

# Alibaba Cloud Application Configuration Management

Quick Start

Issue: 20190111

# Legal disclaimer

---

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectu

al property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

- 6.** Please contact Alibaba Cloud directly if you discover any errors in this document.



# Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 <b>Danger:</b> Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 <b>Warning:</b> Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 <b>Notice:</b> Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 <b>Note:</b> You can use <b>Ctrl + A</b> to select all files.
>	Multi-level menu cascade.	<b>Settings &gt; Network &gt; Set network type</b>
<b>Bold</b>	It is used for buttons, menus, page names, and other UI elements.	Click <b>OK</b> .
Courier font	It is used for commands.	Run the <code>cd /d C:/windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	It indicates that it is an optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	It indicates that it is a required value, and only one item can be selected.	<code>swich {stand   slave}</code>

# Contents

---

<b>Legal disclaimer.....</b>	<b>I</b>
<b>Generic conventions.....</b>	<b>I</b>
<b>1 Activate ACM.....</b>	<b>1</b>
<b>2 Create and dynamically adjust configuration values.....</b>	<b>2</b>
<b>3 Make different settings for a configuration in different environments.....</b>	<b>10</b>

# 1 Activate ACM

---

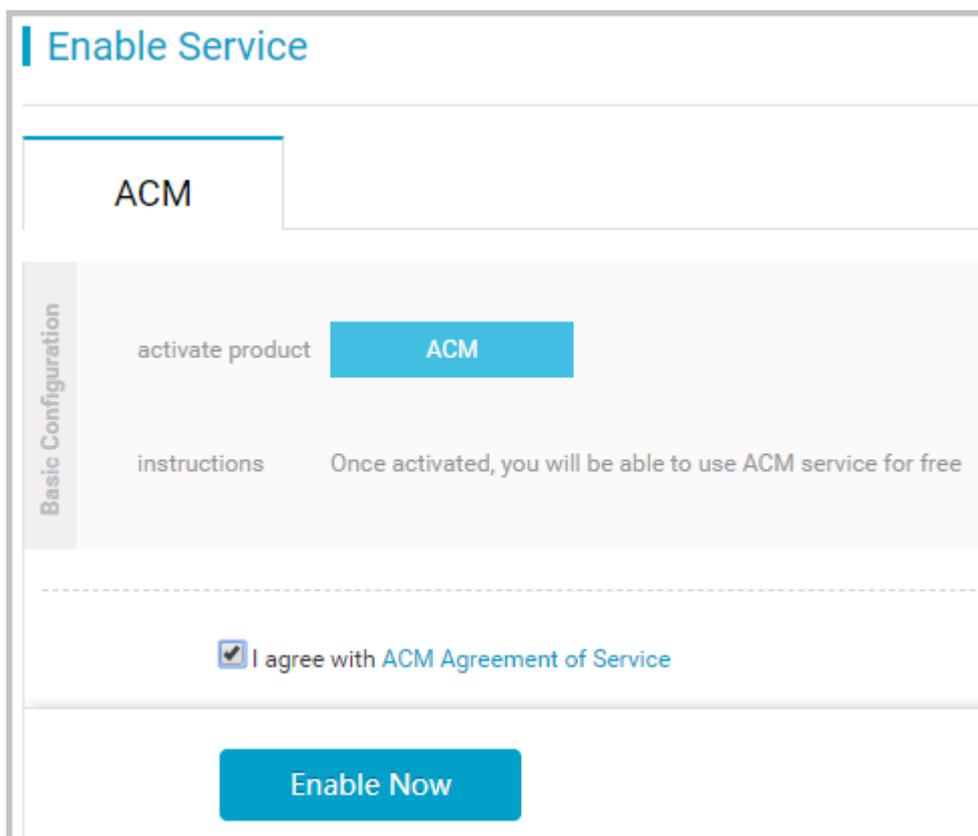
You must activate ACM service before you can use ACM. This topic explains how to activate ACM service.

## Prerequisites

You have registered an Alibaba Cloud account and completed authentication.

