应用配置管理 ACM

快速入门

快速入门

快速入门

本文通过两个快速开始任务，帮助您快速了解 ACM 产品最基本的概念和功能：

- 创建并动态调整配置项
- 管理不同环境的配置

从开通 ACM 服务到完成基本任务的流程如下：

开通 ACM 服务

在使用 ACM 前请先按以下步骤开通服务:

登录阿里云官网。

将鼠标依次移动到**产品 > 云计算基础服务 > 互联网中间件**，然后单击**应用配置管理 ACM**，进入产品主页。

在产品主页上单击**立即开通**，根据提示完成服务开通。

如果您已经开通服务，请直接登录 ACM 控制台。

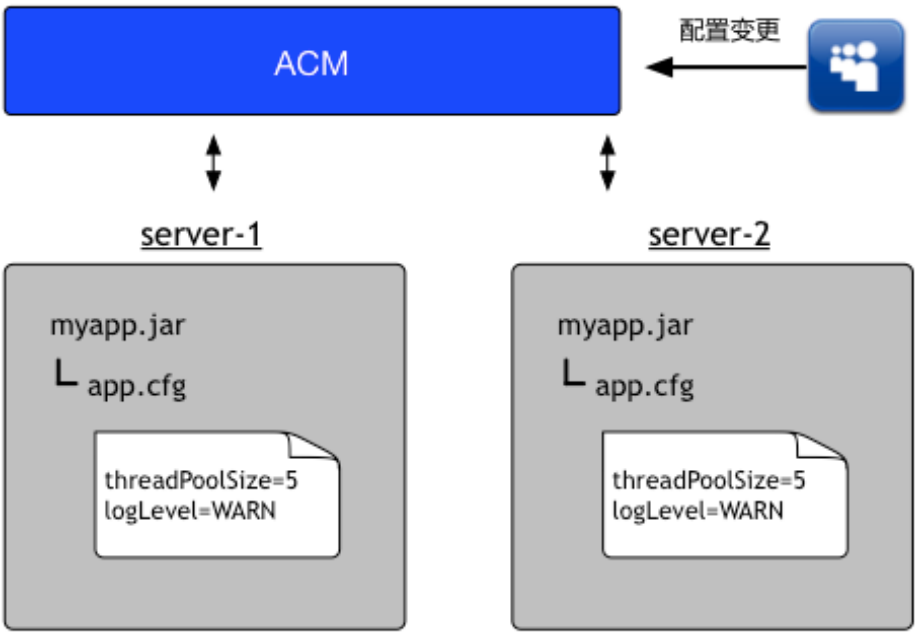
任务一：创建并动态调整配置值

场景介绍

业务应用 myapp.jar 被部署到 2 台生产环境的服务器上。该应用包含一个配置文件 app.cfg，配置文件里包含

线程池大小和日志级别两个配置项。现在需要同时调整应用在两台机器上的配置，并动态刷新应用的状态。

场景如下图:



