

阿里云 资源编排

SDK参考

文档版本：20200430

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务时间约十分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 注意： 权重设置为0，该服务器不会再接受新请求。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	单击 设置 > 网络 > 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
Courier字体	命令。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid Instance_ID</code>
[]或者[a b]	表示可选项，至多选择一个。	<code>ipconfig [-all]-t</code>
{ }或者[a b]	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

法律声明.....	I
通用约定.....	I
1 SDK下载和使用说明.....	1
2 Python SDK使用示例.....	2
3 Java SDK使用示例.....	7

1 SDK下载和使用说明

资源编排支持Java、Python、PHP、.NET、Go和Node.js开发。



说明:

关于SDK的常见问题，请参见[#unique_4](#)。

下表列举了各语言SDK的下载地址和开发指南，更多SDK的信息，请访问[阿里云开发平台](#)。

Alibaba Cloud SDK	资源编排SDK	说明文档
Alibaba Cloud SDK for Java	Alibaba Cloud ROS SDK for Java	快速开始
Alibaba Cloud SDK for Python	Alibaba Cloud ROS SDK for Python	快速开始
Alibaba Cloud SDK for PHP	Alibaba Cloud ROS SDK for PHP	快速开始
Alibaba Cloud SDK for .NET	Alibaba Cloud ROS SDK for .NET	快速开始
Alibaba Cloud SDK for Go	Alibaba Cloud ROS SDK for Go	#unique_9
Alibaba Cloud SDK for Node.js	Alibaba Cloud ROS SDK for Node.js	#unique_10

使用指南

后面的文档为您提供了Java和Python的几种SDK使用示例，如何查询所有SDK使用示例请参考此步骤。

API文档提供了调试功能（需登录后使用），在任意API文档中选择调试功能，可以查看示例代码，输入对应参数调用成功后可以看到详细的示例代码。

2 Python SDK使用示例

本文为您介绍如何使用资源编排服务（ROS）的Python SDK来创建和管理资源栈。

背景信息

您除了可以在ROS控制台创建资源栈，还可以使用API代码来创建和管理资源栈。

准备工作

1. 下载及安装Python SDK。

- a) 使用pip安装aliyun-python-sdk-core。

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



说明：

aliyun-python-sdk-core是所有阿里云官方Python SDK的公共组件。如果安装过程提示权限错误，可能是因为没有Python安装路径的写权限。您也可以将此命令更改为`sudo pip install aliyun-python-sdk-core`。

- b) 安装ROS SDK。

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

2. 初始化SDK。

- a) 导入相关的包。

```
from aliyunsdkcore.client import AcsClient
from aliyunsdkros.request.v20190910.CreateStackRequest import CreateStackRequest
from aliyunsdkros.request.v20190910.GetStackRequest import GetStackRequest
from aliyunsdkros.request.v20190910.DeleteStackRequest import DeleteStackRequest
from aliyunsdkros.request.v20190910.DescribeRegionsRequest import DescribeRegionsRequest
from aliyunsdkros.request.v20190910.ListStacksRequest import ListStacksRequest
```

- b) 初始化SDK客户端对象。

```
AK = '<yourAccessKeyId>'
SECRET = '<yourAccessKeySecret>'
Region = '<yourRegionId>'
client = AcsClient(AK, SECRET, Region)
```



注意：

- 示例中的命令行都使用Linux的shell，如果您是Windows/DOS用户，则需要根据情况修改。

- ROS Python SDK依赖Python 2.7以上版本。
- ROS Python SDK 3.0.0及以上版本，支持v20150901版和v20190910版API。

查询可用地域列表

您可以使用Python SDK查询可用地域列表。

```
def describe_region():
    """describe regions list """
    request = DescribeRegionsRequest()
    request.set_connect_timeout(10000)
    request.set_read_timeout(10000)
    request.set_accept_format("json")
    response = client.do_action_with_exception(request)

    return response.decode("utf-8")
```

