阿里云
专有云Enterprise版
分布式命名和配置服务DNCS

用户指南

产品版本：V3.3.0

文档版本：20180330

为了无法计算的价值 | 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
	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。	**禁止**： 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	**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。	单击 **确定**。
courier字体	命令。	执行 cd /d C:/windows 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	ipconfig *[-all|-t]*
{}或者{a|b}	表示必选项，至多选择一个。	swich *{stand | slave}*

目录

法律声明.....I
通用约定.....I
1 快速入门.....1
1.1 域名服务.....1
1.2 配置服务.....1
1.2.1 创建配置.....1
1.2.2 查询配置.....4
2 使用域名和配置服务.....6
2.1 界面简介.....6
2.2 管理域名.....6
2.2.1 注册域名.....6
2.2.2 搜索域名.....7
2.2.3 修改域名配置.....8
2.2.4 删除域名.....10
2.2.5 添加IP.....10
2.2.6 编辑IP.....11
2.2.7 删除IP.....13
2.2.8 添加集群.....13
2.2.9 编辑集群.....15
2.2.10 删除集群.....16
2.3 管理配置.....17
2.3.1 编辑和删除配置.....17
2.3.2 查看监听状态.....20
2.3.3 管理配置版本.....21
3 SDK接入.....23
3.1 域名服务SDK接入.....23
3.1.1 Java SDK接入.....23
3.1.2 Spring Cloud SDK接入.....24
3.1.3 DNS-F接入.....26
3.2 配置服务SDK.....27
3.2.1 Java SDK接入.....27
3.2.1.1 环境准备.....27
3.2.1.2 监听配置.....29
3.2.1.3 获取配置.....30
3.2.2 Spring Cloud SDK接入.....31
3.2.2.1 环境准备.....31

3.2.2.2 配置注入 32
3.2.2.3 Spring Boot集成 33
4 常见问题 35
4.1 DNS-F无法启动 35
4.2 域名已在vipserver ops上注册，Java客户端无法获取IP 35
4.3 vipserver的容灾策略是怎样的？ 35

1 快速入门

1.1 域名服务

注册域名

1. 单击**域名服务**下的**域名列表**页面，然后单击右上方的**注册域名**，进入**域名注册**页面。
2. 输入域名名称，然后单击**注册**即可。

查询域名

在**域名注册**页面搜索框内输入域名名称，然后单击**搜索**，即可查询到注册的域名。

添加IP

1. 单击域名详情，进入域名详情页面，然后单击**添加IP**，进入**添加机器**页面。
2. 输入IP、端口和权重，并单击**添加**即可。

解析域名

引入maven依赖：

```
<dependency>
 <groupId>com.taobao.vipserver</groupId>
 <artifactId>vipserver-client</artifactId>
 <version>[最新版本]</version>
</dependency>
```

调用如下代码即可获得域名注册的IP：

```
Host host = VIPClient.srvHost(注册的域名);
String ip = host.getIP();
```

1.2 配置服务

1.2.1 创建配置

代码里的一些变量、参数等可以从代码中抽取出来，放到一个配置文件中，代码运行的时候从配置文件中获取，方便快速变更参数，降低代码维护成本。本文介绍如何在DNCS控制台创建配置。