配置内容如下:

```
## app.cfg ##
threadPoolSize=5
logLevel=WARN
```

操作步骤

STEP 1：在 ACM 中创建配置

登录 ACM 控制台。

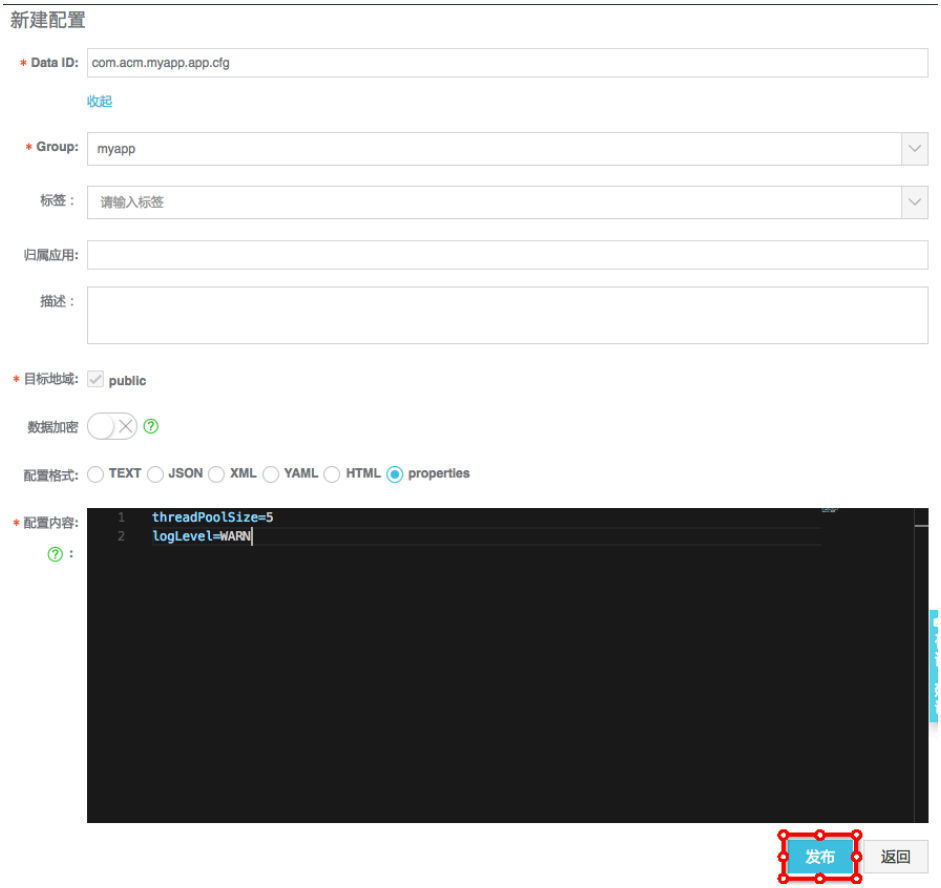
在左侧导航栏选择**配置管理**，单击右上角**新建配置**。

在**新建配置**页面输入以下内容。

- DataID： com.acm.myapp.app.cfg
- Group： myapp
- Content：

```
threadPoolSize=5
logLevel=WARN
```

如图所示：



STEP 2: 创建 maven 项目工程

创建 Maven 工程或者下载工程 myapp.tar。

关于如何安装和使用 Maven，请参考 Maven 官方文档。

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

通过上述命令，我们创建了如下的工程结构：

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

在 Pom 中添加 ACM Client Native API 依赖。

```
 <dependencies>
<dependency>
<groupId>com.alibaba.edas.acm</groupId>
<artifactId>acm-sdk</artifactId>
<version>1.0.6</version>
</dependency>
<!-- 有日志实现，下面可去掉 -->
<dependency>
<groupId>ch.qos.logback</groupId>
<artifactId>logback-classic</artifactId>
<version>1.1.7</version>
</dependency>
</dependencies>
```

使用 API 监听配置变更。

```
    //-- 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;

// 从控制台命名空间管理中拷贝对应值
Properties properties = new Properties();
properties.put("endpoint", "$endpoint");
properties.put("namespace", "$namespace");
properties.put("accessKey", "$accessKey");
properties.put("secretKey", "$secretKey");
// 如果是加密配置，则添加下面两行进行自动解密
// properties.put("openKMSFilter", true);
// properties.put("regionId", "$regionId");

ConfigService.init(properties);

// 启动只用一次场景，直接get获取配置值
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("threadPoolSize"));
System.out.println("current log level: " + appCfg.getProperty("logLevel"));
System.out.println("");
}

public static void main(String[] args) {

initAndWatchConfig();

// 让主线程不退出
while (true) {
try {
```

```
Thread.sleep(1000);
} catch (InterruptedException e) {
}
}

}

}
```

STEP 3：部署并启动应用

将 Jar 包拷贝到两台或者一台服务器上部署并启动应用。

可以通过在 shell 中执行以下命令行完成部署:

