

ALIBABA CLOUD

阿里云

逻辑编排
用户指南

文档版本：20220628

 阿里云

法律声明

阿里云提醒您在使用或阅读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. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，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
 危险	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。	 危险 重置操作将丢失用户配置数据。
 警告	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。
 注意	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。
 说明	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	命令或代码。	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</code> <i>Instance_ID</i>
[] 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

1.通过RAM授权子账号编排 workflow	05
2.通过AccessKey执行 workflow	10
3.使用RAM授权	15
4.自定义授权策略	17
5.可用地域	19
6. workflow 定义语法	20
6.1. 基本介绍	20
7.逻辑编排支持被审计的事件说明	23

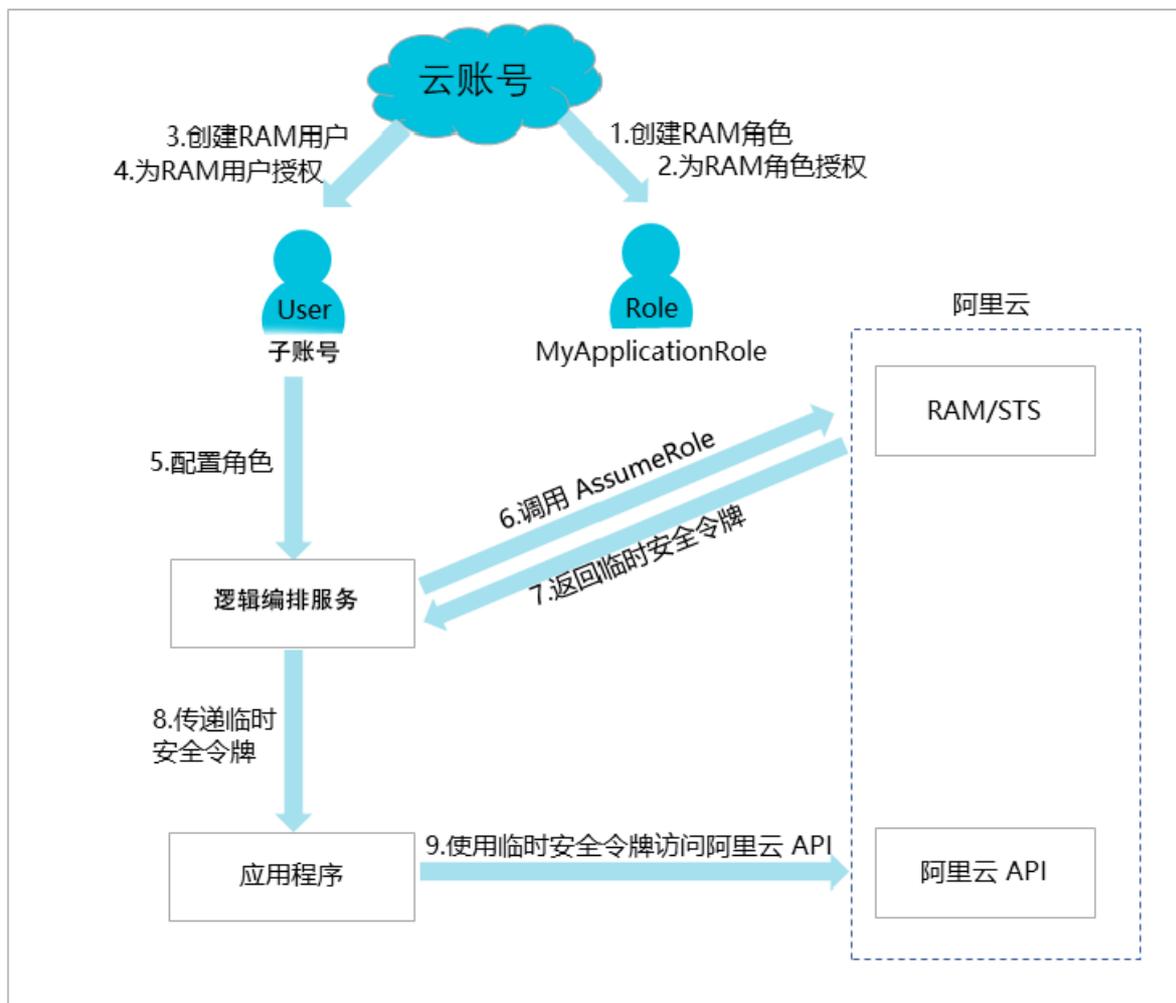
1.通过RAM授权子账号编排 workflow

本文介绍了如何使用访问控制 RAM（Resource Access Management）在账号级别上实现对逻辑编排访问资源的授权。

背景信息

通过逻辑编排进行 workflow 编排的过程中，可能需要使用各种云产品的 OpenAPI。对于主账号用户，用户可以通过授权 RAM 角色给逻辑编排调用相关的 OpenAPI。但是，逻辑编排服务有权访问这个 RAM 角色，并不代表执行逻辑编排的子账号，有权使用这个 RAM 角色。只有对子账号和目标 RAM 角色进行 PassRole 鉴权通过之后，该子账号才能通过逻辑编排服务，使用目标 RAM 角色。

主要的执行流程如下所示：



注意

用户仅需完成 1-5 步的操作，后续 6-9 步由被授权产品完成。

操作步骤（子账号没有 RAM 读写权限）

1. 创建 RAM 角色。在 RAM 控制台的 **RAM角色管理** 页面可以查看您当前拥有的角色，这里可以创建一个名为 `AliyunLogicComposerDefaultRole` 的角色，如果您没有创建过角色，创建方法请参见 [创建可信实体为阿里云服务的RAM角色](#)。

 注意

