ALIBABA CLOUD

Alibaba Cloud

Resource Orchestration Service

FAQ

Document Version: 20220427

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.Why did template validation fail?

This topic describes why template validation fails.

Format

Make sure that the template is a valid JSON or YAML file with UTF-8 encoding. The files are valid if their content can be correctly interpreted as JSON or YAML.

Version (ROSTemplateFormatVersion)

Make sure that ROSTemplateFormatVersion is correctly spelled and the value is 2015-09-01.

Mappings

Make sure that mapping definitions meet the Resource Orchestration Service (ROS) requirements.

> **Notice** Functions cannot be used in mappings.

Parameters

Make sure that parameter definitions meet the ROS requirements.

> **Notice** Functions cannot be used in parameters. If the parameter definition contains a constraint and a default value, the default value must also conform to the parameter constraint.

Resources

Resource IDs cannot contain forward slashes (/).

A resource definition must contain a Type property whose value is of the String type.

A resource definition can only contain values of the Type, Properties, Metadata, DependsOn, DeletionPolicy, and Description properties.

Outputs

A value must be defined in an output.

Unsupported resource types

If the template contains unsupported resources, validation fails.

Others

Make sure that the size of the template file is no larger than 512 KB.

Make sure that the template contains only the following top-level objects: ROSTemplateFormatVersion, Description, Mappings, Parameters, Resources, and Outputs.

2.Why a stack fails to be created?

This topic describes the causes why a Resource Orchestration Service (ROS) stack fails to be created.

Name duplication

Stacks that you create must have unique names.

Invalid name format

The name of a stack can contain only letters and digits. The name cannot start with a digit.

Resource creation failure

If a resource in a stack fails to be created, the stack also fails to be created. For example, if an Elastic Compute Service (ECS) resource in a stack fails to be created due to insufficient memory in a specified region, the stack also fails to be created.

Quantity limit reached

You can create up to 200 stacks in ROS. For more information, see Limits.

3.What do I do if a stack fails to be deleted?

This topic describes the possible reasons why a stack fails to be deleted and the corresponding solutions.

A stack may not be deleted due to the following reasons:

- The stack is in the process of an operation.
- You do not have the permissions to delete the stack.

When a stack fails to be deleted, perform the following operations:

1. If the stack is in the process of an operation such as create or update, you must wait until the operation is complete before you can delete the stack.
2. If you use a Resource Access Management (RAM) user to delete a stack, you must make sure that the RAM user has the AliyunROSFullAccess permission. For more information, see Grant permissions to a RAM user.
3. If the problem persists, submit a ticket.

4.How do I configure password confirmation?

This topic describes how to configure password confirmation when you create resource instances.

Context

When you create a stack, you enter the logon password only once. However, when you enter the password, it is possible to enter a wrong password and get locked out of your service instances, such as ECS and ApsaraDB for RDS instances.

Procedure

1. When you define `InstancePassword`, set Confirm to *true* to enable password confirmation.

```
"InstancePassword": {
    "Confirm": true,
    "NoEcho": true,
    "MaxLength": "41",
    "Description": "The login password of ECS instances",
    "Type": "String",
    "ConstraintDescription": "Consist of 8 to 41 characters of alphanumeric characters"
,
    "MinLength": "8",
    "AllowedPattern": "[a-zA-Z0-9]*"
},
```

> **Notice** Confirm can only be used when NoEcho is set to *true*.

2. Configure Instance Password in the **Configure Template Parameters** step of the Create Stack wizard.

5.How do I specify images for ECS resources?

This topic describes how to use Resource Orchestration Service (ROS) to specify images for ECS resources.

Prerequisites

An Alibaba Cloud account is created. To create an Alibaba Cloud account, go to the Alibaba Cloud official website.

Context

If you use one of the following resource types to create ECS instances, you must specify images for the instances.

- ALIYUN::ECS::Instance
- ALIYUN::ECS::InstanceClone
- ALIYUN::ECS::InstanceGroup
- ALIYUN::ECS::InstanceGroupClone

You can use one of the following methods to set the ImageId parameter in a stack template:

- Specify an image ID.
- Specify an image by fuzzy match.
- Select an image by using the AssociationProperty parameter.