创建资源栈

创建资源栈时，您必须指定以下参数：

- **StackName**：将要创建的资源栈的名称。每个用户空间下的资源栈名称不能重复。
- **TimeoutInMinutes**：资源栈创建时长，单位为分钟。如果在指定时间内不能完成创建，则超时失败。
- **TemplateBody**：创建的资源栈使用的模板内容。
- **TemplateURL**：模板主体的文件的位置。必须指定**TemplateBody**或**TemplateURL**，但不能同时指定两者。
- **Parameters**：创建的资源栈所需要的参数。需要在模板中定义key。

```
stack_name = "MyStack"
timeout = 10
template_body = """
{
  "ROSTemplateFormatVersion": "2015-09-01",
  "Parameters": {
    "VpcName": {
      "Type": "String",
      "Description": "Vpc Name",
      "Label": "Vpc Name"
    },
    "CidrBlock": {
      "Type": "String",
      "Description": "Vpc CidrBlock",
      "Label": "Vpc CidrBlock"
    }
  },
  "Resources": {
    "Vpc": {
      "Type": "ALIYUN::ECS::VPC",
      "Properties": {
        "CidrBlock": {
          "Ref": "CidrBlock"
        },
        "VpcName": {
```



```
return response.decode("utf-8")
```

```
import json
from time import sleep
from aliyunsdkcore.client import AcsClient
from aliyunsdkros.request.v20190910.CreateStackRequest import CreateStackRequest
from aliyunsdkros.request.v20190910.GetStackRequest import GetStackRequest
from aliyunsdkros.request.v20190910.DeleteStackRequest import DeleteStackRequest
from aliyunsdkros.request.v20190910.DescribeRegionsRequest import DescribeRegionsRequest
from aliyunsdkros.request.v20190910.ListStacksRequest import ListStacksRequest

AK = '<yourAccessKeyId>'
SECRET = '<yourAccessKeySecret>'
Region = '<yourRegionId>' # 例如: 'cn-beijing'、'cn-hangzhou'

client = AcsClient(AK, SECRET, Region)

def describe_region():
    """describe regions list"""
    request = DescribeRegionsRequest()
    request.set_connect_timeout(10000)
    request.set_read_timeout(10000)
    request.set_accept_format("json")
    response = client.do_action_with_exception(request)

    return response.decode("utf-8")

def create_stack(stack_name, timeout, template_body, params=[]):
    """create stack"""
    request = CreateStackRequest()
    request.set_accept_format("json")
    request.set_StackName(stack_name)
    request.set_TimeoutInMinutes(timeout)
    # 若模板较大, 建议使用TemplateURL参数, 避免URL过长, 调用失败。
    # 您也可以使用Body Param + HTTP POST的方式传递模板内容。
    # request.set_method("POST")
    # request.add_body_params("TemplateBody", template_body)
    request.set_TemplateBody(template_body)
    response = client.do_action_with_exception(request)

    return response.decode("utf-8")

def get_stack(stack_id):
    """get descriptions of the stack"""
    request = GetStackRequest()
    request.set_accept_format("json")
    request.set_StackId(stack_id)
    response = client.do_action_with_exception(request)

    return response.decode("utf-8")

def delete_stack(stack_id):
```

```
"""delete stack"""
request = DeleteStackRequest()
request.set_accept_format("json")
request.set_StackId(stack_id)
response = client.do_action_with_exception(request)

return response.decode("utf-8")

if __name__ == '__main__':
    test_template = """
    {
      "ROSTemplateFormatVersion": "2015-09-01",
      "Parameters": {
        "VpcName": {
          "Type": "String",
          "Description": "Vpc Name",
          "Label": "Vpc Name"
        },
        "CidrBlock": {
          "Type": "String",
          "Description": "Vpc CidrBlock",
          "Label": "Vpc CidrBlock"
        }
      },
      "Resources": {
        "Vpc": {
          "Type": "ALIYUN::ECS::VPC",
          "Properties": {
            "CidrBlock": {
              "Ref": "CidrBlock"
            },
            "VpcName": {
              "Ref": "VpcName"
            }
          }
        }
      }
    }
    """

    parameters = [
      {"ParameterKey": "CidrBlock", "ParameterValue": "192.168.0.0/16"},
      {"ParameterKey": "VpcName", "ParameterValue": "TestVpc"}
    ]

    describe_region()
    stack = create_stack('MyStack', 10, test_template, parameters)
    get_stack(json.loads(stack)["StackId"])
    sleep(3) # 等待创建资源栈任务执行完毕
    delete_stack(json.loads(stack)["StackId"])
```