这里创建的角色名称为 `AliyunLogicComposerDefaultRole`，受信服务请选择 **逻辑编排**。

创建 RAM 角色 ×

1 选择类型2 配置角色3 创建完成

选择可信实体类型
阿里云服务

角色类型
 普通服务角色 服务关联角色

* 角色名称

不超过64个字符，允许英文字母、数字，或“-”

备注

* 选择受信服务

2. 为创建的角色添加权限。在 RAM 控制台的 **权限策略管理** 页面可以查看您当前的权限策略，这里需要创建自定义权限策略。如果您没有创建过自定义策略，创建方法请参见 [创建自定义策略](#)。这里具体的权限策略可参考以下示例，这里权限应该根据实际需要进行填写。

```
{
  "Statement": [
    {
      "Action": [
        "*:TagResources",
        "oss:PutBucketTags"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ],
  "Version": "1"
}
```

接着给第一步创建的角色添加刚刚创建的权限。在 RAM 控制台的 **RAM角色管理** 中选择第一步创建的角色，授权方法请参见 [为RAM用户授权](#)。注意这里需要选择 **自定义权限策略**，选择之前创建的权限。

添加权限

被授权主体

ALiyunLogicComposerDefaultRole@role.ali-inner.onaliyunservice.com

选择权限

自定义权限策略 请输入

已选择 (1) 清除

权限策略名称	备注
ghtest1	
logic-composer-bss-query	
xreplay-accessing	
for-config	

for-config

3. 使用主账号创建一个子账号，如果已经有子账号可以直接跳过。关于如何创建 RAM 用户，请参见 [创建RAM用户](#)。创建完成后，授予子账号 **逻辑编排** 的权限，请参见 [使用RAM授权](#)。

4. 在第二步中我们为逻辑编排创建好了 **AliyunLogicComposerDefaultRole** 角色。接下来需要将该角色授予给第三步创建的子账号，首先创建一条新的 **权限策略**。方法与第二步创建权限方式一样，这里权限策略可参考如下示例：

```
{
  "Statement": [
    {
      "Action": "ram:PassRole",
      "Resource": "acs:ram::<parent uid>:role/aliyunlogiccomposerdefaultrole",
      "Effect": "Allow",
      "Condition": {
        "StringEquals": {
          "acs:Service": "composer.aliyuncs.com"
        }
      }
    }
  ],
  "Version": "1"
}
```