## Procedure

1. Open [ACM product homepage](https://www.alibabacloud.com/product/acm) (<https://www.alibabacloud.com/product/acm>).
2. In the upper-right corner of the page, click **Log In**.  
The **Log In** page is displayed.
3. Enter your Alibaba Cloud username and password on this page, and click **Sign In**.  
Once you sign in successfully, you are redirected to ACM product homepage.
4. On the product homepage, click **Get it Free**, and then on the **Enable Service** page, select **I agree with ACM Agreement of Service**, and click **Enable Now**.



The ACM console is displayed.

## 2 Create and dynamically adjust configuration values

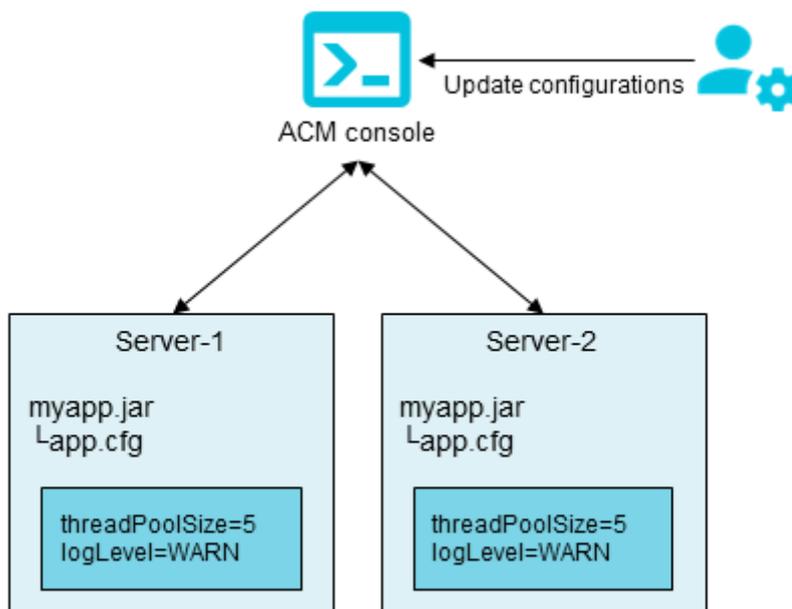
If an application is deployed on multiple servers, once you need to change the configuration, you'll have to make the same changes on all servers, which apparently is inefficient. With ACM, you can create a configuration for your application, and use the native API of ACM to listen for changes to this configuration in the program. Once you change the configuration in the ACM console, every server to which this application is deployed receives the changed configurations, and the application status changes accordingly.

### Prerequisites

- You finished the following task: [#unique\\_5](#).
- [JDK](#) has been installed on the server, and the environment variable `JAVA_HOME` has been set.

### Context

The business application `myapp.jar` is deployed to two servers in the production environment. This application has a configuration file `app.cfg`, which contains two configuration items: `threadPoolSize` and `logLevel`. Now, you need to adjust the configuration of the application on these two servers simultaneously and refresh the status of the application dynamically. The scenario is shown in the following figure:



