0NDIzN1IFMjY4NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIz
MzA2MjkwNTI3NDk1OTU4ODVyD3JhbXRlc3RhcHB3cml0ZQ==="
},
"RequestId": "5F92B248-F200-40F8-A05A-C9C7D018E351"
}
```

2. Use Table Store CLI tool to write data. (The Token parameter is supported by Table Store CLI tool v1.2.)

```
 python2.7 ots_console --url https://TableStoreTest.cn-hangzhou
.ots.aliyuncs.com --id STS.x4gG7KMsfHckQe8nPKLO --key IA6CJh5kE5
J5m8mR6aQXWbMemSL63Xh7SIhrEcke --token=CAESgAQIARKAATDsbhiB
SujhVEHoMKm1i17pyZhPTCe1BnVF5YzdNyRos4WuQjalxLkOE/hNNxg25vTo
9bljKg4VCcrfh6GkJNujMMcJ4V1i/0RMDLfXwa0/vOHP9W/oSQpwAD5EaWJfqVY
/nxwmJ0aKJDHPmSieWssnlmocaOZAgHkpCqQSSDA8GhhTVFMueDRnRzdLTXNm
SGNrUWU4blBLTE8iEjMzMDYyOTA1Mjc0OTU5NTg4NSoGdXNyMDAxMPnCkfmj
KjoGUnNhTUQ1QuIBCgExGtwBCgVBbGxvdxJnCgxBY3Rpb25FcXVhbHMSBkFj
dGlvbhpPCgtvdHM6Q3JlYXRlKgoPb3RzOkJhdGNoV3JpdGUqCghvdHM6UHV0
KgoLb3RzOkluc2VydCoKC290czpVcGRhdGUqCgtvdHM6RGVsZXRlKhJqCg5S
ZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaTgohYWNzOm90czoqOio6aW5zdGFu
Y2UvcmFtLXRlc3QtYXBwCilhY3M6b3RzOio6KjppbnN0YW5jZS9yYW0tdGVz
dC1hcHAvdGFibGUvKkoQMTk4MzQwNzU5Njk0NDIzN1IFMjY4NDJaD0Fzc3Vt
ZWRSb2xlVXNlcmAAahIzMzA2MjkwNTI3NDk1OTU4ODVyD3JhbXRlc3RhcHB3cml0ZQ
==

 OTS-TableStoreTest>$ put test_write_read '001' age:integer=30
 A new row has been put in table test_write_read
```

- Use temporary authorization to read data

  In this example, the policy passed in is the same as the role policy. The expiration time is set to 3,600s, and SessionName is assumed as session002.

  1. Use STS to obtain temporary credentials.

```
python2.7 ./sts.py AssumeRole RoleArn=acs:ram::1983407596944237
:role/ramtestappreadonly RoleSessionName=session002 Policy='{"
Statement": [{"Effect": "Allow","Action": ["ots:BatchGet*","ots
:Describe*","ots:Get*","ots:List*"],"Resource": ["acs:ots:*:*:
instance/ram-test-app","acs:ots:*:*:instance/ram-test-app/table/
*"]}],"Version": "1"}' --id=6iTlV1uhiY71mlRt --secret=clkkuDiq69
IJWJ7PnA9PXJxhRWMr3P
{
"AssumedRoleUser": {
    "Arn": "acs:ram::1983407596944237:role/ramtestappreadonly/
session002",
```

```
    "AssumedRoleId": "396025752746614078:session002"
},
"Credentials": {
    "AccessKeyId": "STS. 0qJ2UE8AalcHdQ6n2Q8Q",
    "AccessKeySecret": "pSaUjb8O9mU5M76nkC6FHt6wKwbCVYO27gxSEBAu",
    "Expiration": "2016-01-14T08:14:16Z",
    "SecurityToken": "CAES6wMIARKAAVtHeNgUnhk132OwDfxZTu8gPQCx
fakYLeWha/FxoEYNqBKQTtyI4WPC5mpYuu8+n+yamSYTI2VPQ/z44fcYCNT1
bQ0km87F3nb6EJxVvCdJIPNGVwQBMdQl/FLwBVhEGJ9BIwog4fMzwhERjqnAP8H
bynAIQpG55BHaIXmv53x+GhhTVFMuMHFKMlVFOEFhbGNIZFE2bjJROFEiEjM5
NjAyNTc1Mjc0NjYxNDA3OCoKc2Vzc2lvbjAwMjConMz5oyo6BlJzYU1ENULG
AQoBMRrAAQoFQWxsb3cSSwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aMwoNb3Rz
OkJhdGNoR2V0KgoNb3RzOkRlc2NyaWJlKgoIb3RzOkdldCoKCW90czpMaXN0
KhJqCg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaTgohYWNzOm90czoqOio6
aW5zdGFuY2UvcmFtLXRlc3QtYXBwCilhY3M6b3RzOio6KjppbnN0YW5jZS9y
YW0tdGVzdC1hcHAvdGFibGUvKkoQMTk4MzQwNzU5Njk0NDIzN1IFMjY4NDJa
D0Fzc3VtZWRSb2xlVXNlcmAAahIzOTYwMjU3NTI3NDY2MTQwNzhyEnJhbXRl
c3RhcHByZWFkb25seQ=="
},
"RequestId": "EE788165-B760-4014-952C-E58ED229C80D"
}
```

2. Use the Table Store CLI tool to read data. (The Token parameter is supported by Table Store CLI tool v1.2.)