注意

这里 **Resource** 可以从角色中的 **ARN** 中获取。

RAM访问控制 / RAM角色管理 / AliyunLogicComposerDefaultRole

← AliyunLogicComposerDefaultRole

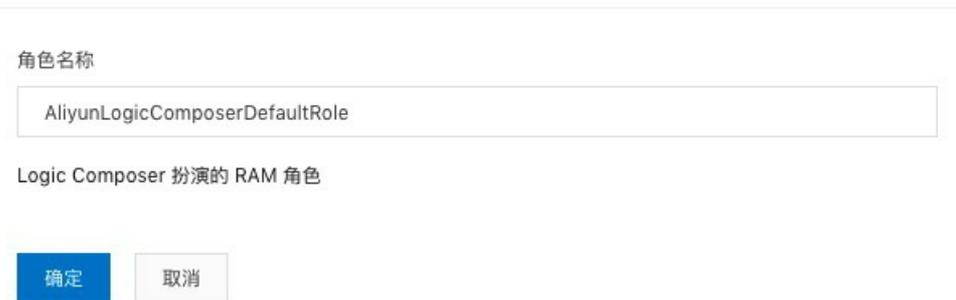
基本信息

RAM角色名称	AliyunLogicComposerDefaultRole	创建时间	2020年3月17日 11:42:08
备注		ARN	acs:ram::1833475365495825:role/aliyunlogiccomposerdefaultrole

5. 给用户授予这个权限策略，创建方法请参见 [为RAM用户授权](#)。

6. 授权完毕，可以将 **角色名** 交给子账号用户，子账号此时可以前往 **逻辑编排** 进行 workflow 编辑，并在调用 OpenAPI 时给 **逻辑编排** 授权。

6.1 如果子账号没有 **RAM 只读权限**，子账号在授权时将该角色填入授权框内，点击“确定”完成授权操作。



6.2 如果子账号有 **RAM 只读权限**，子账号可以通过可视化的方式选择合适的角色。



操作步骤（子账号有 RAM 读写权限）

1. 根据授权面板提示，用户可以选择最合适的 **权限策略** 进行授权。

角色类型

选择已有角色 创建新角色

角色名称

ALiyunLogicComposerDefaultRole

已授权策略

当前角色下没有权限策略，请在下方新增对应的授权策略

新增授权策略

请选择授权策略

- AliyunECSFullAccess **推荐**
管理云服务器服务(ECS)的权限
- AdministratorAccess
管理所有阿里云资源的权限

2.通过AccessKey执行 workflow

使用 AccessKey 鉴权方式执行 workflow，可以通过 SDK 调用执行，为您提供更安全的服务。

SDK逻辑编排目前提供了两种客户端调用方式，如果无需鉴权，可以直接调用 endpoint 执行 workflow；如果选择的是 AccessKey 鉴权方式，需要配合 SDK 调用执行，可以通过以下步骤执行。

步骤

1. 获取 Path

当收到 HTTP 请求时
...

Path ?

/resourcegroups/default/workflows/lc-bp16xkhfnvj9xl/triggers/manual/invoke
复制

* Method ?

GET
▼

* 鉴权方式 ?