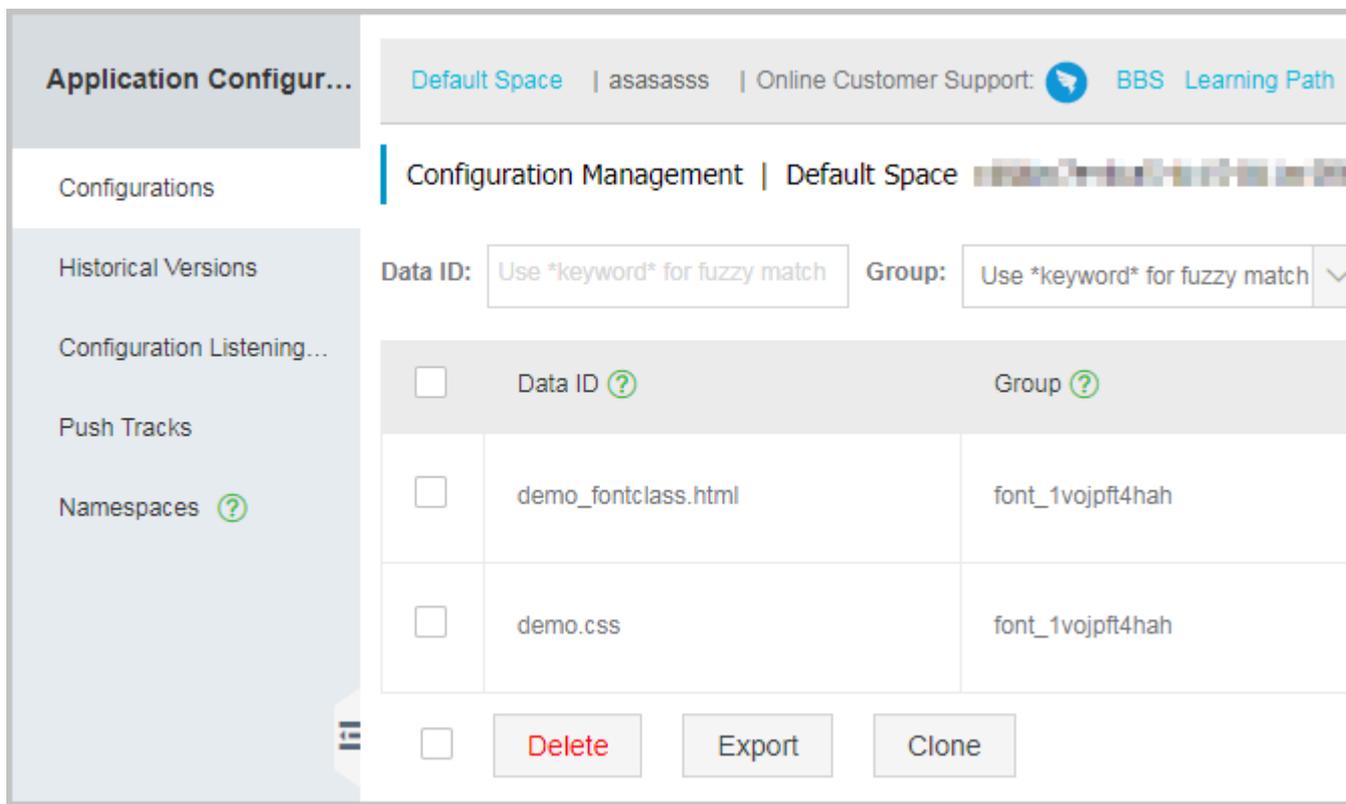
The configuration body:

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

In this example, first we create a configuration for the application myapp on ACM, and then listens for changes to this configuration with the native API of ACM. Once we change this configuration in the ACM console, every server to which this application is deployed receives the changed configurations, and the application status changes accordingly.

### Step 1: Create the configuration in ACM

1. Log on to the [ACM console](#).
2. In the left-side navigation pane, select **Configurations**, and then click the **+** button in the upper-right corner.