```
 python2.7 ots_console --url https://TableStoreTest.cn-hangzhou.
ots.aliyuncs.com --id STS. 0qJ2UE8AalcHdQ6n2Q8Q --key pSaUjb8O9m
U5M76nkC6FHt6wKwbCVYO27gxSEBAu --token=CAES6wMIARKAAVtHeNgU
nhk132OwDfxZTu8gPQCxfakYLeWha/FxoEYNqBKQTtyI4WPC5mpYuu8+n+
yamSYTI2VPQ/z44fcYCNT1bQ0km87F3nb6EJxVvCdJIPNGVwQBMdQl/FLwBVhEGJ9
BIwog4fMzwhERjqnAP8HbynAIQpG55BHaIXmv53x+GhhTVFMuMHFKMlVFOEFh
bGNIZFE2bjJROFEiEjM5NjAyNTc1Mjc0NjYxNDA3OCoKc2Vzc2lvbjAwMjCo
nMz5oyo6BlJzYU1ENULGAQoBMRrAAQoFQWxsb3cSSwoMQWN0aW9uRXF1YWxz
EgZBY3Rpb24aMwoNb3RzOkJhdGNoR2V0KgoNb3RzOkRlc2NyaWJlKgoIb3Rz
OkdldCoKCW90czpMaXN0KhJqCg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2Ua
TgohYWNzOm90czoqOio6aW5zdGFuY2UvcmFtLXRlc3QtYXBwCilhY3M6b3Rz
Oio6KjppbnN0YW5jZS9yYW0tdGVzdC1hcHAvdGFibGUvKkoQMTk4MzQwNzU5
Njk0NDIzN1IFMjY4NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIzOTYwMjU3NTI3
NDY2MTQwNzhyEnJhbXRlc3RhcHByZWFkb25seQ==

 OTS-TableStoreTest>: get test_write_read '001'
 age:INTEGER='30'
```

- Use temporary authorization to access the console

  In the previous example, RAM user ram_test_app can assume role RamTestAppReadOnly to obtain the permission to view all instances and tables. The logon process is as follows:

  1. Log on to the *RAM console* using the primary account.
  2. Log on to the RAM console using the primary account.
  3. Click **RAM User Logon Link** and log on using RAM user ram_test_app.
  4. Click the RAM user name and click **Switch Role**.
  5. Enter the enterprise alias and role name, and click **Switch**.

Use temporary authorization to call Java SDK

Create an OTSClient object, and pass in parameters of STS Token, including AccessKeyID, AccessKeySecret, and Token.

```
OTSClient client = new OTSClient(otsEndpoint, stsAccessKeyId,
stsAccessKeySecret, instanceName, stsToken);
```

4 Customize permissions

Action

Action is an API name that is used to specify APIs that are open or restricted for user access. When creating a Table Store authorization policy, add an `ots:` prefix for each Action and separate multiple Actions using commas. The asterisk (*) wildcard is also supported (including prefix matching and suffix matching).

Typical Action

- Single API

```
"Action": "ots:GetRow"
```

- Multiple APIs

```
"Action": [
"ots:PutRow",
"ots:GetRow"
]
```

- All read-only API

```
"Action": [
"ots:BatchGet*",
"ots:Describe*",
"ots:Get*",
"ots:List*",
"ots:ComputeSplitPointsBySize"
]
```

- All read and write API

```
"Action": "ots:*"
```

Resource

A Resource in Table Store is composed of multiple fields including product, region, user ID , instance name, and table name. Each field supports asterisk (*) wildcard (including prefix matching and suffix matching).

The format is as follows:

