

# 阿里云 应用配置管理 ACM

快速入门

文档版本：20200630

# 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。	 <b>禁止：</b> 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。	 <b>警告：</b> 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。	 <b>注意：</b> 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。	 <b>说明：</b> 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	单击 <b>设置 &gt; 网络 &gt; 设置网络类型</b> 。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在 <b>结果确认</b> 页面，单击 <b>确定</b> 。
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 开通 ACM 服务.....	1
2 通过ACM动态推送配置.....	2
3 为不同环境设置不同配置.....	9

# 1 开通 ACM 服务

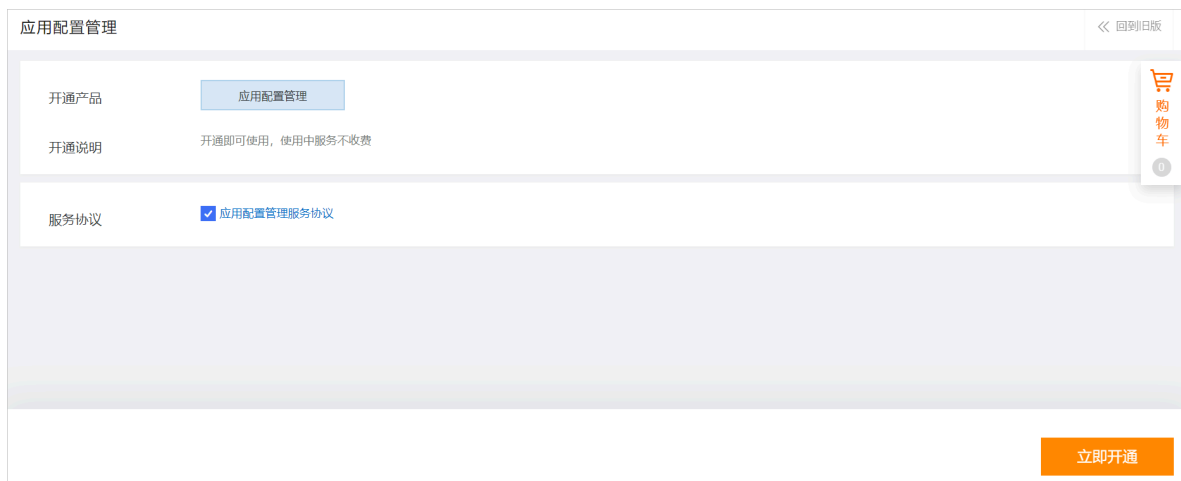
您需要先开通 ACM 服务才能使用 ACM。本文介绍如何开通 ACM 服务。

## 前提条件

您已注册阿里云账号并完成实名认证。

## 操作步骤

1. 打开 [ACM 产品主页](https://www.aliyun.com/product/acm)（<https://www.aliyun.com/product/acm>）。
2. 在页面右上角单击**登录**。
3. 在**登录**页面上输入您的阿里云账号和密码，并单击**登录**。  
登录成功后，您将跳转至 ACM 产品主页。
4. 在产品主页上单击**立即开通**，然后在**云产品开通页**页面上勾选**应用配置管理服务协议**，并单击**立即开通**。



## 2 通过ACM动态推送配置

如需为部署在多台服务器的应用更改配置，在各台服务器上逐一更改显然效率太低。您可以使用ACM集中管理配置。ACM会将更新后的配置动态推送到所有服务器。本文结合具体场景介绍此操作流程。

### 前提条件

- 开通 [ACM 服务](#)。
- 服务器上已安装JDK，并已设置环境变量JAVA\_HOME。
- 服务器上已安装Maven。关于如何安装和使用Maven，请参考[Maven官方文档](#)。