AccessKey 鉴权
▼

? [查看 AccessKey 鉴权使用方式](#)

修改请求参数

参数名	标题	类型	描述	必填
没有请求参数				

修改请求正文

参数名	标题	类型	描述	必填
没有请求参数				

2. 通过 RAM 授权（使用子账号调用）

如果是使用子账号进行访问，请提前给子账号赋予权限，具体操作可参见 [使用RAM授权](#)。

3. 使用 SDK 调用

JAVA

- 安装依赖

将 Maven 依赖项添加到 pom.xml 中

```
<dependency>
  <groupId>com.aliyun</groupId>
  <artifactId>aliyun-java-sdk-core</artifactId>
  <version>4.0.3</version>
</dependency>
```

- 代码示例

```
import com.aliyuncs.CommonRequest;
import com.aliyuncs.CommonResponse;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.http.FormatType;
import com.aliyuncs.http.MethodType;
import com.aliyuncs.profile.DefaultProfile;
public class TestJavaSDK {
    public static void main(String[] args) {
        DefaultProfile profile = DefaultProfile.getProfile("default", "<accessKeyId>", "<accessSecret>");
        IAcsClient client = new DefaultAcsClient(profile);
        CommonRequest request = new CommonRequest();
        request.setProtocol(ProtocolType.HTTPS);
        request.setMethod(MethodType.POST);
        request.setDomain("lc.<regionId>.aliyuncs.com");
        request.setVersion("2019-09-25");
        request.setUriPattern("<Path>");
        // 这里根据 workflow 需求设置参数
        request.putQueryParameter("RegionId", "us-east-1");
        request.putHeadParameter("Content-Type", "application/json");
        // 这里根据 workflow 需求设置正文, 注意是 JSON 字符串
        String requestBody = "";
        request.setHttpContent(requestBody.getBytes(), "utf-8", FormatType.JSON);
        try {
            CommonResponse response = client.getCommonResponse(request);
            System.out.println(response.getData());
        } catch (ServerException e) {
            e.printStackTrace();
        } catch (ClientException e) {
            e.printStackTrace();
        }
    }
}
```

GO

- 代码示例

```
package main
import (
    "fmt"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk"
    "github.com/aliyun/alibaba-cloud-sdk-go/sdk/requests"
)
func main() {
    client, err := sdk.NewClientWithAccessKey("default", "<accessKeyId>", "<accessSecret>")
    if err != nil {
        panic(err)
    }
    request := requests.NewCommonRequest()
    request.Method = "POST"
    request.Scheme = "https" // https | http
    request.Domain = "lc.<regionId>.aliyuncs.com"
    request.Version = "2019-09-25"
    // 注意下面的 [WorkflowId] 替换为刚才复制的 ID
    request.PathPattern = "<Path>"
    request.Headers["Content-Type"] = "application/json"
    // 这里根据工作流需求设置参数
    request.QueryParams["RegionId"] = "us-east-1"
    // 这里根据工作流需求设置正文，注意是 JSON 字符串
    body := ``
    request.Content = []byte(body)
    response, err := client.ProcessCommonRequest(request)
    if err != nil {
        panic(err)
    }
    fmt.Print(response.GetHttpContentString())
}
```

Python

- 代码示例

```
#!/usr/bin/env python
#coding=utf-8
from aliynsdkcore.client import AcsClient
from aliynsdkcore.request import CommonRequest
client = AcsClient('<accessKeyId>', '<accessSecret>', 'default')
request = CommonRequest()
request.set_accept_format('json')
request.set_method('POST')
request.set_protocol_type('https') # https | http
request.set_domain('lc.<regionId>.aliyuncs.com')
request.set_version('2019-09-25')
# 注意下面的 [WorkflowId] 替换为刚才复制的 ID
request.set_uri_pattern('<Path>')
# 这里根据 workflow 需求设置参数
request.add_query_param('RegionId', "us-east-1")
request.add_header('Content-Type', 'application/json')
# 这里根据 workflow 需求设置正文, 注意是 JSON 字符串
body = ''''''
request.set_content(body.encode('utf-8'))
response = client.do_action_with_exception(request)
# python2: print(response)
print(str(response, encoding = 'utf-8'))
```