Specify an image ID

If you have the ID of the image that you want to use, specify the ImageId parameter in the template.

1. Log on to the ECS console.
2. In the left-side navigation pane, choose **Instances & Images** > **Images**.
3. Click the **Public Image** tab.
   Find the target image ID on the Public Image tab and record it for later use.
4. In the ROS template, set the ImageId parameter to the image ID.

   For more information about template creation, see Create a template.

```
"ImageId": { "Type": "String", "Description": "Image Id, represents the image resource to startup one ECS instance", "Default": "centos_7_04_64_20G_alibase_201701015.vhd" },
```

Specify an image by fuzzy match

If you do not need specific versions of CentOS or Ubuntu images, you can use fuzzy match to specify an image ID. ROS uses the value you enter to find the best match.

ROS uses the following rules to find a match:

- If you specify only the operating system of images, such as CentOS, Windows, or Ubuntu, ROS matches the latest 64-bit image version.
- If you specify a major version of an operating system, ROS matches the latest 64-bit version of the

system based on the major version. For example, if you enter CentOS_6, ROS matches the latest 64-bit version of CentOS 6. If you enter Ubutun_14, ROS matches the latest 64-bit version of Ubutun_14. If you enter Win2008r2, ROS matches the latest 64-bit version of Windows Server 2008 R2.

- If you use an asterisk (*) to replace a part of the image ID, such as `centos_6_09_64_20G_alibase*.vhd`, ROS matches the latest centos_6_09_64_20G_alibase public image version. Fuzzy match is used in the sample templates. In many cases, the image ID is set to CentOS_7 or Ubuntu_14.

Examples

```
"ImageId": {
   "Type": "String",
   "Description": "ECS Image",
   "Label": "ECS Image",
   "Default": "cent****"
 },
```

Select an image by using the AssociationProperty parameter

Include AssociationProperty as part of your image ID definition when you declare it as a parameter in the Parameters section. Then, ROS lists the images within a region that you can select when you create the ECS instance.

AssociationProperty is used in the following example:

```
   "ImageId": {
      "AssociationProperty":"ALIYUN::ECS::Instance:ImageId",
      "Type" : "String",
      "Default": "centos_7_04_64_20G_alibase_20170****.vhd",
      "Description": "IDs of available images"
    }
```

In addition to the available image IDs, ROS displays the default image ID, or indicates whether the values specified by AllowedValues are available. Select the appropriate image ID to create ECS instances.

6.How do I specify different instance names and hostnames when I create multiple ECS instances at a time?

You can use ALIYUN::ECS::InstanceGroup to create multiple ECS instances at a time.

Specifically, use the InstanceName and HostName properties to define instance names and hostnames. You can use the following syntax to specify each instance name or hostname in Resource Orchestration Service (ROS):

name_prefix[begin_number,bits]name_suffix. The following fields are included in the syntax:

- name_prefix: Required. The prefix of the instance name or hostname.
- [begin_number,bits]: Required. The part that changes for each name. begin_number specifies the starting number of the instance name or hostname. bits specifies the number of digits of each number.

  This part must meet the following requirements to be correctly parsed:

  - begin_number and bits are separated by a comma without spaces.
  - The value of bits ranges from 1 to 4.
  - The value of begin_number ranges from 0 to 9999.

  The value of bits follows the following rules:

  - If you only specify a value for begin_number, the default value of bits is 4.
  - If you do not specify a value for begin_number or bits, begin_number is set to 0 and bits is set to 4.
  - If the specified value of begin_number has more digits than that specified in bits, the larger digit number takes effect. For example, in [1234,1], the value of begin_number is 1234, and the value of digits is 1. 1234 is a four-digit number within the valid value range of 0 to 9999. Therefore, the actual value of bits is 4.
- name_suffix: Optional. The suffix of the instance name or hostname.