### 背景信息

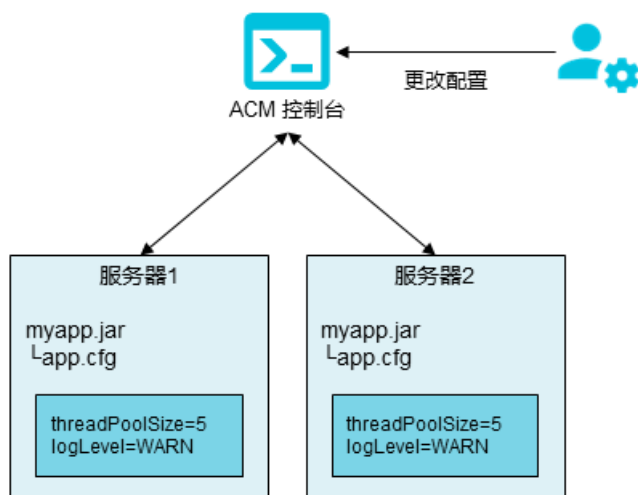
在传统的配置发布模式下，如果一个应用部署在多台服务器上，配置变更就需要再在所有服务器上逐一更改，效率低且工作繁琐。

您可以在ACM中简化此操作。首先您需要在ACM中为应用创建配置，并在应用中使用ACM的原生API监听此配置的变更。当您在ACM控制台更改此配置后，所有部署了该应用的服务器都会收到变更后的配置内容，应用状态也会随之刷新。

以下使用示例场景进行说明。

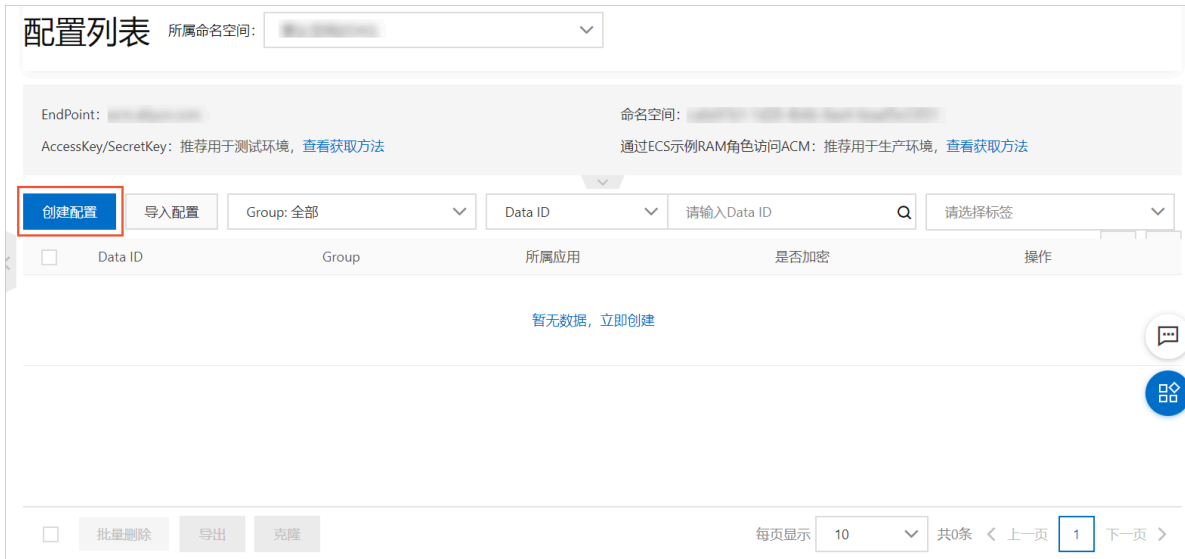
应用myapp.jar部署在生产环境的两台服务器上。该应用包含一个配置文件app.cfg，该配置文件包含线程池大小threadPoolSize和日志级别logLevel这两个配置项。现在需要同时调整该应用在上述两台服务器上的配置，并动态刷新应用的状态。

图 2-1: 示例场景：通过ACM动态推送配置



### 步骤一：在ACM中创建配置

- 1. 登录ACM控制台，在顶部菜单栏选择地域。
- 2. 在左侧导航栏选择配置列表，在页面上方选择命名空间，然后在页面左侧单击创建配置。



3. 在弹出的**创建配置**面板中填写配置信息，配置完成后单击**发布**。

← 创建配置
×

命名空间

Data ID \* ?

com.acm.myapp.app.cfg
21/236

Group \* ?

myapp
5/128

数据加密 ?