3 Java SDK使用示例

本文为您介绍如何使用资源编排服务ROS（Resource Orchestration Service）的Java SDK来创建和管理资源栈。

背景信息

您除了可以在ROS控制台创建资源栈，还可以使用API代码来创建和管理资源栈。

准备工作

1. 下载及安装Java SDK。



说明：

建议您使用JRE 1.8及以上版本。

2. 在pom.xml中，添加aliyun-java-sdk-core及其它依赖包。

```
<dependencies>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.4.0</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-ros</artifactId>
    <version>3.2.0</version>
  </dependency>
</dependencies>
```

3. 初始化SDK。

- a) 导入相关的包。

```
import java.util.ArrayList;
import java.util.List;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.HttpResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.ros.model.v20190910.CreateStackRequest;
import com.aliyuncs.ros.model.v20190910.DeleteStackRequest;
import com.aliyuncs.ros.model.v20190910.DescribeRegionsRequest;
import com.aliyuncs.ros.model.v20190910.GetStackRequest;
```

- b) 初始化SDK客户端对象。

```
private static String ACCESSKEYID = "<yourAccessKeyId>";
private static String SECRET = "<yourAccessKeySecret>";
private static String REGIONID = "<yourRegionId>";
private static DefaultAcsClient client = new DefaultAcsClient(DefaultProfile.
getProfile(REGIONID, ACCESSKEYID, SECRET));
```

```
private static int connectTimeout = 10000;
private static int readTimeout = 10000;
private static Long timeoutInMinutes = 30L;
```

查询可用地域列表

您可以使用Java SDK查询可用地域列表。

```
public String describeRegion() {
    DescribeRegionsRequest request = new DescribeRegionsRequest();
    request.setConnectTimeout(connectTimeout);
    request.setReadTimeout(readTimeout);
    request.setAcceptFormat(FormatType.JSON);
    HttpResponse response = null;
    String responseContent = null;
    try {
        response = client.doAction(request);
        responseContent = response.getHttpContentString();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return responseContent;
}
```

创建资源栈

创建资源栈时，您必须指定以下参数：

- **StackName**：将要创建的资源栈的名称。每个用户空间下的资源栈名称不能重复。
- **TimeoutInMinutes**：资源栈创建时长，单位为分钟。如果在指定时间内不能完成创建，则超时失败。
- **TemplateBody**：创建的资源栈使用的模板内容。
- **TemplateURL**：模板主体的文件的位置。必须指定**TemplateBody**或**TemplateURL**，但不能同时指定两者。
- **Parameters**：创建的资源栈所需的参数。需要在模板中定义Key。

```
String stackName = "MyStack";
String templateBody = "{\n" +
    "\"ROSTemplateFormatVersion\": \"2015-09-01\",\n" +
    "\"Parameters\": {\n" +
    "  \"VpcName\": {\n" +
    "    \"Type\": \"String\",\n" +
    "    \"Description\": \"Vpc Name\",\n" +
    "    \"Label\": \"Vpc Name\"\n" +
    "  },\n" +
    "  \"CidrBlock\": {\n" +
    "    \"Type\": \"String\",\n" +
    "    \"Description\": \"Vpc CidrBlock\",\n" +
    "    \"Label\": \"Vpc CidrBlock\"\n" +
    "  }\n" +
    "},\n" +
    "\"Resources\": {\n" +
    "  \"Vpc\": {\n" +
    "    \"Type\": \"ALIYUN::ECS::VPC\",\n" +
    "    \"Properties\": {\n" +
```

```

    "    \"CidrBlock\": {\n" +
    "        \"Ref\": \"CidrBlock\"\n" +
    "    },\n" +
    "    \"VpcName\": {\n" +
    "        \"Ref\": \"VpcName\"\n" +
    "    }\n" +
    "  }\n" +
    "  }\n" +
    "  }\n" +
    "  }\n" +
    "};