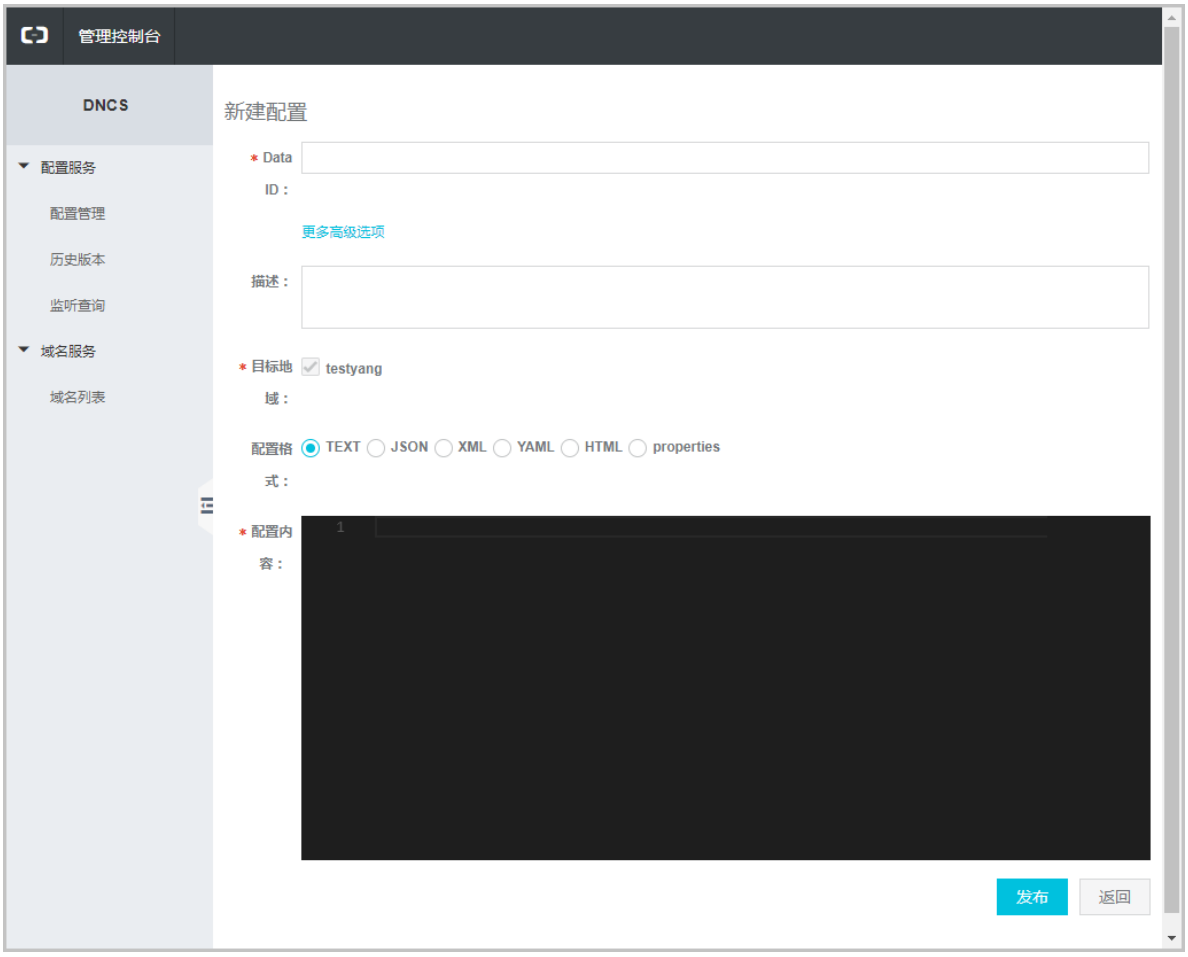
操作步骤

1. 在控制台左侧菜单栏选择**配置服务** > **配置管理**。

   进入**配置列表**页面。
2. 在**配置列表**页面单击**新建配置**。

进入**新建配置**页面。

3. 在**新建配置**页面填写配置信息，单击**发布**。



说明：

在DNCS中一个配置的数据模型是<Namespace+DataId+Group, Content>。DataId和Group不需要申请，在一个命名空间内保持唯一性即可。

各输入项说明如下：

输入项	说明
Data ID	配置ID，采用类似*package*.**class**（如*com.taobao.tc.refund.log*.**level**）的命名规则保证全局唯一性，class部分建议是配置的业务含义。 全部字符小写。只允许英文字符和4种特殊字符（"."、":"、"-"、"_"），不超过256字符。
Group	配置分组（命名空间），建议填写产品名（如ACM）或者模块名，保证唯一性。后续可以根据group进行鉴权。 只允许英文字符和4种特殊字符（"."、":"、"-"、"_"），不超过128字符。
标签	配置标签，方便用户按照自己的维度管理配置，最多支持5个标签，每个标签不超过32字符。
描述	配置描述信息，方便其他人理解配置含义，不超过128字符。
关联应用	配置关联应用名。
目标地域	大部分用户只发当前地域，如果需要把配置发布到多个地域请勾上所需地域。
Beta发布	勾选Beta后面复选框，填写Beta的机器，单击发布后则是Beta发布。
格式校验	如果配置是XML或者JSON格式，可以勾选对应数据格式的单选框，ACM控制台会帮助您做格式校验。
配置内容	配置值，建议不超过10K，最大不超过100K。