```
acs:ots:[region]:[user_id]:instance/[instance_name]/table/[table_name]
```

- The product is ots.
- [xxx] indicates a variable.

- The region is an abbreviation written in English, for example, cn-hangzhou. For more information about regions of service nodes, see *Region*.
- The user ID is the Alibaba Cloud account ID.

> **Note:**
> Instance names are case-insensitive. However, you must use lower case letters for `[instance_name]` in resource definition.

Typical Resource

- All resources of the users in all regions

```
"Resource": "acs:ots:*:*:*"
```

- All instances and their tables of user 123456 in China East 1 region

```
"Resource": "acs:ots:cn-hangzhou:123456:instance/*"
```

- Instance abc and its tables of user 123456 in China East 1 region

```
"Resource": [
"acs:ots:cn-hangzhou:123456:instance/abc",
"acs:ots:cn-hangzhou:123456:instance/abc/table/*"
]
```

- All instances whose names begin with abc and their tables

```
"Resource": "acs:ots:*:*:instance/abc*"
```

- All instances whose names begin with abc and their tables whose names begin with xyz (excluding instance resources, and not match acs:ots:*:*:instance/abc*)

```
"Resource": "acs:ots:*:*:instance/abc*/table/xyz*"
```

- All instances whose names end with abc and their tables whose names end with xyz.

```
"Resource": [
"acs:ots:*:*:instance/*abc",
"acs:ots:*:*:instance/*abc/table/*xyz"
]
```

API types

Table Store has two types of APIs

- Management APIs for reading from, and writing to, instances.
- Data APIs for reading from, and writing to, tables and rows.

The following table describes these APIS:

API/Action	API Type	Description
ListInstance	Management	Get instance list, called by console only
InsertInstance	Management	Create instance, called by console only
GetInstance	Management	Get instance meta, called by console only
DeleteInstance	Management	Delete instance, called by console only
ListTable	Data	Get table list, called by console or SDK
CreateTable	Data	Create table, called by console or SDK
UpdateTable	Data	Update table meta, called by console or SDK
DescribeTable	Data	Get table meta, called by console or SDK
DeleteTable	Data	Delete table, called by console or SDK
GetRow	Data	Read a record, called by SDK only
PutRow	Data	Insert a record, called by SDK only
UpdateRow	Data	Update a record, called by SDK only
DeleteRow	Data	Delete a record, called by SDK only
GetRange	Data	Readrange, called by SDK only
BatchGetRow	Data	Batch read records, called by SDK only
BatchWriteRow	Data	Batch write records, called by SDK only

- Resources accessed by management APIs

Management APIs are generally instance-related operations and can be called only on the console. The actions and resources definitions of management APIs determine subsequent use of the console. The prefix `acs:ots:[region]:[user_id]:` is omitted in the following accessed resources, leaving only the instance and table parts to be described.

API/Action	Resource Access
ListInstance	instance/*
InsertInstance	instance/[instance_name]
GetInstance	instance/[instance_name]
DeleteInstance	instance/[instance_name]

- Resources accessed by data APIs

  Data APIs are genearlly table-related operations and can be called both on the console and by the SDK. The actions and resources definitions of data APIs determine subsequent use of the console. The prefix `acs:ots:[region]:[user_id]:` is omitted in the following accessed resources, leaving only the instance and table parts to be described.

API/Action	Resource Access
ListTable	instance/[instance_name]/table/*
CreateTable	instance/[instance_name]/table/[table_name]
UpdateTable	instance/[instance_name]/table/[table_name]
DescribeTable	instance/[instance_name]/table/[table_name]
DeleteTable	instance/[instance_name]/table/[table_name]
GetRow	instance/[instance_name]/table/[table_name]
PutRow	instance/[instance_name]/table/[table_name]
UpdateRow	instance/[instance_name]/table/[table_name]
DeleteRow	instance/[instance_name]/table/[table_name]
GetRange	instance/[instance_name]/table/[table_name]
BatchGetRow	instance/[instance_name]/table/[table_name]
BatchWriteRow	instance/[instance_name]/table/[table_name]

- Limits

  — In a policy, actions and resources are verified by string matching. When using the asterisk (*) wildcard, prefix matching and suffix matching are distinguished. For example, if a

resource is defined as `acs:ots:*:*:instance/*/`, then `acs:ots:*:*:instance/abc` cannot be matched. If a resource is defined as `acs:ots:*:*:instance/abc`, then `acs:ots:*:*:instance/abc/table/xyz` cannot be matched.

- To manage instance resources on the Table Store console, you must have permission to read the `acs:ots:[region]:[user_id]:instance/*` resource to obtain the instance list on the console.
- For Batch APIs (such as BatchGetRow and BatchWriteRow), the backend service performs authentication for each table being accessed. Operations can be performed only when authentication is successful for all tables. Otherwise, a permission error is returned.

Condition

The policy supports multiple authentication conditions that are supported on all APIs of Table Store, including access IP address restriction, whether to access through HTTPS, whether to access through Multi-Factor Authentication (MFA), and access time restriction.

- Access IP address restriction

  Resource Access Management can restrict the source IP addresses used to access Table Store, and filter IP addresses based on the network segment. The following are typical application scenarios:

  - Multiple IP addresses are restricted. For example, only requests from 10.101.168.111 and 10.101.169.111 are allowed.

