

ALIBABA CLOUD

Alibaba Cloud

函数计算
SDK 示例

文档版本：20200909

 阿里云

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.SDK列表 ----- 05
- 2.Go ----- 06
 - 2.1. 设置实例类型 ----- 06
 - 2.2. 设置按量实例 ----- 07
 - 2.3. 创建Custom Container Runtime函数 ----- 09
- 3.Node.js ----- 11
 - 3.1. 设置实例并发度 ----- 11

1.SDK列表

本文列出了已支持的SDK。更多语言的SDK正在开发中，敬请期待。

语言	SDK
Node.js	fc-nodejs-sdk
Java	fc-java-sdk
Python	fc-python-sdk
Go	fc-go-sdk
PHP	fc-php-sdk
C#	fc-csharp-sdk

2.Go

2.1. 设置实例类型

本文介绍如何使用Go SDK中的示例代码设置实例类型。

您可以通过SDK在创建函数或更新函数时指定使用弹性实例还是性能实例，系统后端会将您的请求路由至设置的实例类型中。

实例类型参数为：

- e1：弹性实例
- c1：性能实例

如果不指定实例类型，则默认为弹性实例。

SDK示例

```
// Go sdk
package main

import (
    "github.com/aliyun/fc-go-sdk"
)

func main() {
    // new fc client
    client, _ := fc.NewClient("endpoint", "2016-08-15",
        "accessKeyID", "accessKeySecret")

    // CreateService
    serviceName := "TestService"
    _, err := client.CreateService(fc.NewCreateServiceInput().
        WithServiceName(serviceName))
    if err != nil {
        panic(err)
    }

    // CreateFunction
    functionName := "TestFunction"
    createFunctionInput := fc.NewCreateFunctionInput(serviceName).
        WithFunctionName(functionName).
        WithHandler("index.handler").
        WithRuntime("python2.7").
        WithCode(fc.NewCode().WithFiles("./code/index.py")).
        WithTimeout(5).
        WithInstanceType("c1") // 指定函数所属实例类型为性能实例。
    _, err = client.CreateFunction(createFunctionInput)
    if err != nil {
        panic(err)
    }

    return
}
```

2.2. 设置按量实例

本文介绍使用SDK设置函数的按量实例。

SDK示例

本文以Go语言为例，可以通过以下方法指定函数的按量实例。

```
package main

import (
    "fmt"
    fc "https://github.com/aliyun/fc-go-sdk"
)

const (
    endpoint = "your-endpoint"
    fcAPIVersion = "2016-08-15"
    AccessKeyID = "your-ak-id"
    AccessKeySecret = "your-ak-secret"
)

var (
    serviceName = "service-name"
    qualifier = "LATEST"
    functionName = "function-name"
)

func main() {
    fcClient, _ := fc.NewClient(endpoint, fcAPIVersion, AccessKeyID, AccessKeySecret)

    // set
    putInput := fc.NewPutOnDemandConfigInput(serviceName, qualifier, functionName)
    putInput.WithMaximumInstanceCount(10)
    putResp, err := fcClient.PutOnDemandConfig(putInput)
    fmt.Println("put on-demand config", putResp, err)

    // list
    listInput := fc.NewListOnDemandConfigsInput()
    listInput.WithPrefix(fmt.Sprintf("services/%s", serviceName))
    listResp, err := fcClient.ListOnDemandConfigs(listInput)
    fmt.Println("list on-demand configs", listResp, err)

    // get
    getInput := fc.NewGetOnDemandConfigInput(serviceName, qualifier, functionName)
    getResp, err := fcClient.GetOnDemandConfig(getInput)
```



```
fmt.Println("get on-demand configs", getResp, err)

// delete
deleteInput := NewDeleteOnDemandConfigInput(serviceName, qualifier, functionName)
deleteResp, err := fcClient.DeleteOnDemandConfig(deleteInput)
fmt.Println("delete on-demand configs", deleteResp, err)
}
```

2.3. 创建Custom Container Runtime函数

文本介绍使用SDK创建Custom Container Runtime函数。

SDK示例

本文以Go语言为例，可以通过以下代码示例创建Custom Container Runtime函数。

```
import (
    "fmt"

    fc "github.com/aliyun/fc-go-sdk"
)

const (
    AK_ID    = "your ak id"
    AK_SECRET = "your ak secret"
    ROLE     = "your role arn" // role with AliyunContainerRegistryReadOnlyAccess/AliyunContainerRegistryFullAccess, for example: acs:ram::123456:role/service-role
    IMAGE    = "your ACR image" // for example: registry-vpc.cn-shanghai.aliyuncs.com/fc-test/hello-world:v1
)

func main() {
    serviceName := "fc-demo-ccr"
    funcName := "test"
    client, _ := fc.NewClient("http://123456.cn-shanghai.fc.aliyuncs.com", "2016-08-15", AK_ID, AK_SECRET)

    _, err := client.CreateService(fc.NewCreateServiceInput().
        WithServiceName(serviceName).
        WithRole(ROLE).
        WithDescription("this is a test service for docker image"))
    if err != nil {
```

```
    panic(err)
}

// CreateFunction
createFunctionInput := fc.NewCreateFunctionInput(serviceName).WithFunctionName(funcName).
    WithDescription("test function").
    WithHandler("just-a-string").WithRuntime("custom-container").
    WithCustomContainerConfig(fc.NewCustomContainerConfig().WithImage(IMAGE)).
    WithTimeout(30)

_, err = client.CreateFunction(createFunctionInput)
if err != nil {
    panic(err)
}

invokeInput := fc.NewInvokeFunctionInput(serviceName, funcName).WithPayload([]byte("hello world
"))
invokeOutput, err := client.InvokeFunction(invokeInput)
if err != nil {
    panic(err)
} else {
    fmt.Printf("InvokeFunction response: %s \n", string(invokeOutput.Payload))
}
}
```

3.Node.js

3.1. 设置实例并发度

本文介绍使用SDK指定函数的实例并发数。

SDK示例

本文以Node.js SDK为例，可以通过以下的方法指定函数的实例并发度。

```
// create function
var resp = await client.createFunction(serviceName, {
  functionName: funcName,
  handler: 'counter.handler',
  memorySize: 512,
  runtime: 'nodejs10',
  code: {
    zipFile: fs.readFileSync('/tmp/counter.zip', 'base64'),
  },
  instanceConcurrency: 10,
});

// update function
var resp = await client.updateFunction(serviceName, funcName, {
  instanceConcurrency: 20,
});

// get function
var resp = await client.getFunction(serviceName, funcName);
```