后续操作

创建好配置后，即可在**配置管理**页面看到您新建的配置，通过示例代码获取到该配置。

1.2.2 查询配置

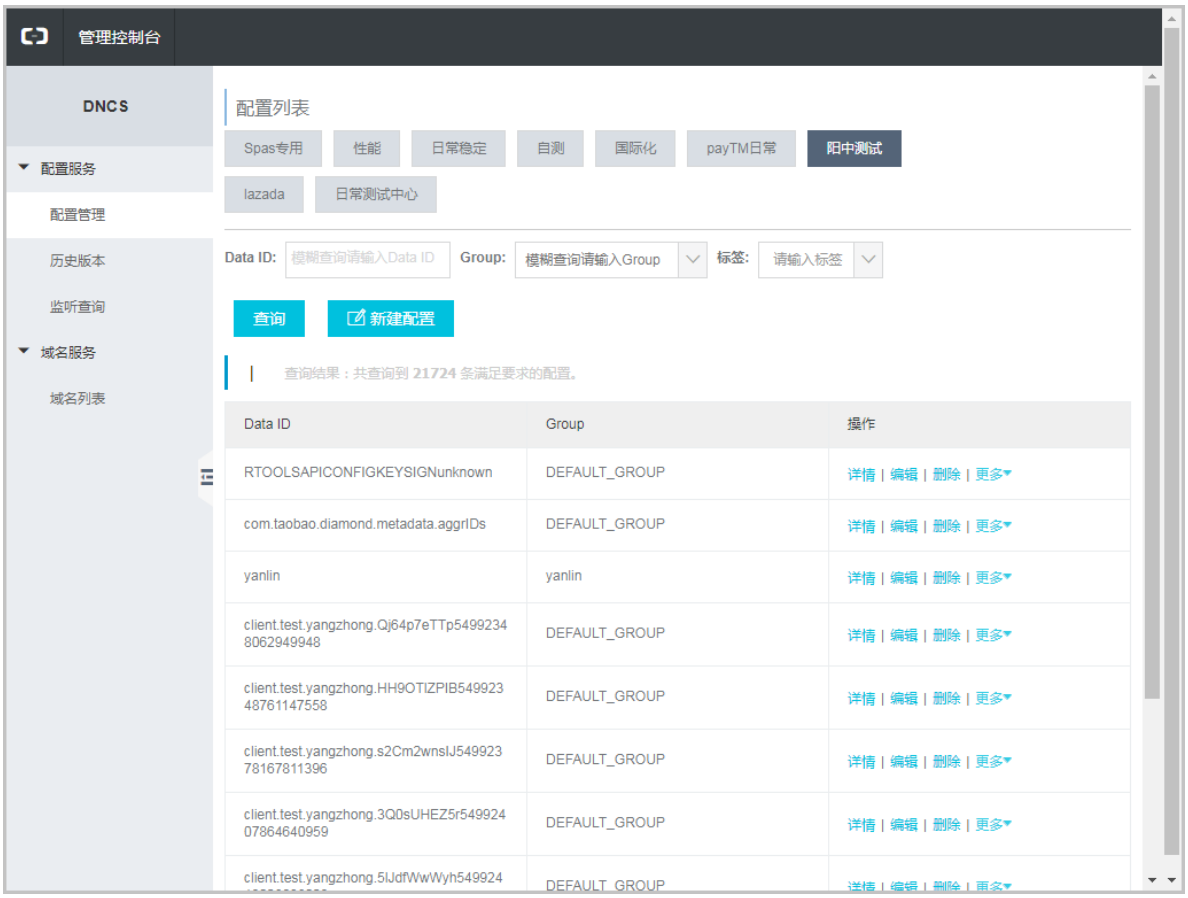
DNCS控制台支持Data ID、Group ID组合查询或者其中任何一个条件查询。