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

为了运行 Java 程序，您需要在机器上安装 JDK 并设置环境变量 JAVA_HOME。

STEP 4：在 ACM 控制台查询并变更配置

登录 ACM 控制台。

搜索 STEP1 中创建的配置。

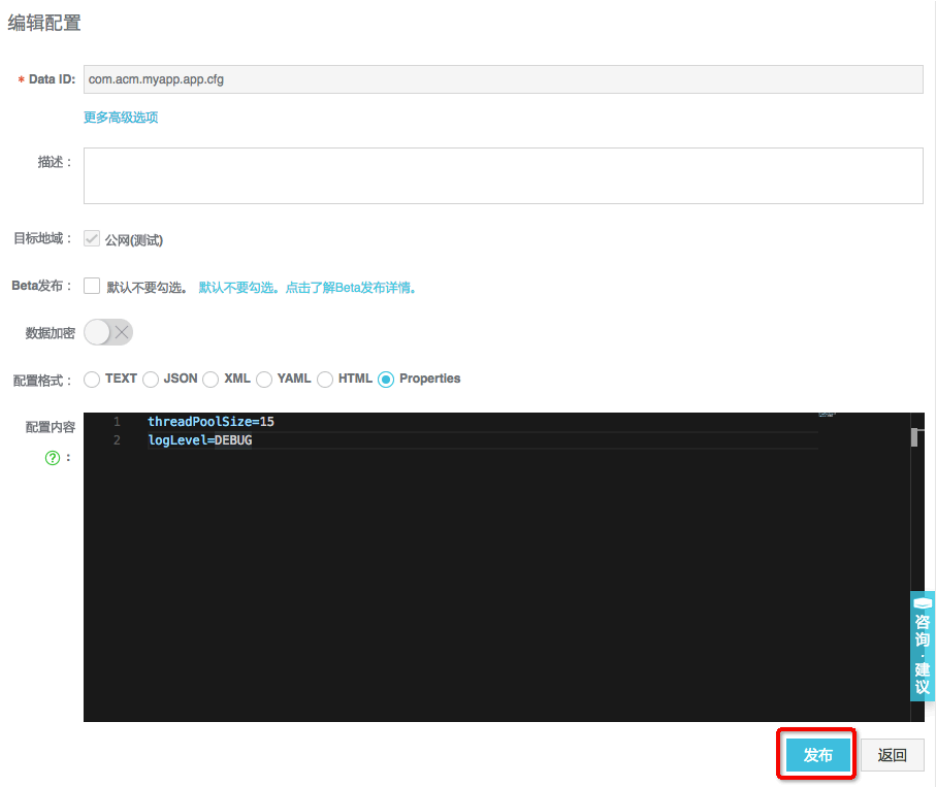
单击**详情**查看配置详情。

编辑配置内容。

调整配置项内容为：

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

如图:



单击**发布**。

STEP 5：检查结果

发布配置之后，可以看到在两台部署的机器上，应用同时收到配置变更信息，打印了如下信息。

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

总结

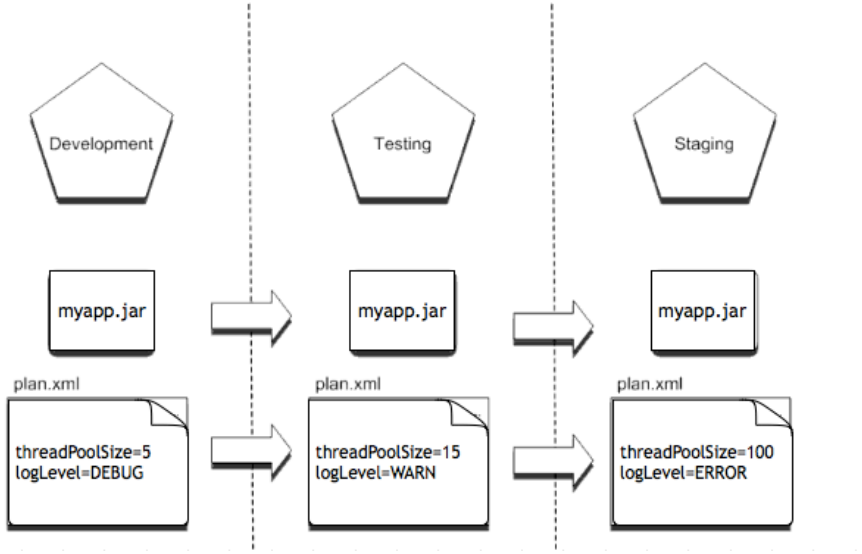
在**快速开始任务一**中，我们为应用 myapp 在 ACM 上创建了一个配置，同时在程序中使用 ACM 的 Native API 监听这个配置的变更。当我们在 ACM Console 上做了配置变更后，所有部署应用的服务器上会收到变更的配置内容，应用状态也随之刷新。

任务二：在不同的环境中设置不同的配置

场景介绍

在该任务中，我们将使用 ACM 提供的 Namespace 功能为应用的同一个配置在测试环境，预生产和生产环境设置不同的值。

如图:



操作步骤

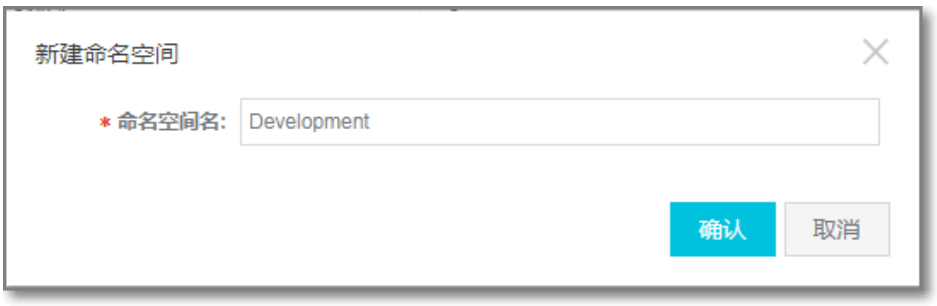
STEP 1： 在 ACM 上创建命名空间

下面以创建命名空间 Development 为例。

登录 ACM 控制台。

在左侧导航栏选择 **命名空间**，单击右上角的**添加**按钮:

在**新建命名空间**对话框中输入命名空间名 **Development**。



按 1-3 同样步骤继续创建 Testing，Staging 命名空间。

STEP 2：在命名空间下创建配置

在配置管理页面，选择命名空间 **Development**。

采用与**任务一：STEP 1** 同样的步骤创建同名配置。

总结

在任务二中我们完成了典型的多环境配置管理。在实际的业务场景中，经常需要针对不同的环境为同一个配置项设置不同的配置值。在 ACM 中可以通过 Namespace 的功能来方便的实现多环境配置管理。