```
{
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ots:*",
        "Resource": "acs:ots:*:*:*",
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": [
                    "10.101.168.111",
                    "10.101.169.111"
                ]
            }
        }
    }
],
"Version": "1"
}
```

  - A single IP address is restricted. For example, only requests from 10.101.168.111 or 10.101.169.111/24 are allowed.

```
{
```

```
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ots:*",
        "Resource": "acs:ots:*:*:*",
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": [
                    "10.101.168.111",
                    "10.101.169.111/24"
                ]
            }
        }
    }
],
"Version": "1"
}
```

- HTTPS access restriction

  Resource Access Management can specify the use of HTTPS for access.

  Access by requests only through HTTPS

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ots:*",
            "Resource": "acs:ots:*:*:*",
            "Condition": {
                "Bool": {
                    "acs:SecureTransport": "true"
                }
            }
        }
    ],
    "Version": "1"
}
```

- MFA access restriction

  Resource Access Management can specify the use of MFA for access.

  Access by requests only through MFA

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ots:*",
            "Resource": "acs:ots:*:*:*",
            "Condition": {
                "Bool": {
                    "acs:MFAPresent ": "true"
                }
            }
        }
    ],
    "Version": "1"
```

```
}
```

- Access time restriction

  Resource Access Management can specify the time to grant access by a request, that is, it can determine if access is allowed or rejected by requests only before a specified time. For example,

  user access is allowed only before 00:00:00 January 1, 2016.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ots:*",
            "Resource": "acs:ots:*:*:*",
            "Condition": {
                "DateLessThan": {
                    "acs:CurrentTime": "2016-01-01T00:00:00+08:00"
                }
            }
        }
    ],
    "Version": "1"
}
```

Typical application scenarios

This section defines specific policies in typical scenarios and offers authorization methods.

- Multiple authorization conditions

  In this scenario, users accessing the 10.101.168.111/24 network segment can read from and write to all instances named online-01 and online-02 (including all tables of these instances). A restrictive access policy means access is allowed only before 0:00:00 January 1, 2016 through HTTPS.

  To grant policy permissions to a RAM user, follow these steps:

  1. Use the primary account to log on to the *RAM console* .
  2. In the left-side navigation pane, click **Policies**.
  3. In the upper-right corner, click **Create Authorization Policy**.
  4. Select **Blank Template**.
  5. Enter the **Authorization Policy Name** and copy the following content to **Policy Content**.

```
{
"Statement": [
    {
        "Effect": "Allow",
        "Action": "ots:*",
        "Resource": [
```

```
            "acs:ots:*:*:instance/online-01",
            "acs:ots:*:*:instance/online-01/table/*",
            "acs:ots:*:*:instance/online-02",
            "acs:ots:*:*:instance/online-02/table/*"
        ],
        "Condition": {
            "IpAddress": {
                "acs:SourceIp": [
                    "10.101.168.111/24"
                ]
            },
            "DateLessThan": {
                "acs:CurrentTime": "2016-01-01T00:00:00+08:00"
            },
            "Bool": {
                "acs:SecureTransport": "true"
            }
        }
    }
],
"Version": "1"
}
```

6. Click **Create Authorization** Policy and then click **Close**.
7. In the left-side navigation pane, click **Users**.
8. Locate the RAM user to be authorized, and click **Authorize**.
9. Select the policy created in the preceding steps, and click  .

10. Click **OK**.

- Reject requests

  In this scenario, users accessing the IP address 10.101.169.111 are not allowed to write to all tables of instances in the Beijing region whose names begin with `online` and `product`. Operations related to instances are not involved.