操作步骤

1. 在控制台左侧菜单栏选择**配置服务 > 配置管理**。

   进入**配置列表**页面。

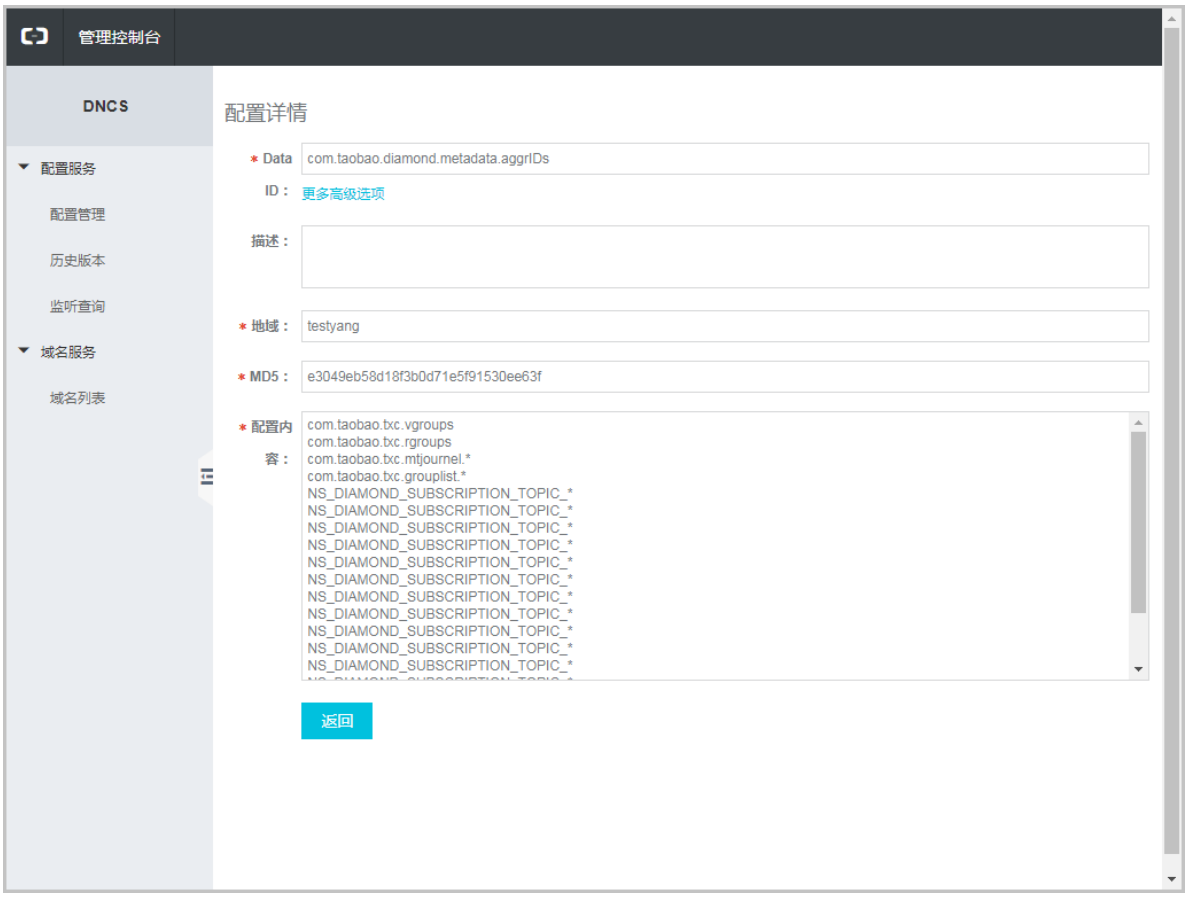
2. 在**配置列表**页面输入搜索条件，并单击**查询**。



   搜索到配置之后，即可对相应配置进行查看详情、编辑或删除操作。

3. 在搜索结果列表中单击**详情**。

   进入**配置详情**页面，可查看如下配置详细信息。


管理控制台
DNCS
配置服务
配置管理
历史版本
监听查询
域名服务
域名列表
配置详情
* Data com.taobao.diamond.metadata.aggrIDs
ID： 更多高级选项
描述：
* 地域： testyang
* MD5： e3049eb58d18f3b0d71e5f91530ee63f
* 配置内容：
com.taobao.txc.vgroups
com.taobao.txc.rgroups
com.taobao.txc.mtjournel.*
com.taobao.txc.grouplist.*
NS_DIAMOND_SUBSCRIPTION_TOPIC_*
返回

