Alibaba Cloud Resource Orchestration Service

SDK Reference

Issue: 20190307

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.	 Notice: 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]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 Download and use instructions

Click on the link below to see the download address and instructions for the Alibaba Cloud SDK.

Download link:

Alibaba Cloud SDK

Instructions:

Python SDK

Java SDK

PHP SDK

.NET SDK

GO SDK

Node.js SDK

2 Examples of using the ROS SDK

This topic describes how to manage Resource Orchestration Services (ROS) resource stacks by using an SDK.

Overview

You can create and manage resource stacks by using the ROS console or APIs. This topic shows how to use the Python SDK to manage ROS resource stacks, including the following operations:

- View the list of available regions.
- Create a resource stack.
- View the status of a resource stack.
- Delete a resource stack.

Before you start

Download and install the Python SDK

> Notice:
>
> - The commands in the example run in Linux Shell. Windows or DOS users need to modify the syntax accordingly.
> - The ROS Python SDK is only compatible with Python 2.7 and later versions.

Use pip to install aliyun-python-sdk-core:

```
pip install aliyun-python-sdk-core
```

aliyun-python-sdk-core is included in all Python SDKs of Alibaba Cloud. If the installation fails and you receive an error message, a likely reason is that the user does not have the write access permission to the Python installation path. In this case , modify the preceding command as follows. Modify the command similarly in case you receive a permission error message when you install the ROS SDK.

```
sudo pip install aliyun-python-sdk-core
```

Install the ROS SDK:

```
pip install aliyun-python-sdk-ros
```

Initialize the SDK

Import the required libraries:

```
import   json
from   aliyunsdkc  ore   import   client
from   aliyunsdkr  os . request . v20150901 .  CreateStac  ksRequest
  import   CreateStac  ksRequest
from   aliyunsdkr  os . request . v20150901 .  DescribeSt
ackDetailR  equest   import   DescribeSt  ackDetailR  equest
from   aliyunsdkr  os . request . v20150901 .  DeleteStac  kRequest
import   DeleteStac  kRequest
from   aliyunsdkr  os . request . v20150901 .  DescribeRe
gionsReque  st   import   DescribeRe  gionsReque  st
```

Initialize the SDK client: When creating the client instance, you must specify the Region ID, AccessKey ID, and AccessKey Secret parameters.

```
AK  = '< Your   Access   Key   Id >'
SECRET  = '< Your   Access   Key   Secrect >'
Region  = '< Region   Id >'
clt  =  client . AcsClient ( AK ,  SECRET ,  Region )
```

Procedure

View the list of available regions

Run the following commands to view the list of available regions.

```
def   describe_r  egion ():
   """ descrbe   regions   list """
    req  =  DescribeRe  gionsReque  st ()   # This   line   is   used
  to   construct   the   request   object .
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  200 :
        regions  =  json . loads ( body )
        return   regions
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )
```

Create a resource stack

To create a resource stack, you must specify the stack parameters.

- Name indicates the name of the resource stack to be created. The specified name must be unique in the same userspace.
- TimeoutMins indicates the specified timeout for stack creation, measured in minutes. A failure will occur if the stack cannot be created within the specified time.
- Template indicates the template based on which the stack is created.
- status indicates the response status of the request in the form of HTTP response status code. Generally, 2xx indicates success, and 4xx or 5xx indicate errors.

- headers indicate the HTTP response headers, which are the message headers returned in response to receiving an HTTP request.
- body indicates the HTTP response body, which is the message body returned in response to receiving an HTTP request.

```
def   create_sta  ck ():
   """ create   stack """
    global   result
    req  =  CreateStac  ksRequest ()
    create_sta  ck_body  =  dict ()   # This   line   is   used   to
  create   the   body   of   the   request .
    create_sta  ck_body [" Name "] = ' empty - template - test '
    create_sta  ck_body [" Template "] = '{" ROSTemplat  eFormatVer
sion ": " 2015 - 09 - 01 "}'
    create_sta  ck_body [" TimeoutMin  s "] =  60
    req . set_conten  t ( json . dumps ( create_sta  ck_body ))
    req . set_conten  t_type (' applicatio  n / json ')
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  201 :   # The   status   code   201   indicates
  that   the   request   has   been   been   processed .
        token  =  json . loads ( body )
        return   result
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )
```

If the request for creating a stack has been processed, then the returned body includes the stack ID and name.

```
{' Id ': ' 2ffcfe5d - d35c - 4f35 - 858d - dda78922b7  c6 ', ' Name ':
 ' empty - template - test '}
```

The response is returned at the same time the request is processed. However, it does not mean that the resource stack has been created because the creation progress is running in the background using ROS. You can use the ROS console or APIs to view the stack status and events.