配置格式 ?

TEXT
  JSON
  XML
  YAML
  HTML
  Properties

配置内容 \* ?

```

1  threadPoolSize=5
2  logLevel=WARN

```

配置描述 ?

请输入配置描述
0/128

> 更多配置

发布

取消

参数	描述
<b>Data ID</b>	配置ID。建议采用package.class的命名规范，其中class部分是具有业务含义的配置名称，例如：com.foo.bar.log.level。Data ID在一个Group下是唯一的。
<b>Group</b>	配置分组，建议填写产品名或模块名。Group是全局唯一的。



参数	描述
数据加密	<p>您可完成如下配置来统一使用密钥管理服务（KMS）对配置进行加密：</p> <ol style="list-style-type: none"> <li>打开数据加密。</li> <li>在数据加密区域单击前往授权。</li> <li>在云资源访问授权页面选择AliyunACMAccessingKMSRole并单击同意授权。</li> <li>刷新创建配置面板，选择KMS加密方式。</li> </ol> <div style="background-color: #f0f0f0; padding: 5px;"> <p> <b>注意：</b> KMS服务API调用每位用户每个月有20000次免费额度，您在ACM上每次变更会调用2~3次API，因此免费额度可以支持您每月进行6000多次变更。当KMS API使用量超出后，将会收取0.6元/万次的费用。</p> </div>
配置格式	配置内容的数据格式。
配置内容	<p>输入配置的内容，例如：</p> <pre>threadPoolSize=5 logLevel=WARN</pre>
配置描述	配置描述信息。
更多配置	<ul style="list-style-type: none"> <li><b>应用：</b>配置归属的应用名。</li> <li><b>标签：</b>在文本框中输入标签信息，并单击标签选择器。</li> </ul>

## 步骤二：使用ACM原生API监听配置变更

1. 下载样例工程myapp.zip，或者运行以下命令来创建Maven工程。

```
mvn archetype:generate -DgroupId=com.acm.sample -DartifactId=myapp -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false
```

创建的工程结构如下：

```
myapp
|-- pom.xml
  |-- src
    |-- main
      |-- java
        |-- com
          |-- acm
            |-- sample
              |-- App.java
    |-- test
      |-- java
        |-- com
          |-- mycompany
            |-- app
```

```
`-- AppTest.java
```

2. 在pom.xml中添加对ACM Client原生API的依赖。

```
<dependencies>
  <dependency>
    <groupId>com.alibaba.edas.acm</groupId>
    <artifactId>acm-sdk</artifactId>
    <version>1.0.9</version>
  </dependency>
  <!-- 如果已有日志实现，则可去除以下依赖 -->
  <dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.1.7</version>
  </dependency>
</dependencies>
```

3. 在pom.xml中添加maven-assembly-plugin打包插件。

```
<plugin>
  <artifactId>maven-assembly-plugin</artifactId>
  <version>2.4</version>
  <configuration>
    <finalName>myapp</finalName>
    <descriptorRefs>
      <descriptorRef>jar-with-dependencies</descriptorRef>
    </descriptorRefs>
    <appendAssemblyId>>false</appendAssemblyId>
    <archive>
      <manifest>
        <mainClass>com.acm.sample.App</mainClass>
      </manifest>
    </archive>
  </configuration>
  <executions>
    <execution>
      <id>make-assembly</id>
      <phase>package</phase>
      <goals>
        <goal>single</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

4. 使用ACM Client原生API监听配置变更。



#### 说明：

以下代码中的\$endpoint、\$namespace、\$accesskey、\$secretKey等变量的值，可在ACM控制台的**配置列表**页面获取和查看。

应用配置管理 / 配置列表

## 配置列表

所属命名空间:

EndPoint:   
命名空间:

AccessKey/SecretKey: 推荐用于测试环境, [查看获取方法](#)  
通过ECS实例RAM角色访问ACM: 推荐用于生产环境, [查看获取方法](#)