  To reject requests, first see the preceding steps to create a new policy and grant policy permissions to the designated RAM user. Then, during policy creation, copy the following content to **Policy Content**.

```
{
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ots:Create*",
                "ots:Insert*",
                "ots:Put*",
                "ots:Update*",
                "ots:Delete*",
                "ots:BatchWrite*"
            ],
            "Resource": [
```

```
                "acs:ots:cn-beijing:*:instance/online*/table/*",
                "acs:ots:cn-beijing:*:instance/product*/table/*"
            ],
            "Condition": {
                "IpAddress": {
                    "acs:SourceIp": [
                        "10.101.169.111"
                    ]
                }
            }
        }
    ],
    "Version": "1"
}
```

5 Use case

This sections explains safe practices when configuring a RAM user for authentication processes . Assume your Alibaba Account has no RAM user, You must replace the default AccessKey with your own AccessKey.

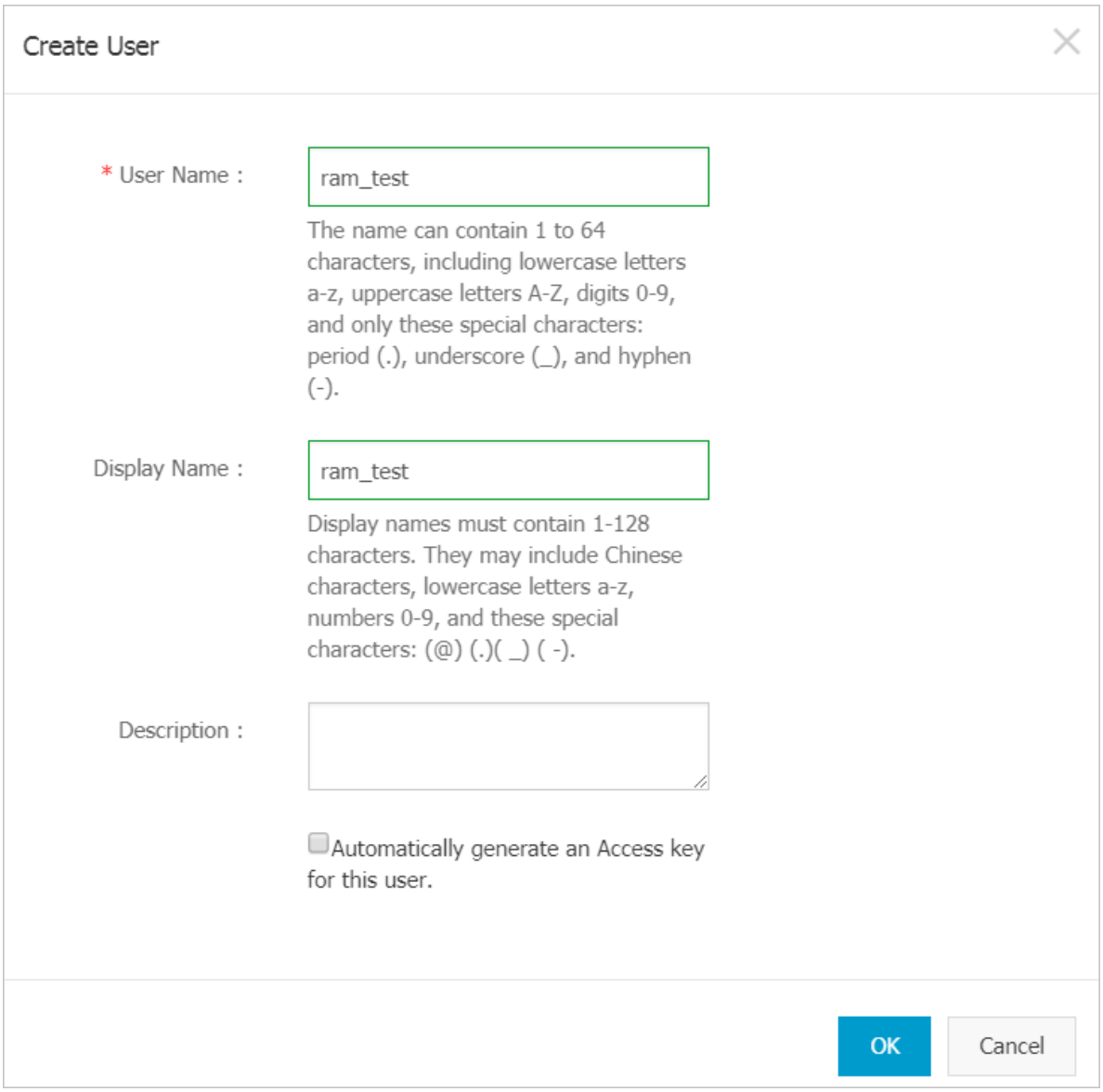
Create a subaccount

Assume you have a Table Store instance named ram-test-dev.