Ruby

```
# gem install aliynsdkcore
require 'aliynsdkcore'
client = ROAClient.new(
  access_key_id: '<accessKeyId>',
  access_key_secret: '<accessSecret>',
  endpoint: 'https://lc.<regionId>.aliyuncs.com',
  api_version: '2019-09-25'
)
response = client.request(
  method: 'POST',
  # 注意下面的 [WorkflowId] 替换为刚才复制的 ID
  uri: '<Path>',
  # 这里根据 workflow 需求设置参数
  queries: {
    "RegionId": "cn-hangzhou",
  },
  headers: {
    "Content-Type": "application/json"
  },
  # 这里根据 workflow 需求设置正文, 注意是 JSON 字符串
  body: '',
  options: {}
)
print response
```

Node.js

- 安装依赖

```
npm install @alicloud/pop-core
```

● 代码示例

```
'use strict'
var ROAClient = require('@alicloud/pop-core').ROAClient;
var client = new ROAClient({
  accessKeyId: '<your access key id>',
  accessKeySecret: '<your access key secret>',
  endpoint: 'https://lc.<regionId>.aliyuncs.com',
  apiVersion: '2019-09-25',
});
async function test() {
  try {
    var res = await client.request(
      'POST',
      '<Path>',
      {}, // 这里根据 workflow 需求设置参数
      '<request body>', // 这里根据 workflow 需求设置正文, 注意是 JSON 字符串
      {
        'content-type': 'application/json',
      }
    );
    console.log('invoke workflow: ', res);
  } catch (err) {
    console.error(err);
  }
}
test().then();
```

接口详细入参可以查看 [SDK 源码](#)。

逻辑编排支持的 Region

Region	RegionId	Endpoint
上海	cn-shanghai	https://lc.cn-shanghai.aliyuncs.com
美东	us-east-1	https://lc.us-east-1.aliyuncs.com