2 使用域名和配置服务

2.1 界面简介

控制台界面左侧导航栏有DNCS包含的各个服务组件，单击相应子页面可以进入各组件的管理页面。

图 2-1: DNCS控制台界面

2.2 管理域名

2.2.1 注册域名

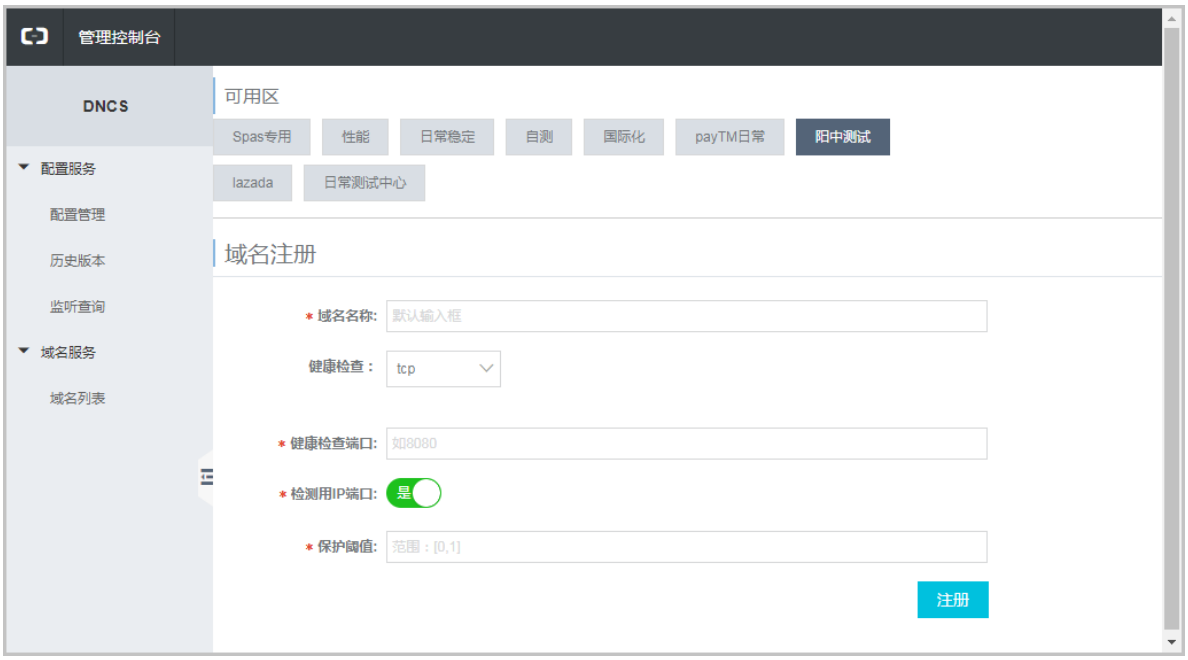
操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面单击**注册域名**。

   进入**域名注册**页面。



3. 在**域名注册**页面，按照下表填写相关属性。

属性	含义	是否必选
域名名称	域名服务名称	是
健康检查	对域名下IP的默认健康检查方式，以后可针对各集群单独配置健康检查方式	是
检测路径	如果选择HTTP健康检查方式，这里填入作为健康检查的URL路径（形如/health）	否，默认为根路径
健康检查端口	健康检查连接的端口	否，默认为80
检测用IP端口	是否用注册IP的端口进行健康检查	是
保护阈值	一个0到1之间的浮点数，用来表示不健康的机器占机器总数的比例，具体含义参考名词解释章节	否，默认为0