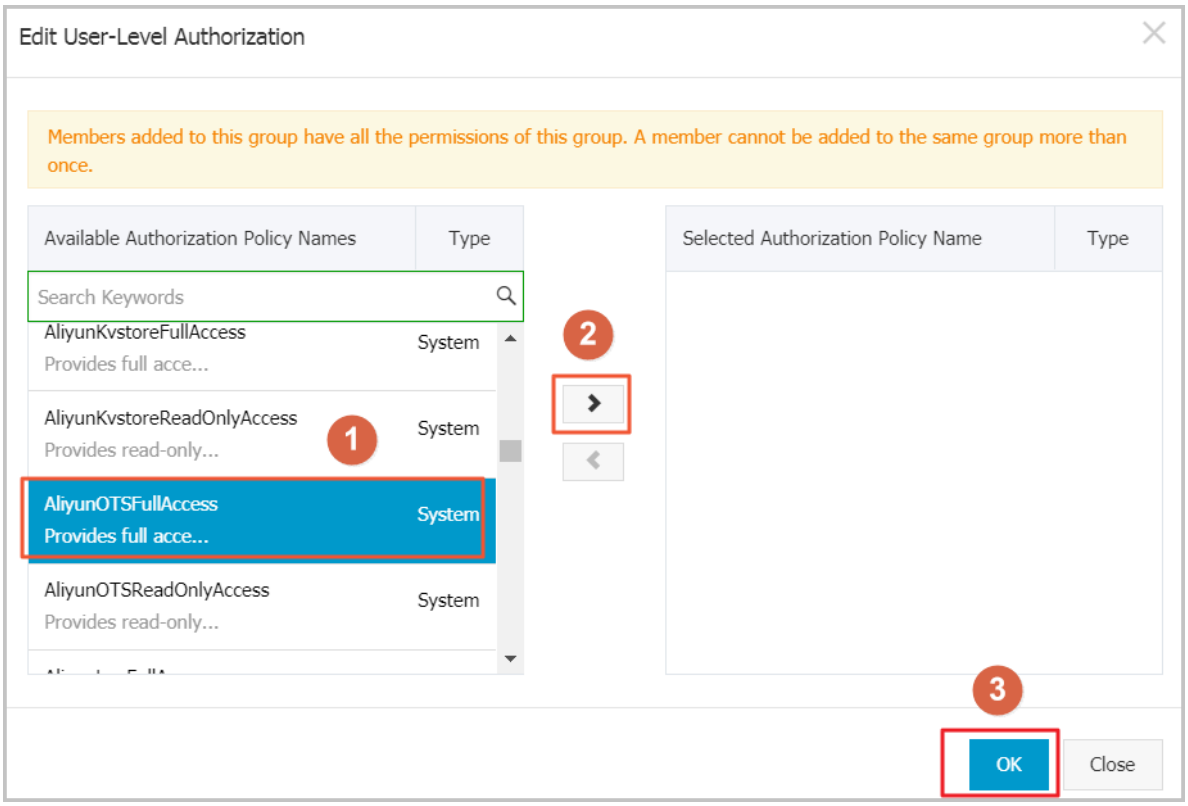
In this scenario, we do not recommend that you use the primary account to access an instance so as to avoid potential problems caused by uninitentionally exposing the AccessKey and password.

Procedure

1. Use the primary account to log on to the *RAM console*.
2. In the left-side navigation pane, click **Users**.
3. Click **Create User** to create a RAM user. Designate it with the same Table Store access permissions as the primary account.
4. Click **OK**. The AccessKey for the new RAM user ram_test is generated.



5. Save the AccessKey information.
6. Click **Authorize**to grant the RAM user full access permissions for Table Store.



7. (Optional) Click **Manage** to grant the account console logon or other permissions.

Example

In this example, the AccessKey is for ram_test. In actual scenarios, replace it with your own AccessKey.