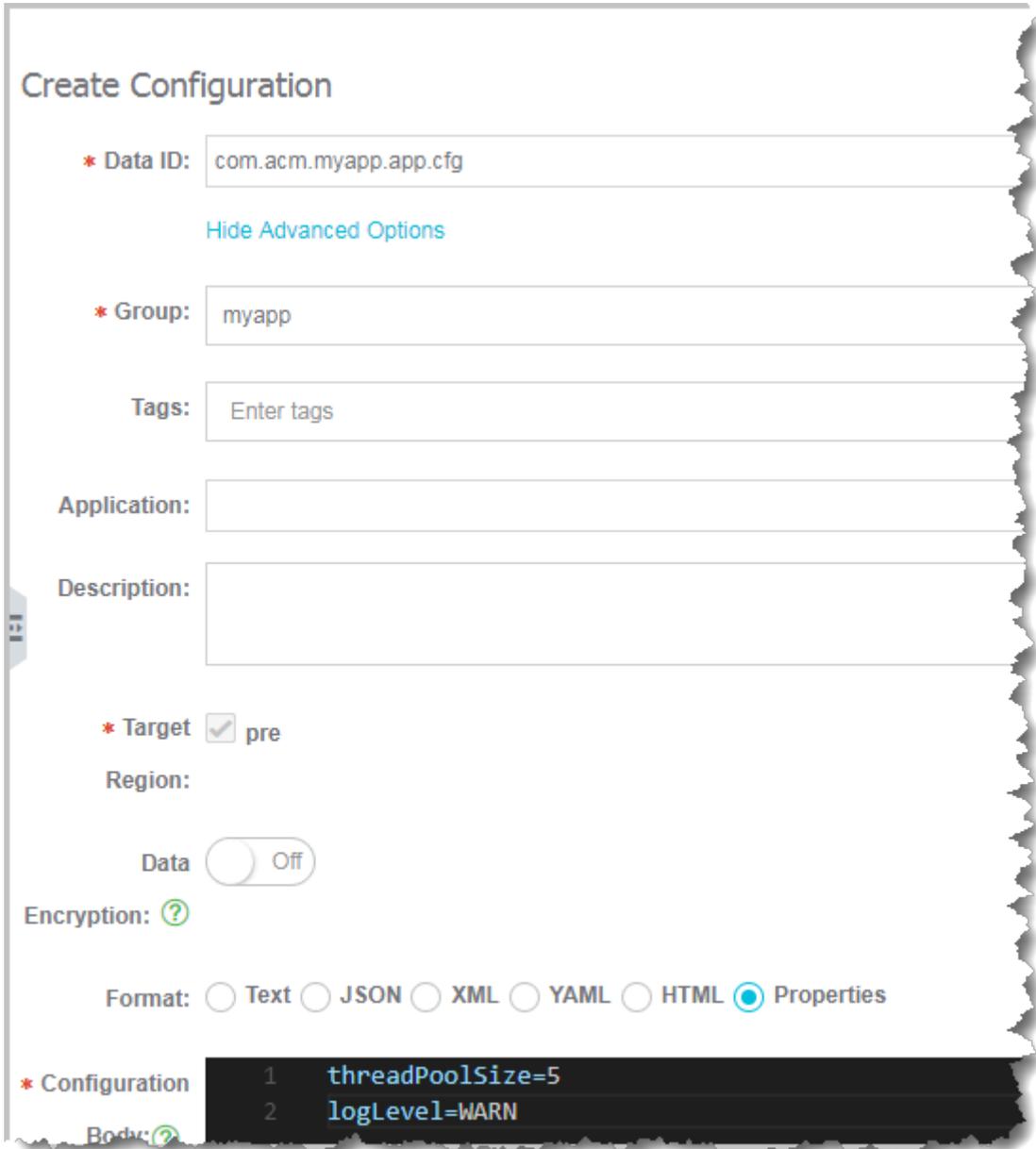
3. Enter the following data on the **Create Configuration** page, and click **Publish**.

- DataID: com.acm.myapp.app.cfg
- Group: myapp
- Configuration body:

```
threadPoolSize=5
```

```
logLevel=WARN
```

See the figure below:



### Step 2: Use the API to listen for configuration changes

1. Run the following command to create a Maven project, or download the sample project [myapp.zip](#).



**Note:**

For instructions on how to install and use Maven, see [Maven documentation](#).