2.2.2 搜索域名

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

2.2.3 修改域名配置

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

4. 在域名详情页面，单击**域名配置**旁的**更新配置**。

   进入**域名更新**页面。

5. 在**域名更新**页面更新相关属性。

表 2-1: 在域名更新页面可以更新的属性

属性	含义	是否必选
保护阈值	参见名词解释章节	是
管理员	域名所有者，不建议修改	是

2.2.4 删除域名

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

   符合条件的域名显示在结果列表中。

3. 在**域名列表**页面的表格中，对要删除的域名单击**删除**。

   进入确认对话框。

4. 在确认对话框中单击**确认**。

   域名被立即删除。

2.2.5 添加IP

该操作可以向指定域名添加IP。

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

4. 在目标集群名称右侧单击**添加IP**。

   进入**添加机器**页面。

5. 在**添加机器**页面，根据需要执行以下操作。

   - 如需添加一个IP，则单击**单机添加**，并按照下表填写参数。

属性	含义	是否必选
IP	注册的IP	是
端口	注册的端口	否，默认为80
权重	注册IP的权重	否，默认为1

   - 如需添加多个IP，则单击**批量添加**，并按照下表填写参数。

属性	含义	是否必选
IP列表	注册的IP列表，格式为IP:port_weight_cluster，每行一个IP	是

2.2.6 编辑IP

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

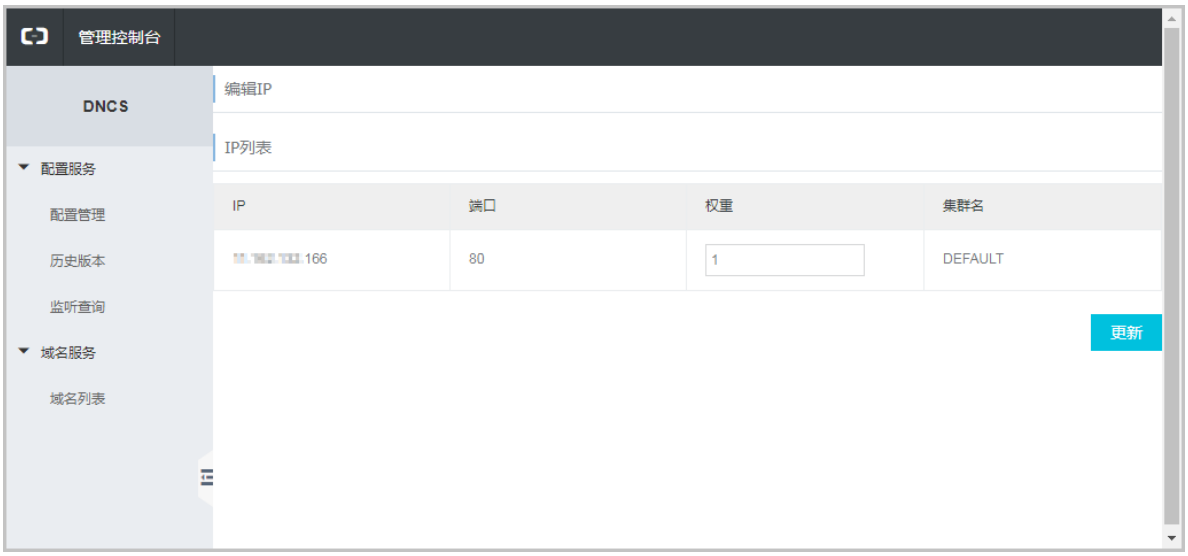
   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