```

```

List<CreateStackRequest.Parameters> parametersss = new ArrayList<CreateStackRequest.
Parameters>();
CreateStackRequest.Parameters cidrBlock = new CreateStackRequest.Parameters();
cidrBlock.setParameterKey("CidrBlock");
cidrBlock.setParameterValue("192.168.0.0/16");
CreateStackRequest.Parameters vpcName = new CreateStackRequest.Parameters();
vpcName.setParameterKey("VpcName");
vpcName.setParameterValue("TestVpc");
parametersss.add(cidrBlock);
parametersss.add(vpcName);

```

```

public String createStack(String stackName, String templateBody, List<CreateStackRequest.Parameters> parametersss) {
    CreateStackRequest request = new CreateStackRequest();
    request.setConnectTimeout(connectTimeout);
    request.setReadTimeout(readTimeout);
    request.setTimeoutInMinutes(timeoutInMinutes);
    request.setStackName(stackName);
    // 若模板较大, 建议使用TemplateURL参数, 避免URL过长, 调用失败。
    // 您也可以继承CreateStackRequest类, 通过调用putBodyParameter (protected方法) 自行实现setTemplateBody方法。
    // request.putBodyParameter("TemplateBody", templateBody)
    request.setTemplateBody(templateBody);
    request.setParametersss(parametersss);
    HttpResponse response = null;
    String responseContent = null;
    try {
        response = client.doAction(request);
        responseContent = response.getHttpContentString();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return responseContent;
}

```

查询资源栈

输入目标资源栈的ID。

```

public String getStack(String stackId) {
    GetStackRequest request = new GetStackRequest();
    request.setConnectTimeout(connectTimeout);
    request.setReadTimeout(readTimeout);
    request.setStackId(stackId);
    request.setAcceptFormat(FormatType.JSON);
    HttpResponse response = null;
    String responseContent = null;
    try {
        response = client.doAction(request);
        responseContent = response.getHttpContentString();
    } catch (ClientException e) {
        e.printStackTrace();
    }
}