```
//-- App.java
package com.acm.sample;

import java.io.IOException;
import java.io.StringReader;
import java.util.Properties;
import com.alibaba.edas.acm.listener.ConfigChangeListener;
import com.alibaba.edas.acm.ConfigService;
import com.alibaba.edas.acm.exception.ConfigException;

public class App {

    private static Properties appCfg = new Properties();

    public static void initAndWatchConfig() {
        final String dataId = "com.acm.myapp.app.cfg";
        final String group = "myapp";
        final long timeoutInMills = 3000;

        // 从命名空间详情对话框中拷贝各变量的值
        Properties properties = new Properties();
        properties.put("endpoint", "$endpoint");
        properties.put("namespace", "$namespace");

        // 如果通过AK/SK访问ACM
        properties.put("accessKey", "$accessKey");
        properties.put("secretKey", "$secretKey");

        // 如果通过ECS实例RAM角色访问ACM
        // properties.put("ramRoleName", "$ramRoleName");

        // 如果是加密配置, 则添加以下两行进行自动解密
        // properties.put("openKMSFilter", true);
        // properties.put("regionId", "$regionId");

        ConfigService.init(properties);

        // 直接获取配置内容
        try {
            String configInfo = ConfigService.getConfig(dataId, group, timeoutInMills);
            appCfg.load(new StringReader(configInfo));
        } catch (ConfigException e1) {
            e1.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }

        // 监听配置变化, 获取最新推送值
        ConfigService.addListener(dataId, group, new ConfigChangeListener() {
            public void receiveConfigInfo(String configInfo) {
                try {
                    appCfg.load(new StringReader(configInfo));
                } catch (Exception e) {
                    // process exception
                }
            }
        });
    }
}
```

```
    }
    refreshApp();
  }
});
}

public static void refreshApp() {
    System.out.println("current thread pool size: " + appCfg.getProperty("threadPool
Size"));
    System.out.println("current log level: " + appCfg.getProperty("logLevel"));
    System.out.println("");
}

public static void main(String[] args) {
    initAndWatchConfig();

    // 以下代码用于测试，作用是让主线程不退出。订阅配置是守护线程，如果主线程退出守
    // 护线程就会退出。while (true) {
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
    }
}
}
```

### 步骤三：部署并启动应用

1. 在项目根目录执行以下打包命令，将应用打包成JAR文件，并拷贝到两台服务器上。

```
mvn clean package
```

2. 在Shell中部署并启动应用。

```
${JAVA_HOME}/java -cp myapp.jar com.acm.sample.App
```

### 步骤四：在ACM控制台更改配置

1. 登录[ACM控制台](#)，在顶部菜单栏选择地域。
2. 在左侧导航栏选择**配置列表**，在**配置列表**页面选择创建的配置，然后在**操作列**单击**编辑**。
3. 在**编辑配置**页面上将**配置内容**更改为以下内容，并单击**发布**。

```
threadPoolSize=15
logLevel=DEBUG
```

4. 在**配置内容对比**对话框中查看更改前后的对比，确认无误后单击**发布**。

### 执行结果

发布配置之后，部署了myapp应用的两台服务器均收到了更新后的配置，并在Console中打印了以下日志。