3.使用RAM授权

默认情况下，只有阿里云主账号可以操作自己创建的编排实例。如果子账号需要操作主账号创建的编排实例，则需要先为子账号进行授权。

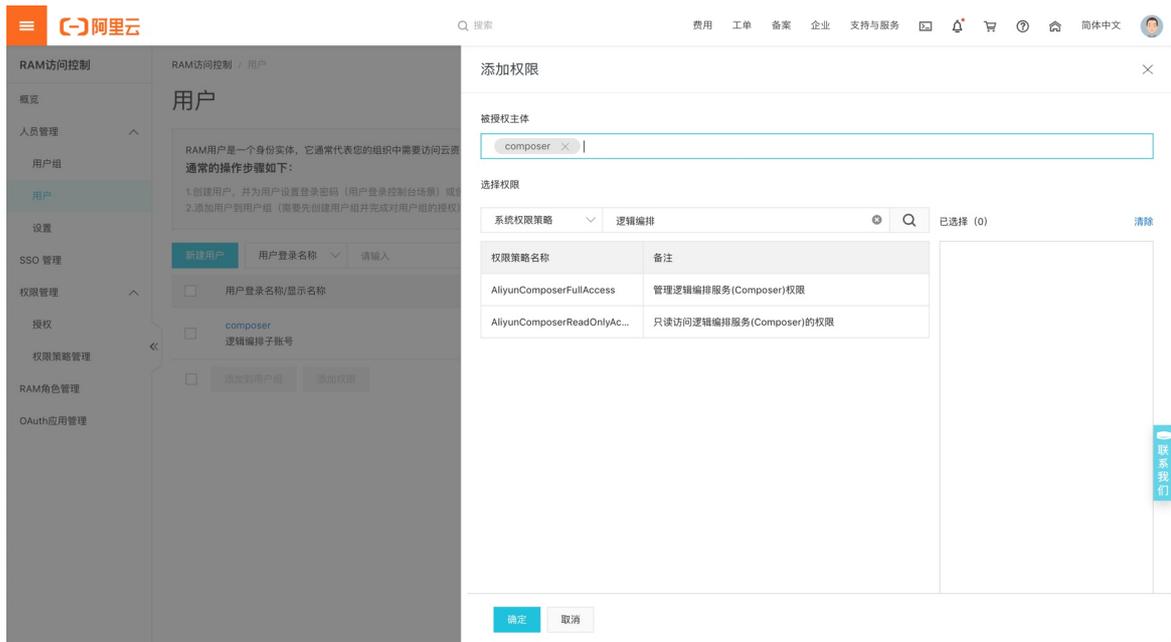
背景信息

操作步骤

1. 登录 [阿里云 RAM 控制台](#)。
2. 在 RAM 控制台的 [RAM概览](#) 页面可以查看您是否有子账号。如果您没有子账号，请先创建子账号，创建方法请参见 [创建RAM用户](#)。
3. 使用子账号登录 [阿里云 RAM 控制台](#)，单击 [人员管理 > 用户](#)，找到待授权的用户。



4. 单击待授权的用户名后面的 **添加权限**。
5. 在 **添加权限** 页面中，选择逻辑编排的系统权限策略。根据需要选择相应的授权策略到右侧待选框，单击 **确定** 完成添加。
 - 逻辑编排的读写权限，请选择 `AliyunComposerFullAccess`
 - 逻辑编排的只读权限，请选择 `AliyunComposerReadOnlyAccess`



4. 自定义授权策略

逻辑编排默认提供的系统授权策略是粗粒度的授权策略。如果这种粗粒度授权策略不能满足您的需求，您可以创建自定义授权策略。例如您想控制对某个具体的编排实例的操作权限，则必须使用自定义授权策略才能满足这种细粒度要求。

背景信息

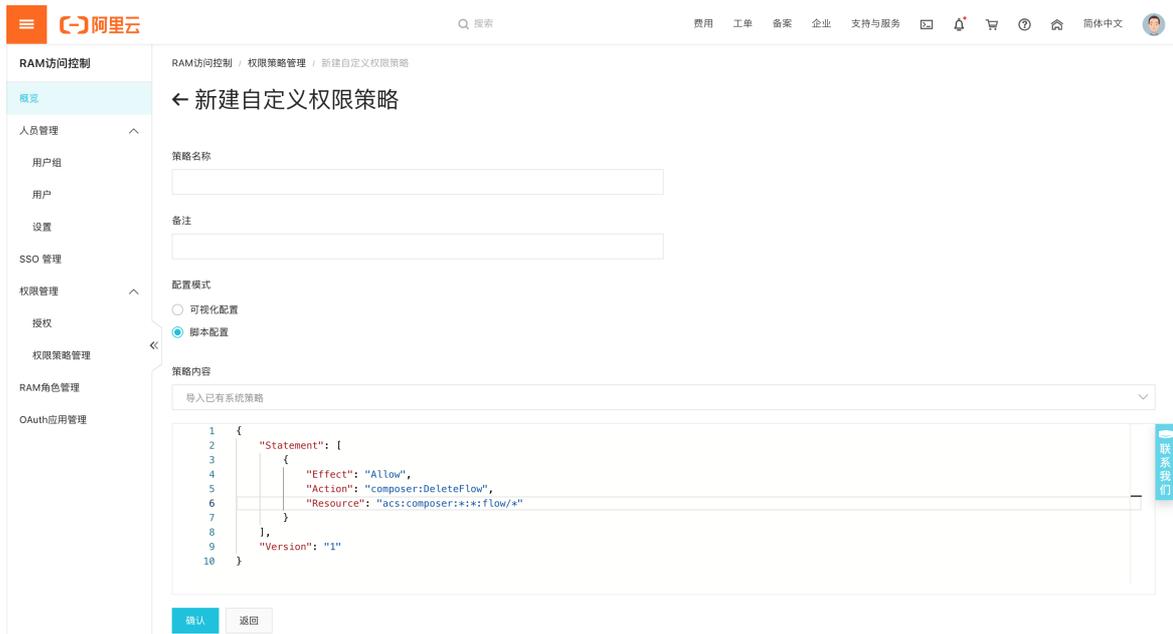
在创建自定义授权策略时，您需要了解授权策略语言的基本结构和语法，详情请参见 [权限策略语法和结构](#)。