```

```
    }  
    return responseContent;  
}
```

删除资源栈

输入目标资源栈的ID。

```
public String deleteStack(String stackId) {  
    DeleteStackRequest request = new DeleteStackRequest();  
    request.setConnectTimeout(connectTimeout);  
    request.setReadTimeout(readTimeout);  
    request.setStackId(stackId);  
    request.setAcceptFormat(FormatType.JSON);  
    HttpResponse response = null;  
    String responseContent = null;  
    try {  
        response = client.doAction(request);  
        responseContent = response.getHttpContentString();  
    } catch (ClientException e) {  
        e.printStackTrace();  
    }  
    return responseContent;  
}
```

```
package com;  
  
import com.aliyuncs.DefaultAcsClient;  
import com.aliyuncs.exceptions.ClientException;  
import com.aliyuncs.http.FormatType;  
import com.aliyuncs.http.HttpResponse;  
import com.aliyuncs.profile.DefaultProfile;  
import com.aliyuncs.ros.model.v20190910.CreateStackRequest;  
import com.aliyuncs.ros.model.v20190910.DeleteStackRequest;  
import com.aliyuncs.ros.model.v20190910.DescribeRegionsRequest;  
import com.aliyuncs.ros.model.v20190910.GetStackRequest;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Stack {  
    private static String ACCESSKEYID = "<yourAccessKeyId>";  
    private static String SECRET = "<yourAccessKeySecret>";  
    private static String REGIONID = "<yourRegionId>";  
    private static DefaultAcsClient client = new DefaultAcsClient(DefaultProfile.  
        getProfile(REGIONID, ACCESSKEYID, SECRET));  
    private static int connectTimeout = 10000;  
    private static int readTimeout = 10000;  
    private static Long timeoutInMinutes = 30L;  
  
    public String describeRegion() {  
        DescribeRegionsRequest request = new DescribeRegionsRequest();  
        request.setConnectTimeout(connectTimeout);  
        request.setReadTimeout(readTimeout);  
        request.setAcceptFormat(FormatType.JSON);  
        HttpResponse response = null;  
        String responseContent = null;  
        try {  
            response = client.doAction(request);  
            responseContent = response.getHttpContentString();  
        } catch (ClientException e) {
```

```
        e.printStackTrace();
    }
    return responseContent;
}

public String createStack(String stackName, String templateBody, List<
CreateStackRequest.Parameters> parameterss) {
    CreateStackRequest request = new CreateStackRequest();
    request.setConnectTimeout(connectTimeout);
    request.setReadTimeout(readTimeout);
    request.setTimeoutInMinutes(timeoutInMinutes);
    request.setStackName(stackName);
    // 若模板较大, 建议使用TemplateURL参数, 避免URL过长, 调用失败。
    // 您也可以继承CreateStackRequest类, 通过调用putBodyParameter (
protected方法) 自行实现setTemplateBody方法。
    // request.putBodyParameter("TemplateBody", templateBody)
    request.setTemplateBody(templateBody);
    request.setParameterss(parameterss);

    HttpResponse response = null;
    String responseContent = null;
    try {
        response = client.doAction(request);
        responseContent = response.getHttpContentString();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return responseContent;
}

public String getStack(String stackId) {
    GetStackRequest request = new GetStackRequest();
    request.setConnectTimeout(connectTimeout);
    request.setReadTimeout(readTimeout);
    request.setStackId(stackId);
    request.setAcceptFormat(FormatType.JSON);
    HttpResponse response = null;
    String responseContent = null;
    try {
        response = client.doAction(request);
        responseContent = response.getHttpContentString();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return responseContent;
}

public String deleteStack(String stackId) {
    DeleteStackRequest request = new DeleteStackRequest();
    request.setConnectTimeout(connectTimeout);
    request.setReadTimeout(readTimeout);
    request.setStackId(stackId);
    request.setAcceptFormat(FormatType.JSON);
    HttpResponse response = null;
    String responseContent = null;
    try {
        response = client.doAction(request);
        responseContent = response.getHttpContentString();
    } catch (ClientException e) {
        e.printStackTrace();
    }
    return responseContent;
}
```

```

public static void main(String[] args) throws ClassNotFoundException {
    StackDemo stack = new StackDemo();
    String stackName = "MyStack";
    String templateBody = "{\n" +
        "  \"ROSTemplateFormatVersion\": \"2015-09-01\",\n" +
        "  \"Parameters\": {\n" +
        "    \"VpcName\": {\n" +
        "      \"Type\": \"String\",\n" +
        "      \"Description\": \"Vpc Name\",\n" +
        "      \"Label\": \"Vpc Name\"\n" +
        "    },\n" +
        "    \"CidrBlock\": {\n" +
        "      \"Type\": \"String\",\n" +
        "      \"Description\": \"Vpc CidrBlock\",\n" +
        "      \"Label\": \"Vpc CidrBlock\"\n" +
        "    },\n" +
        "  },\n" +
        "  \"Resources\": {\n" +
        "    \"Vpc\": {\n" +
        "      \"Type\": \"ALIYUN::ECS::VPC\",\n" +
        "      \"Properties\": {\n" +
        "        \"CidrBlock\": {\n" +
        "          \"Ref\": \"CidrBlock\"\n" +
        "        },\n" +
        "        \"VpcName\": {\n" +
        "          \"Ref\": \"VpcName\"\n" +
        "        }\n" +
        "      }\n" +
        "    },\n" +
        "  },\n" +
        "  }\n" +
        "};";
    List<CreateStackRequest.Parameters> parametersss = new ArrayList<
CreateStackRequest.Parameters>();
    CreateStackRequest.Parameters cidrBlock = new CreateStackRequest.
Parameters();
    cidrBlock.setParameterKey("CidrBlock");
    cidrBlock.setParameterValue("192.168.0.0/16");
    CreateStackRequest.Parameters vpcName = new CreateStackRequest.
Parameters();
    vpcName.setParameterKey("VpcName");
    vpcName.setParameterValue("TestVpc");
    parametersss.add(cidrBlock);
    parametersss.add(vpcName);
    stack.createStack(stackName, templateBody, parametersss);
    stack.describeRegion();
    stack.getStack("<yourStackId>");
    stack.deleteStack("<yourStackId>");
  }
}

```