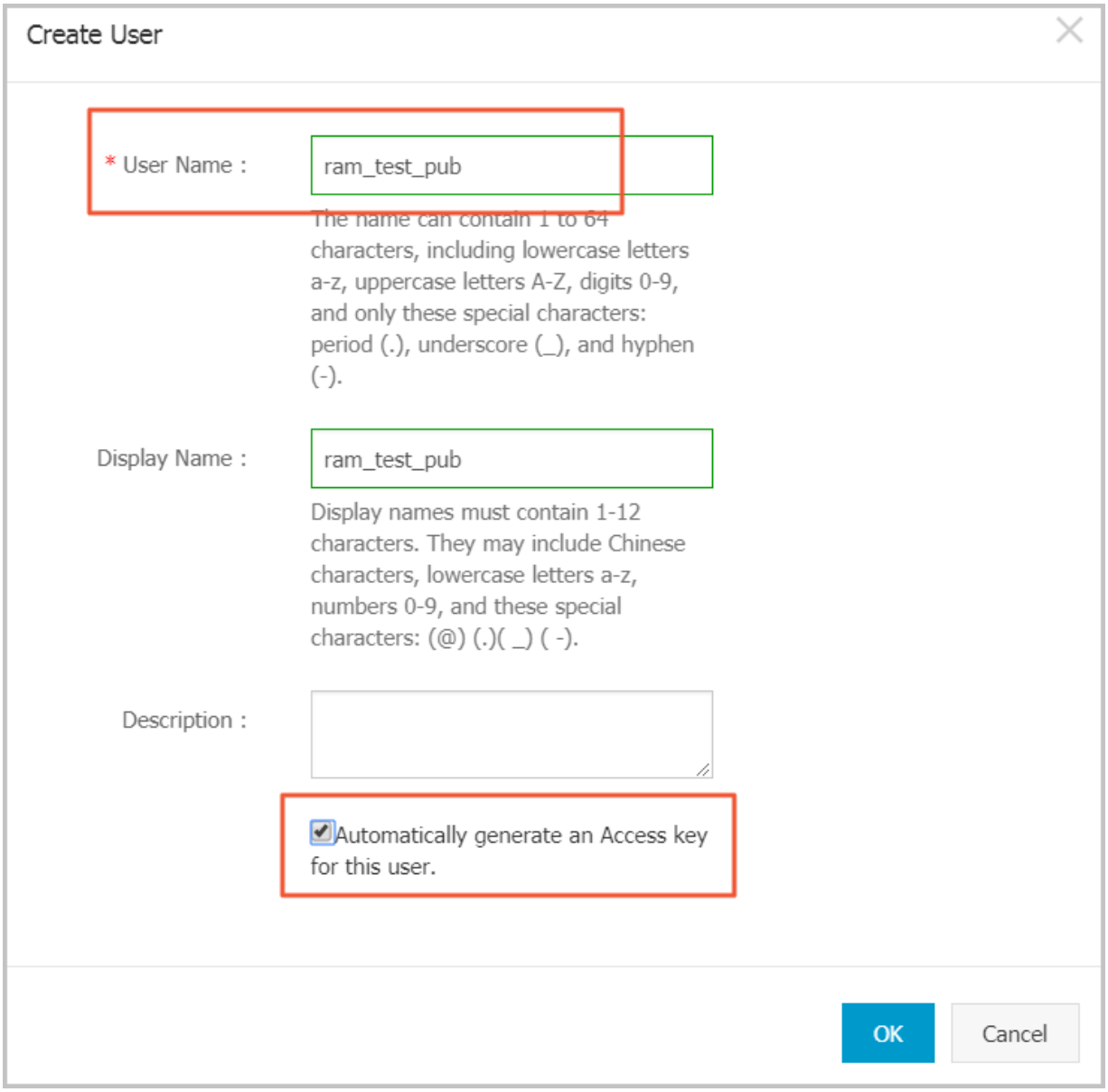
```
$python ots_console --url https://TableStoreTest.cn-hangzhou.ots
.aliyuncs.com --id VPIzjuDB6T4FGoWM --key r1usnIQ4Tw1yI6bNJkKa
y6A8EJoMvs

$OTS-TableStoreTest>: ct test pk1:string,pk2:integer readrt:1 writert:
1
Table test has been created successfully.

$OTS-TableStoreTest>: dt test
You will delete the table:test!

press Y (confirm) :Y
Table test has been deleted successfully.
```