4. 在域名详情页面，勾选列表中待编辑的一个或多个IP，然后单击**编辑选中IP**。

   进入**编辑IP**页面。



5. 在**编辑IP**页面，按需更新属性，然后单击**更新**。

2.2.7 删除IP

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

4. 在域名详情页面，勾选列表中待编辑的一个或多个IP，然后单击**删除选中IP**。

   选中IP被立即删除。

2.2.8 添加集群

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

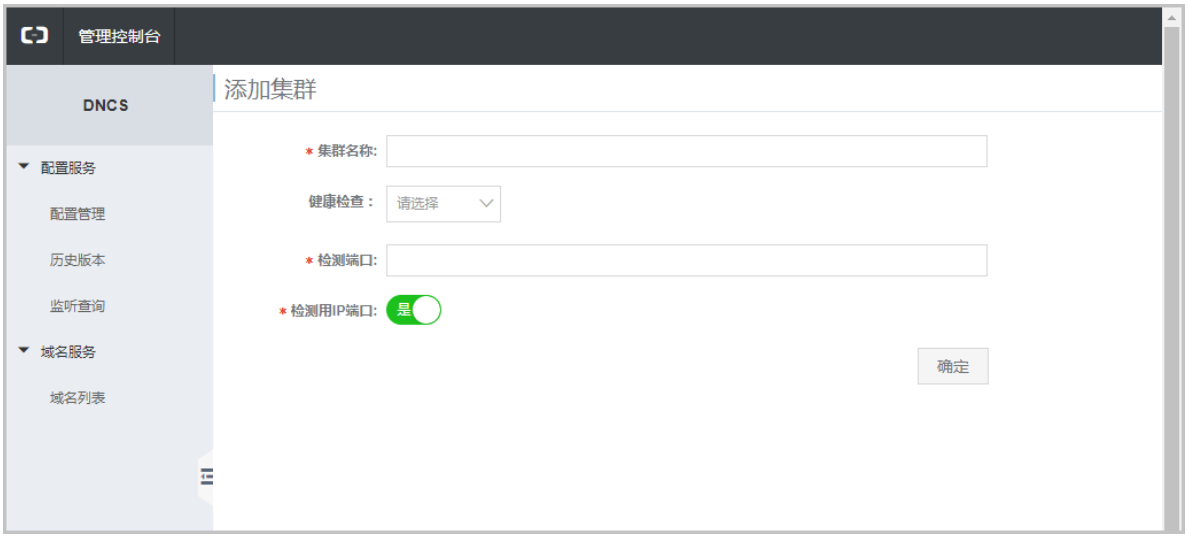
   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

4. 在域名详情页面，单击**域名配置**旁的**添加集群**。

   进入**添加集群**页面。



5. 在**添加集群**页面，按照下表填写相关属性。

属性	含义	是否必选
集群名称	域名服务名称	是
健康检查	对域名下IP的默认健康检查方式，以后可针对各集群单独配置健康检查方式	是

属性	含义	是否必选
检测路径	如果选择HTTP健康检查方式，这里填入作为健康检查的URL路径（形如/health）	否，默认为根路径
健康检查端口	健康检查连接的端口	否，默认为80
检测用IP端口	是否用注册IP的端口进行健康检查	是

2.2.9 编辑集群

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

4. 在域名详情页面，单击集群名称旁的**更新集群配置**。

   进入**配置集群**页面。

5. 在**配置集群**页面，按照下表更新相关属性。

> **说明**：
> 该页面上的**集群名称**不可编辑。

属性	含义	是否必选
集群名称	域名服务名称	是
健康检查	对域名下IP的默认健康检查方式，以后可针对各集群单独配置健康检查方式	是
检测路径	如果选择HTTP健康检查方式，这里填入作为健康检查的URL路径（形如/health）	否，默认为根路径
健康检查端口	健康检查连接的端口	否，默认为80
检测用IP端口	是否用注册IP的端口进行健康检查	是