```
current thread pool size: 15
current log level: DEBUG
```

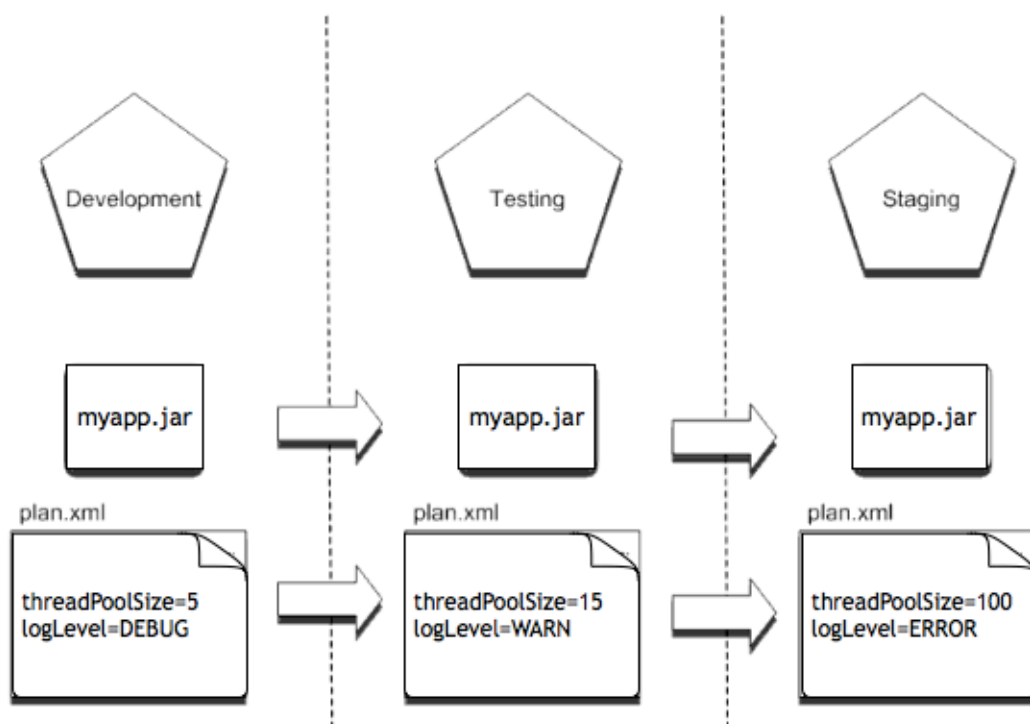
## 3 为不同环境设置不同配置

本文以一个例子介绍了如何使用ACM的命名空间（Namespace）功能，为应用在测试、预生产和生产环境下的同一个配置设置不同的值。

### 背景信息

在实际业务场景中，经常需要针对不同环境的某个配置项设置不同的值。ACM的命名空间（Namespace）功能可以帮助您实现多个环境的资源和配置的隔离。

在本示例任务中，我们将使用ACM的命名空间（Namespace）功能，为应用在测试、预生产和生产环境下的同一个配置设置不同的值。期望实现的效果如下：



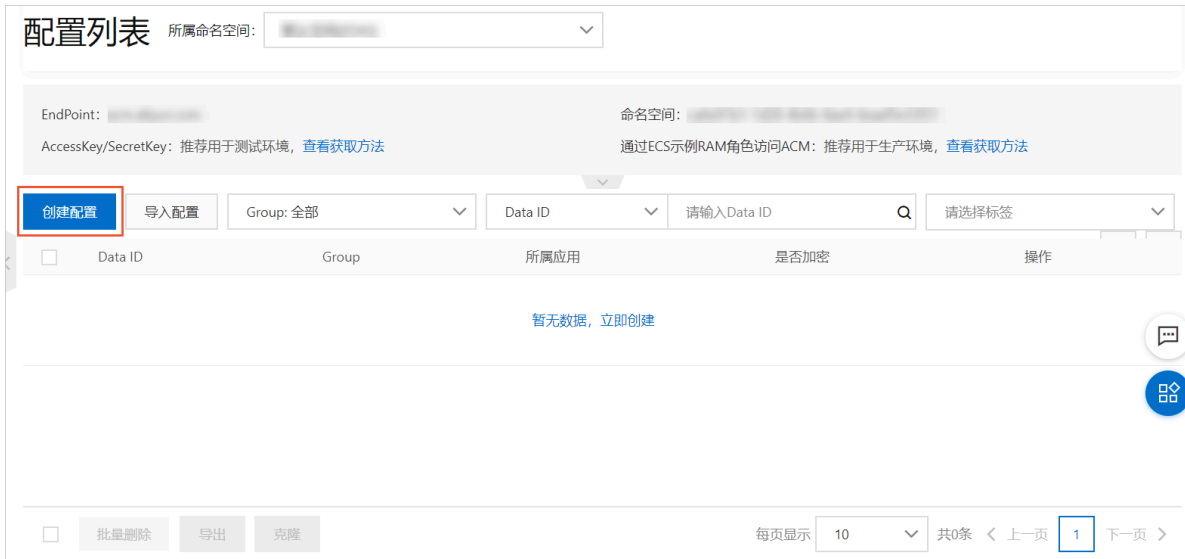
### 步骤一：在ACM上创建命名空间

下面以创建命名空间Development为例。

1. 登录[ACM控制台](#)，在顶部菜单栏选择地域。
2. 在左侧导航栏选择**命名空间**，在页面左上角单击**创建命名空间**。
3. 在弹出的**创建命名空间**面板中输入命名空间名称Development，然后单击**确定**。
4. 重复步骤2~3，继续创建Testing和Staging命名空间。

## 步骤二：在命名空间下创建配置

1. 登录ACM控制台，在顶部菜单栏选择地域。
2. 在左侧导航栏选择**配置列表**，在页面上方选择**命名空间**，然后在页面左侧单击**创建配置**。



3. 在弹出的面板中填写配置信息，配置完成后单击。

← 创建配置
×

命名空间

**Data ID \*** ?

com.acm.myapp.app.cfg
21/236

**Group \*** ?

myapp
5/128

数据加密 ?

配置格式 ?

TEXT
  JSON
  XML
  YAML
  HTML
  Properties

配置内容 ?