View the status of a resource stack

To view the status of a resource stack, you must provide the stack ID and name. You can obtain the stack ID and name from the response after you have submitted the request for creating a stack.

```
def   describe_s  tack ():
   """ describe   stack """
    req  =  DescribeSt  ackDetailR  equest ()
    req . set_StackN  ame ( result [' Name '])
    req . set_StackI  d ( result [' Id '])
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  200 :   # The   status   code   200   indicates
  that   the   request   has   been   received   and   is   being
processed .
        res  =  json . loads ( body )
```

```
        if   res [' Status '] ! = ' CREATE_IN_  PROGRESS ':  # This
line   describes   the   condition   that   the   stack   status   is
  not  " CREATE_IN_  PROGRESS ".
            return   res
        else :
            return   describe_s  tack ()
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )
```

Delete a resource stack

To delete a resource stack, you must provide the stack ID and name. You can obtain the stack ID and name from the response after you have submitted the request for creating a stack.

```
def   delete_sta  ck ():
   """ delete   stack """
    req  =  DeleteStac  kRequest ()
    req . set_StackN  ame ( result [' Name '])
    req . set_StackI  d ( result [' Id '])
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  204 :   # The   status   code   204   indicates
  that   the   request   has   been   fulfilled .
        return   body
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )
```

Sample code

You can provide your account and stack details to perform the previous operations. The sample code is described as follows.

```
import   json
from   aliyunsdkc  ore   import   client
from   aliyunsdkr  os . request . v20150901 .  CreateStac  ksRequest
  import   CreateStac  ksRequest
from   aliyunsdkr  os . request . v20150901 .  DescribeSt
ackDetailR  equest   import   DescribeSt  ackDetailR  equest
from   aliyunsdkr  os . request . v20150901 .  DeleteStac  kRequest
import   DeleteStac  kRequest
from   aliyunsdkr  os . request . v20150901 .  DescribeRe
gionsReque  st   import   DescribeRe  gionsReque  st

AK  = '< Your   Access   Key   Id >'
SECRET  = '< Your   Access   Key   Secrect >'
Region  = '< Region   Id >'   # Specify   the   region   parameter
in   the   following   format : " cn - beijing ",  " cn - hangzhou ".


clt  =  client . AcsClient ( AK ,  SECRET ,  Region )


def   describe_r  egion ():
   """ descrbe   regions   list  """
    req  =  DescribeRe  gionsReque  st ()
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  200 :
```

```
        regions  =  json . loads ( body )
        return   regions
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )


def   create_sta  ck ():
   """ create   stack """
    global   result
    req  =  CreateStac  ksRequest ()
    create_sta  ck_body  =  dict ()
    create_sta  ck_body [" Name "] = ' empty - template - test000000
'
    create_sta  ck_body [" Template "] = '{" ROSTemplat  eFormatVer
sion ": " 2015 - 09 - 01 "}'
    create_sta  ck_body [" Parameters "] =  dict ()
    create_sta  ck_body [" TimeoutMin  s "] =  60
    req . set_conten  t ( json . dumps ( create_sta  ck_body ))
    req . set_conten  t_type (' applicatio  n / json ')
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  201 :
        result  =  json . loads ( body )
        return   result
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )


def   describe_s  tack ():
   """ describe   stack """
    req  =  DescribeSt  ackDetailR  equest ()
    req . set_StackN  ame ( result [' Name '])
    req . set_StackI  d ( result [' Id '])
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  200 :
        res  =  json . loads ( body )
        if   res [' Status '] ! = ' CREATE_IN_  PROGRESS ':
            return   res
        else :
            return   describe_s  tack ()
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )


def   delete_sta  ck ():
   """ delete   stack """
    req  =  DeleteStac  kRequest ()
    req . set_StackN  ame ( result [' Name '])
    req . set_StackI  d ( result [' Id '])
    status ,  headers ,  body  =  clt . get_respon  se ( req )
    if   status  ==  204 :
        return   body
    else :
        return  ' Unexpected   errors :  status =% d ,  error =% s '
% ( status ,  body )


if   __name__  == ' __main__ ':
    describe_r  egion ()
    create_sta  ck ()
    describe_s  tack ()
```

```
    delete_sta  ck ()
```