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

The created project structure is as follows:

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

## 2. Add ACM client native API dependencies in POM. xml.

```
<dependencies>
  <dependency>
    <groupId>com.alibaba.edas.acm</groupId>
    <artifactId>acm-sdk</artifactId>
    <version>1.0.8</version>
  </dependency>
  <!-- Remove the following if logging implementation is
available. -->
  <dependency>
    <groupId>ch.qos.logback</groupId>
    <artifactId>logback-classic</artifactId>
    <version>1.1.7</version>
  </dependency>
</dependencies>
```

## 3. Add the raven-assembly-plugin packaging plug-in pom.xml.

```
<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>

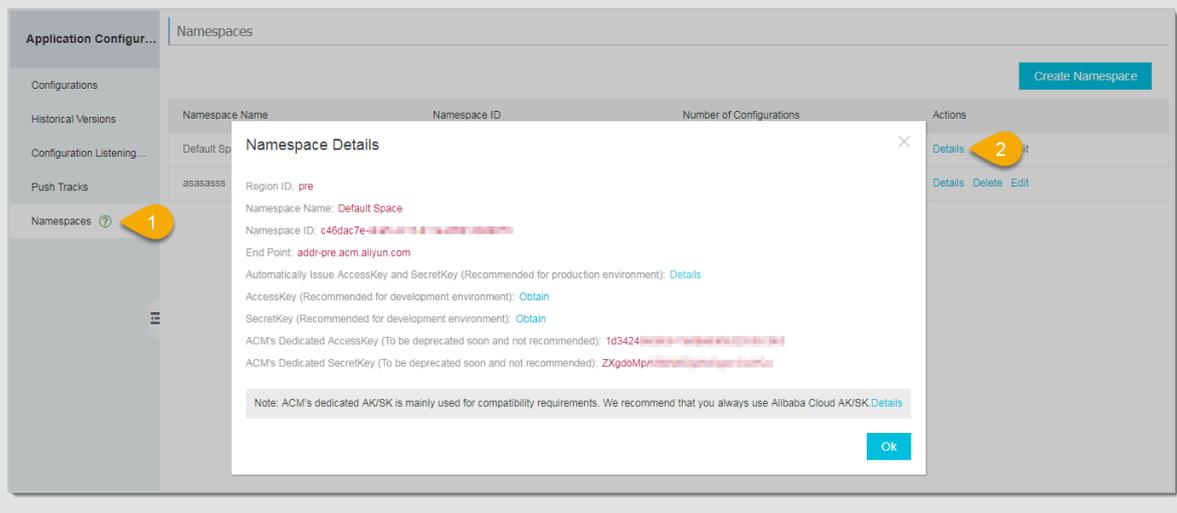
```

#### 4. Listen for configuration changes with API.



#### Note:

The user variables in the following code, such as `$endpoint`, `$namespace`, and `$accesskey` can be found on the **Namespace** page of the ACM console, as shown in the following figure.



```

/-- 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;

        // Copy the corresponding values from the namespace page of
        the console.
        Properties properties = new Properties();
        properties.put("endpoint", "$endpoint");
        properties.put("namespace", "$namespace");
        properties.put("accessKey", "$accessKey");
        properties.put("secretKey", "$secretKey");
    }
}

```

```
// If it is an encrypted configuration, then add the
following two lines for automatic decryption.
// properties.put("openKMSFilter", true);
// properties.put("regionId", "$regionId");

ConfigService.init(properties);

// Get configuration body directly.
try {
    String configInfo = ConfigService.getConfig(dataId,
group, timeoutInMills);
    appCfg.load(new StringReader(configInfo));
} catch (ConfigException e1) {
    e1.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

// Listen for configuration changes to get the latest
values.
ConfigService.addListener(dataId, group, new ConfigChan
geListener() {
    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();

    // Make sure the main thread does not exit.
    while (true) {
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
        }
    }
}
```

```
}
```

### Step 3: Deploy and launch the application

1. Package your application into a JAR file and copy it to both servers. Execute the following packaging command under the root directory of the project:

```
mvn clean package
```