Example:

```
{
  "ROSTemplateFormatVersion" : "2015-09-01",
  "Resources" : {
    "WebServer": {
      "Type": "ALIYUN::ECS::InstanceGroup",
      "Properties": {
        "ImageId" : "Cen****",
        "InstanceType": "ecs.n4.large",
        "Password": "Test1234",
        "MinAmount": 2,
        "MaxAmount": 2,
        "SecurityGroupId": "sg-2zedcm7ep5quses0****",
        "SystemDiskCategory": "cloud_efficiency",
        "IoOptimized": "optimized",
        "InstanceName": "my.test-[1114]",
        "HostName": "host[]"
      }
    }
  }
}
```

ROS will create two ECS instances at a time based on the preceding template.

- The instance names of the two instances are my.test-1114 and my.test-1115.
- The hostnames of the two instances are host0000 and host0001.

Notice The InstanceName and HostName properties must be specified based on the preceding naming conventions to be correctly parsed. Otherwise, the template cannot be validated.

7.FAQ about SDKs

This topic provides answers to some commonly asked questions about SDKs.

What do I do if the ErrCode:SDK.ServerUnreachable error is reported when I use SDKs?

Problem description

The following error may occur when you use SDKs:

```
ErrCode:ServerUnreachable
ErrMsg:SocketTimeoutException has occurred on a socket read or accept.
```

Solution

1. Configure the ClientToken parameter in the request to prevent repeated request processing.

   The sample Python code:

```
#!/usr/bin/env python
# coding=utf-8
import time
import uuid
from aliyunsdkcore.client import AcsClient
from aliyunsdkcore.acs_exception.exceptions import ClientException, ServerException
from aliyunsdkros.request.v20190910.CreateStackRequest import CreateStackRequest
ACCESS_KEY_ID = ''
ACCESS_KEY_SECRET = ''
REGION_ID = 'cn-beijing'
def generate_client_token(prefix, add_uuid, *suffix_list):
    if prefix:
        t = [prefix]
    else:
        t = []
    if add_uuid:
        t.append(str(uuid.uuid1())[:-13])
    t.extend(suffix_list)
    r = '_'.join(t)
    if len(r) > 64:
        r = r[:64]
    return r
def retry_with_client_token(retry_interval=10, retry_time_out=300):
    def wrapper(fun):
        def retry_func(*args, **kwargs):
            retry_timeout = retry_time_out
            elapsed_time = 0
            # This parameter is used to ensure the idempotence of the request. You can
use the client to generate the value, but you must ensure that it is unique among the d
ifferent requests.
            kwargs['ClientToken'] = generate_client_token(None, True)
            while True:
                try:
                    retry_timeout -= elapsed_time
                    return fun(*args, **kwargs)
```

```
                    return fun(*args, **kwargs)
                except Exception as e:
                    if isinstance(e, ClientException):
                        if e.error_code != 'SDK.HttpError':
                            raise
                    elif isinstance(e, ServerException):
                        if e.error_code not in ('LastTokenProcessing', 'ServiceUnavaila
ble'):
                            raise
                    else:
                        raise
                    time.sleep(retry_interval)
                    elapsed_time += retry_interval
                    if elapsed_time >= retry_timeout:
                        raise
        return retry_func
    return wrapper
class RosClient(object):
    def __init__(self):
        self.client = AcsClient(ACCESS_KEY_ID, ACCESS_KEY_SECRET, REGION_ID)
    def _send_request(self, request, params, **kwargs):
        request.set_accept_format('json')
        request.set_connect_timeout(10000)
        request.set_read_timeout(0.01)
        params.update(kwargs)
        for name, value in params.items():
            set_func = getattr(request, 'set_{}'.format(name))
            set_func(value)
        return self.client.do_action_with_exception(request)
    @retry_with_client_token(retry_interval=10, retry_time_out=300)
    def create_stack(self, params, **kwargs):
        request = CreateStackRequest()
        return self._send_request(request, params, **kwargs)
if __name__ == '__main__':
    ros_client = RosClient()
    result = ros_client.create_stack(dict(
        StackName='test',
        TemplateBody="{\"ROSTemplateFormatVersion\": \"2015-09-01\"}"
    ))
    print('stack_id:', result)
```

2. If the error persists, submit a ticket.