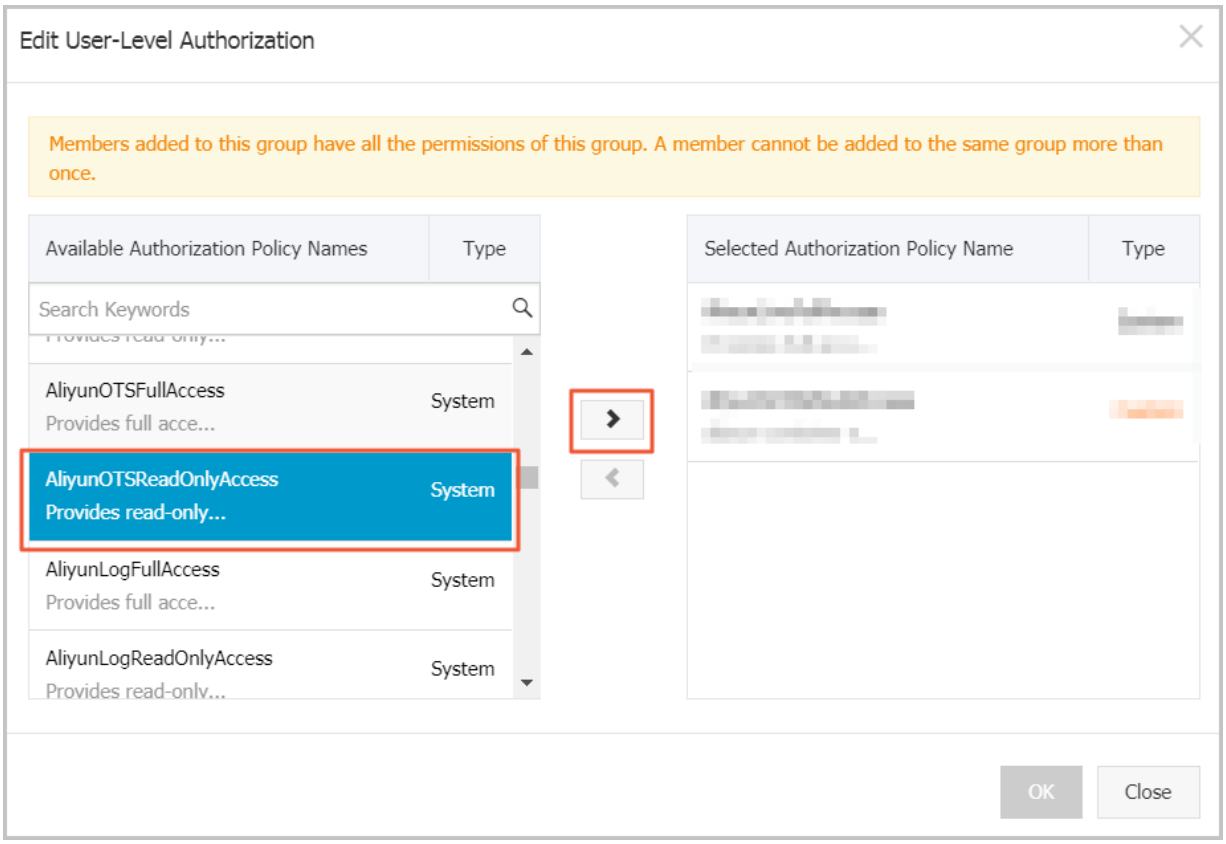
The ram_test subaccount can be used for all general operations, so as to avoid exposing the AccessKey of the primary account.

Read/write permission separation

To share data of an instance in Table Store without data modification, you can separate read/write permission by creating a subaccount with read-only permission.

Create a RAM user named ram_test_pub. Select ReadOnly on the **Edit User-Level Authorization** page to grant the RAM user read-only access permission for Table Store.





Example

Use the AccessKey of the RAM user to test the permissions of creating and deleting a table. In this example, the AccessKey is for ram_test_pub. In actual scenarios, replace it with your own AccessKey.

```
$python ots_console --url https://TableStoreTest.cn-hangzhou.ots
.aliyuncs.com --id ftWyMEYu1rBYTbWM --key u4qR5IGu5xJsvSO1y8mo
yC6n5vA7af

$OTS-TableStoreTest>: ct test pk1:string,pk2:integer readrt:1 writert:
1
Fail to create table test.

$OTS-TableStoreTest>: dt test
You will delete the table:test!

press Y (confirm) :Y
Fail to delete table test.
```

Note:

Due to the read-only access permissions granted to RAM user ram_test_pub, it cannot be used to create or delete a table.