2. Deploy and start the application in Shell.

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



#### Note:

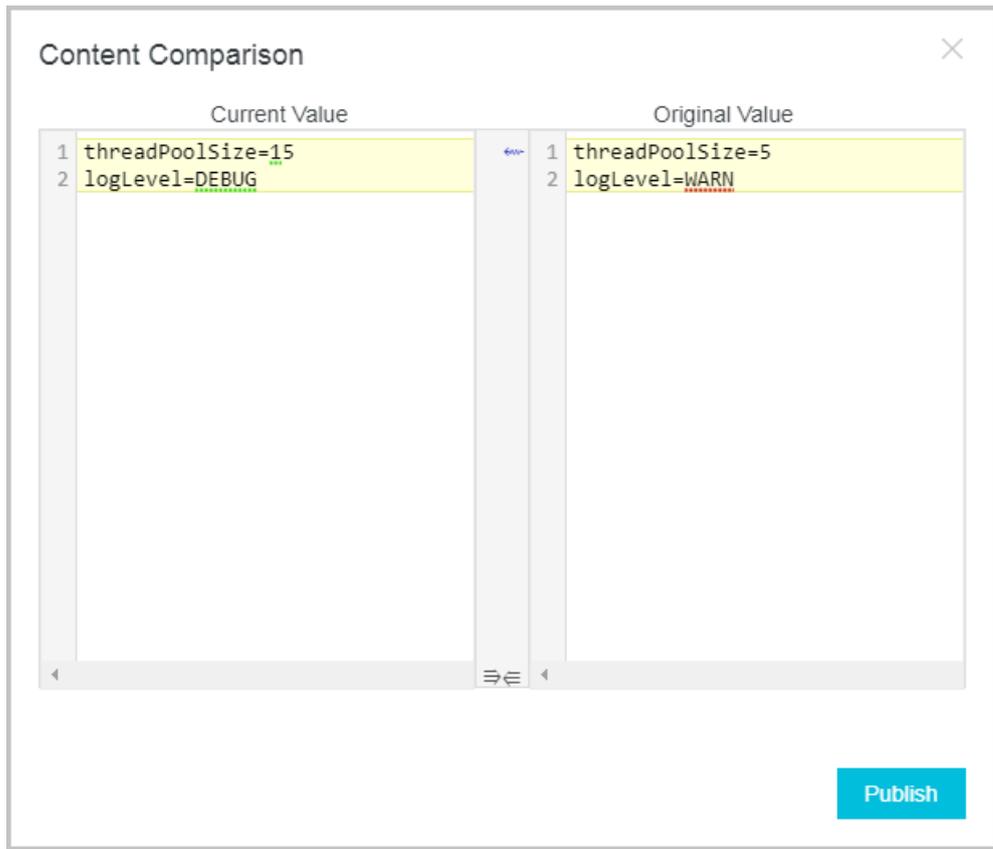
To run Java programs, you must install [JDK](#) on the server and set environment variable `JAVA_HOME`.

### Step 4: Search for and change the configuration in the ACM console

1. On the **Configurations** page of the ACM console, search for the configuration created in [Step 1: Create the configuration in ACM](#).
2. In the **Actions** column, click **Edit**.
3. On the **Edit Configuration** page, change the configuration body as follows and click **Publish**.

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

4. In the **Content Comparison** dialog box, verify that the configuration changes are correct, and click **Publish**.



### Verify the result

After the configuration is published, we can see that the configuration changes are received simultaneously on both servers on which the application is deployed, and the following information is printed.