本文以授予子账号查询、扩容和删除集群的权限为例进行说明。

操作步骤

操作步骤

1. 使用阿里云账号登录 [RAM管理控制台](#)。
2. 单击左侧导航栏中的 [权限管理 > 权限策略管理](#)。
3. 单击页面上的 [新建权限策略](#)。
4. 在 [新建自定义权限策略](#) 页面中，填写 [策略名称](#) 和 [备注](#)，配置模式选择 [脚本配置](#)，在下方编辑器中编辑您的授权策略内容。
5. 编辑完成后，单击 [确认](#)。



执行结果

权限定义列表

API	鉴权 Action	鉴权 Resource	说明
ListFlow	composer:ListFlow	acs:composer:\$regionid:\$accountid:flow/*	查询编排实例列表

API	鉴权 Action	鉴权 Resource	说明
DescribeFlow	composer:DescribeFlow	acs:composer:\$regionid:\$accountid:flow/\$flowId	查询某个编排实例详情
CreateFlow	composer:CreateFlow	acs:composer:\$regionid:\$accountid:flow/*	创建一个编排实例
DeleteFlow	composer>DeleteFlow	acs:composer:\$regionid:\$accountid:flow/\$flowId	删除某个编排实例
ModifyFlow	composer:ModifyFlow	acs:composer:\$regionid:\$accountid:flow/\$flowId	修改某个编排实例
CloneFlow	composer:CloneFlow	acs:composer:\$regionid:\$accountid:flow/*	复制某个编排实例
InvokeFlow	composer:InvokeFlow	acs:composer:\$regionid:\$accountid:flow/\$flowId	运行某个编排实例
ListFlowVersion	composer:ListFlowVersion	acs:composer:\$regionid:\$accountid:flow/\$flowId	查询某个编排实例的所有版本
DescribeFlowVersion	composer:DescribeFlowVersion	acs:composer:\$regionid:\$accountid:flow/\$flowId	查询某个编排实例的某个版本信息

5. 可用地域

地域是指物理的数据中心。阿里云的地域和可用区的详情，请参见。

目前逻辑编排只开放 华东2（上海） 和 美国（弗吉尼亚） 两个地域，后续将陆续开放其它可用地域，敬请关注。请根据您的业务所在地选择合适的地域创建您的业务 workflows。

地域	所在城市	Region ID	Endpoint
华东2	上海	cn-shanghai	composer.cn-shanghai.aliyuncs.com
美国	弗吉尼亚	us-east-1	composer.us-east-1.aliyuncs.com

6. 工作流定义语法

6.1. 基本介绍

本文为您介绍如何定义工作流。

一个工作流编排实例的逻辑定义至少包含一个触发器 (Trigger) 和需要执行的任务 (Action)。目前定义只支持 JSON 语法。最外层结构定义如下：

```
{
  "schemaVersion": "${flow-schema-version}",
  "triggers": { "${flow-trigger-definitions}" },
  "actions": { "${flow-action-definitions}" }
}
```

字段	必选	说明
schemaVersion	否	描述当前定义所使用的 Schema 的版本
triggers	否	用于实例化工作流的一个或多个触发器的定义
actions	否	要在工作流运行时执行的一个或多个任务的定义

版本 schemaVersion

描述当前定义所使用的 Schema 的版本，目前只支持 `2018-12-12` 版。

触发器 Triggers

所有逻辑应用均以触发器开始。它定义了可以实例化并启动逻辑应用工作流的调用。

```
"${trigger-name}": {
  "type": "${trigger-type}",
  "inputs": { "${trigger-inputs}" },
},
```