```

1  threadPoolSize=5
2  logLevel=WARN

```

配置描述 ?

请输入配置描述

0/128

> 更多配置

发布

取消

参数	描述
<b>Data ID</b>	配置ID。建议采用package.class的命名规范，其中class部分是具有业务含义的配置名称，例如：com.foo.bar.log.level。Data ID在一个Group下是唯一的。
<b>Group</b>	配置分组，建议填写产品名或模块名。Group是全局唯一的。

参数	描述
	<p>您可完成如下配置来统一使用密钥管理服务（KMS）对配置进行加密：</p> <ol style="list-style-type: none"> <li>打开。</li> <li>在区域单击。</li> <li>在云资源访问授权页面选择AliyunACMAccessingKMSRole并单击同意授权。</li> <li>刷新面板，选择KMS加密方式。</li> </ol> <div style="background-color: #f0f0f0; padding: 5px;">  <b>注意：</b>            KMS服务API调用每位用户每个月有20000次免费额度，您在ACM上每次变更会调用2~3次API，因此免费额度可以支持您每月进行6000多次变更。当KMS API使用量超出后，将会收取0.6元/万次的费用。         </div>
	配置内容的数据格式。
	<p>输入配置的内容，例如：</p> <div style="background-color: #f0f0f0; padding: 5px;"> <pre>threadPoolSize=5 logLevel=WARN</pre> </div>
	配置描述信息。
	<ul style="list-style-type: none"> <li>：配置归属的应用名。</li> <li><b>标签：</b>在文本框中输入标签信息，并单击标签选择器。</li> </ul>

4. 重复上述步骤2~3，为不同的命名空间创建不同的配置。