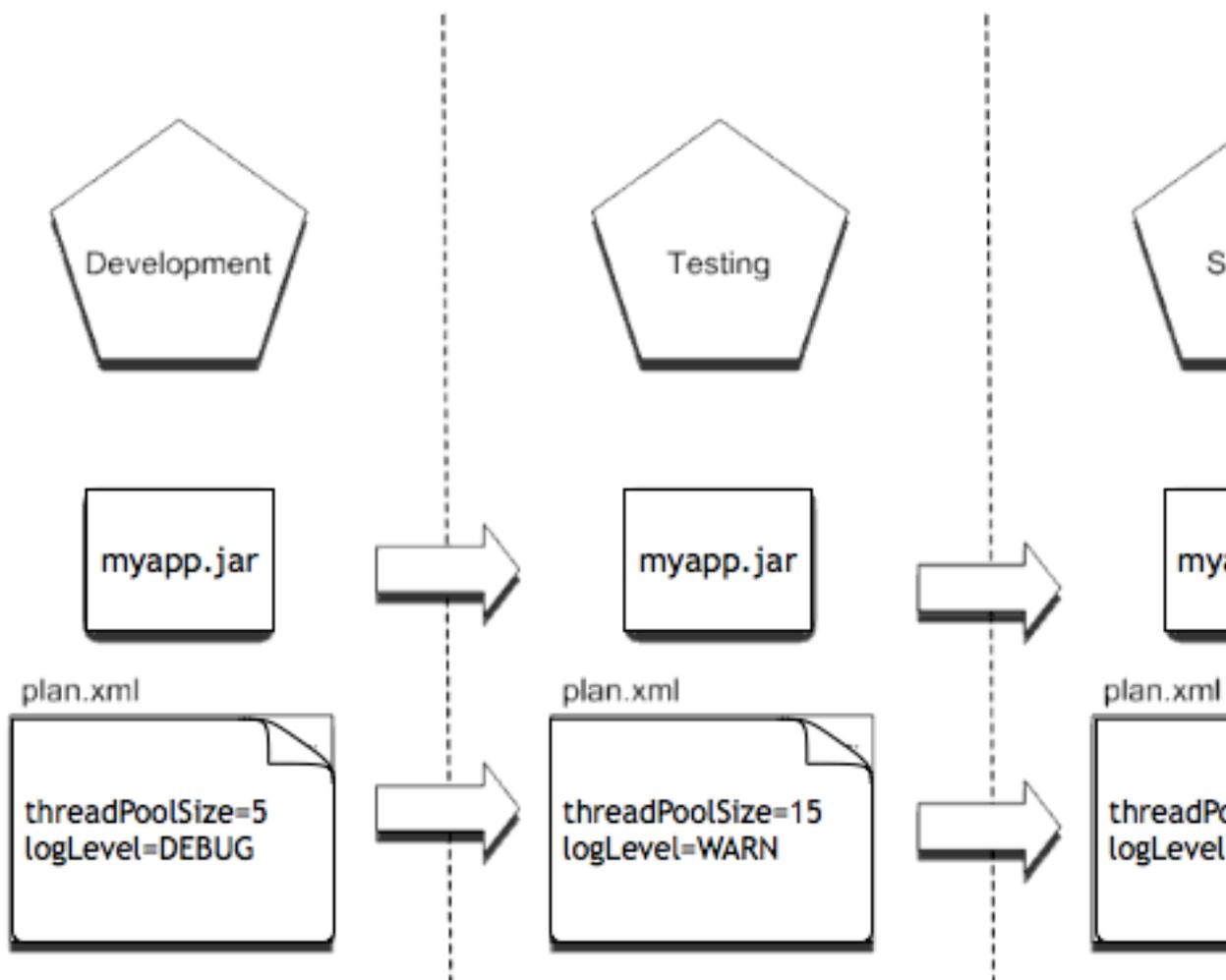
```
current thread pool size: 15
current log level: DEBUG
```

### 3 Make different settings for a configuration in different environments

This topic explains how to set different values for the same configuration in testing, staging, and development environment with ACM's namespaces.

#### Background information

In this task, we will use ACM's namespaces to set different values for the same configuration in testing, staging, and development environment. The expected result is as follows:



#### Step 1: Create a namespace in ACM

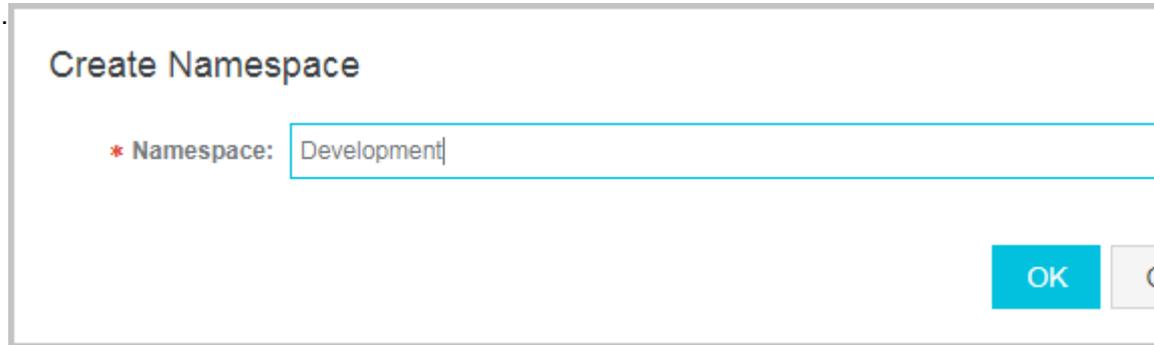
The following is an example of creating the namespace "Development".

1. Log on to the [ACM console](#).

2. In the left-side navigation pane, select **Namespaces**, and click the **Create Namespace** button in the upper-right corner: The **Create Namespace** dialog box is displayed.

3. In the dialog box, enter the namespace name

Development.

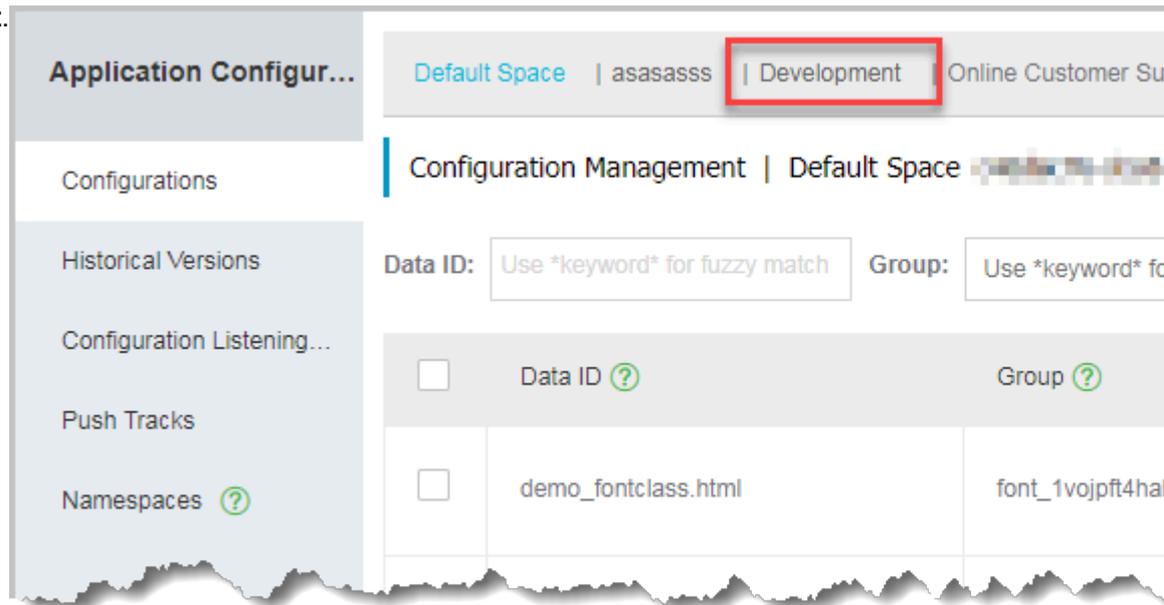


4. Repeat Steps 1 through 3 to create namespaces “Testing” and “Staging”.

### Step 2: Create a configuration under each namespace

1. On the **Configurations** page, select the namespace

Development.



2. Follow the instructions of [#unique\\_7/unique\\_7\\_Connect\\_42\\_section\\_ljb\\_bgt\\_42b](#) to create configurations with the same name.

### Conclusion

In real-world business scenarios, we often need to set different values for one configuration item based on different environments. As you can see in this example, you can easily do so with the Namespace feature of ACM.