字段	Type	描述
<code>\${trigger-name}</code>	String	触发器的名称
type	String	触发器类型，例如 <code>request</code> 表示一个 HTTP 请求类的触发器
inputs	JSON 对象	定义触发器行为的输入

HTTP 请求触发器示例：

```

"request": {
  "type": "Request",
  "kind": "Http",
  "inputs": {
    "method": "${method-type}",
    "schema": {
      "type": "object",
      "properties": {
        "${property-name)": {
          "type": "${property-type}"
        }
      },
      "required": [ "${required-properties}" ]
    }
  }
}

```

`inputs` 里定义了该触发器接受的传入参数。

字段	Type	描述
<code>\${method-type}</code>	String	请求的方法： <code>GET</code> 、 <code>PUT</code> 、 <code>POST</code> 、 <code>PATCH</code> 、 <code>DELETE</code>
<code>schema</code>	Object	定义入参的 Schema，须为标准的 JSON Schema 格式，用于入参校验

任务 Actions

任务的概念是使用一些参数执行一个特定的连接器。

每个任务类型标准的结构如下。某些连接器提供了更多的配置项，详情请参见对应连接器的配置项。

```

"${action-name)": {
  "type": "HTTP",
  "inputs": {
    "uri": "https://example.com/api",
    "method": "GET",
    "headers": {
      "User-Agent": "Logic Composer"
    }
  },
  "runAfter": {}
}

```

字段	必选	Type	描述
<code>type</code>	是	String	该任务对应的连接器的名称

字段	必选	Type	描述
inputs	是	Int、Float、Boolean、JSON Object	定义该连接器接受的入参，需要参考对应连接器的配置项
runAfter	是	Object	在当前任务可以运行之前，必须运行过的触发器或任务的名称和结果状态

例如，在某个任务成功之后发送响应 Response：

```
"返回请求结果": {  
  "type": "Response",  
  "inputs": {  
    "body": "@body('请求接口')"  
  },  
  "runAfter": {  
    "请求接口": ["Succeeded"]  
  }  
},
```

7. 逻辑编排支持被审计的事件说明

逻辑编排支持被操作审计服务审计的事件列表。

逻辑编排已与操作审计服务集成，您可以在操作审计中查询用户操作逻辑编排产生的管控事件。操作审计支持将管控事件投递到日志服务SLS的LogStore或对象存储OSS的存储空间中，满足实时审计、问题回溯分析等需求。

操作审计记录了用户通过OpenAPI或控制台等方式操作云资源时产生的管控事件，逻辑编排支持在操作审计中查询的事件如下表所示。

OpenAPI 名称	描述
CloneFlow	克隆工作流
CreateFlow	创建工作流
DeleteFlow	删除工作流
DescribeAccountSummary	查询当前账号下的统计信息
DescribeConnectorAttribute	查询连接器属性
DescribeConnectorCapability	查询连接器支持的能力
DescribeFlow	查询工作流详情
DescribeFlowMetric	查询工作流运行指标
DescribeFlowTemplate	查询模板详情
DescribeInvocationLog	查询执行日志详情
DescribeMetricDetail	查询统计详情
DisableFlow	禁用工作流
EnableFlow	启用工作流

OpenAPI 名称	描述
GetFlow	查询 workflow 详情
GetTemplate	获取模板详情
GroupInvokeFlow	批量执行 workflow
InvokeFlow	执行 workflow
ListConnectors	查询连接器列表
ListConnectorTriggers	查询连接器触发器列表
ListFlow	查询 workflow 列表
ListFlowConnections	查询 workflow 关联的服务连接列表
ListFlows	查询 workflow 列表
ListFlowTemplate	查询 workflow 模板列表
ListFlowTriggers	查询 workflow 触发器列表
ListFlowVersion	查询 workflow 版本列表
ListInvocationLogs	查询 workflow 运行日志列表
ListTemplates	查询模板列表
ModifyFlow	修改 workflow
UpdateConnection	更新服务连接
UpdateFlow	更新 workflow 信息