2.2.10 删除集群

操作步骤

1. 在控制台左侧菜单栏选择**域名服务** > **域名列表**。

   进入**域名列表**页面。

2. 在**域名列表**页面的域名名称搜索框内输入要搜索的域名，然后单击**搜索**。

   符合条件的域名显示在结果列表中。

3. 在结果列表中单击**详情**。

   进入域名详情页面，可查看域名详细信息。

4. 在域名详情页面，单击集群名称旁的**删除集群**。

   进入**警告**对话框。

5. 在**警告**对话框中单击**确认**。

   集群被立即删除。

2.3 管理配置

2.3.1 编辑和删除配置

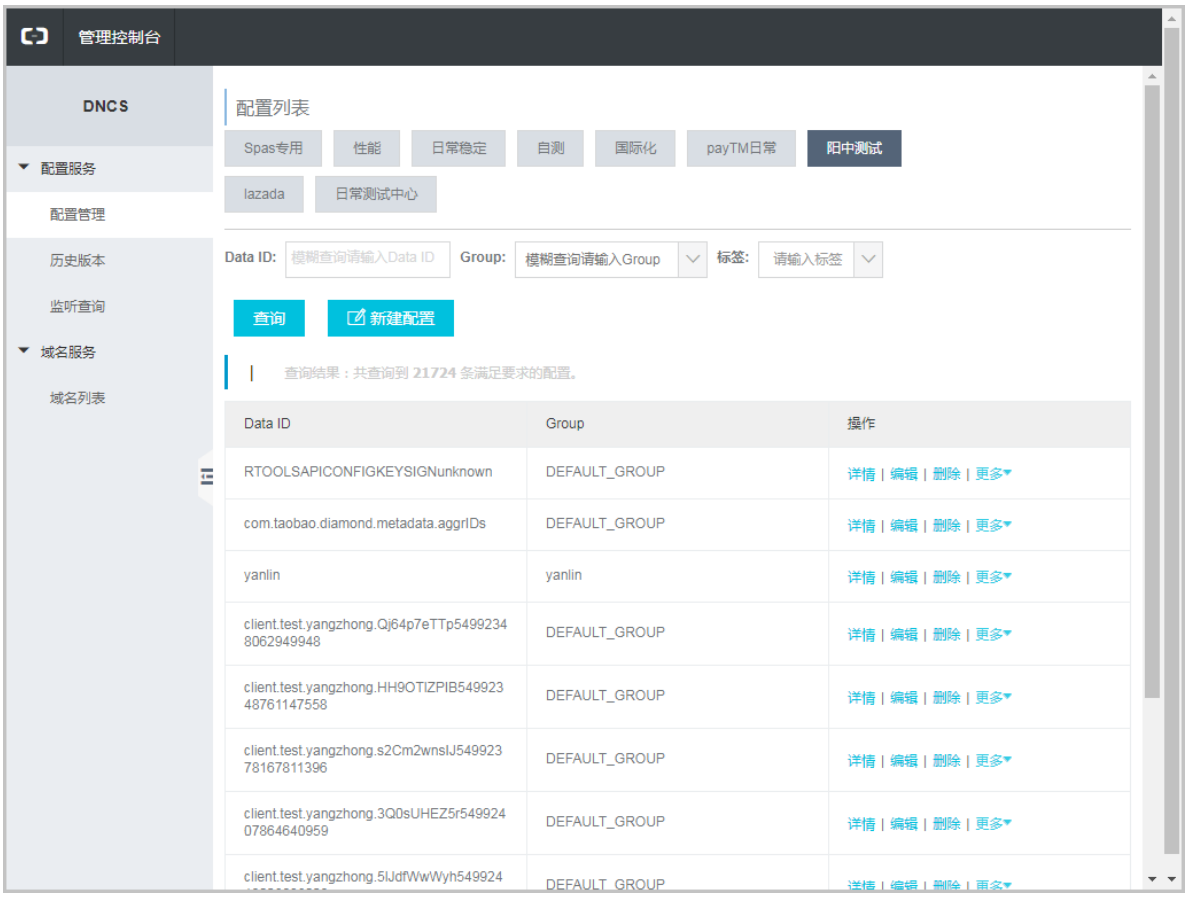
创建完配置后，就可以在代码里使用了。由于配置容易发生变化，因此经常需要在DNCS控制台编辑配置。DNCS控制台提供了搜索、查看、编辑、删除配置等功能。

操作步骤

1. 在控制台左侧菜单栏选择**配置服务** > **配置管理**。

   进入**配置列表**页面。

2. 在**配置列表**页面输入搜索条件，并单击**查询**。



搜索到配置之后，即可对相应配置进行查看详情、编辑或删除操作。

3. 在查询结果列表中，对目标配置单击**编辑**。

   - 如需删除配置，对目标配置单击**删除**，并在**删除配置**对话框中单击**确认**。

   进入**编辑配置**页面。

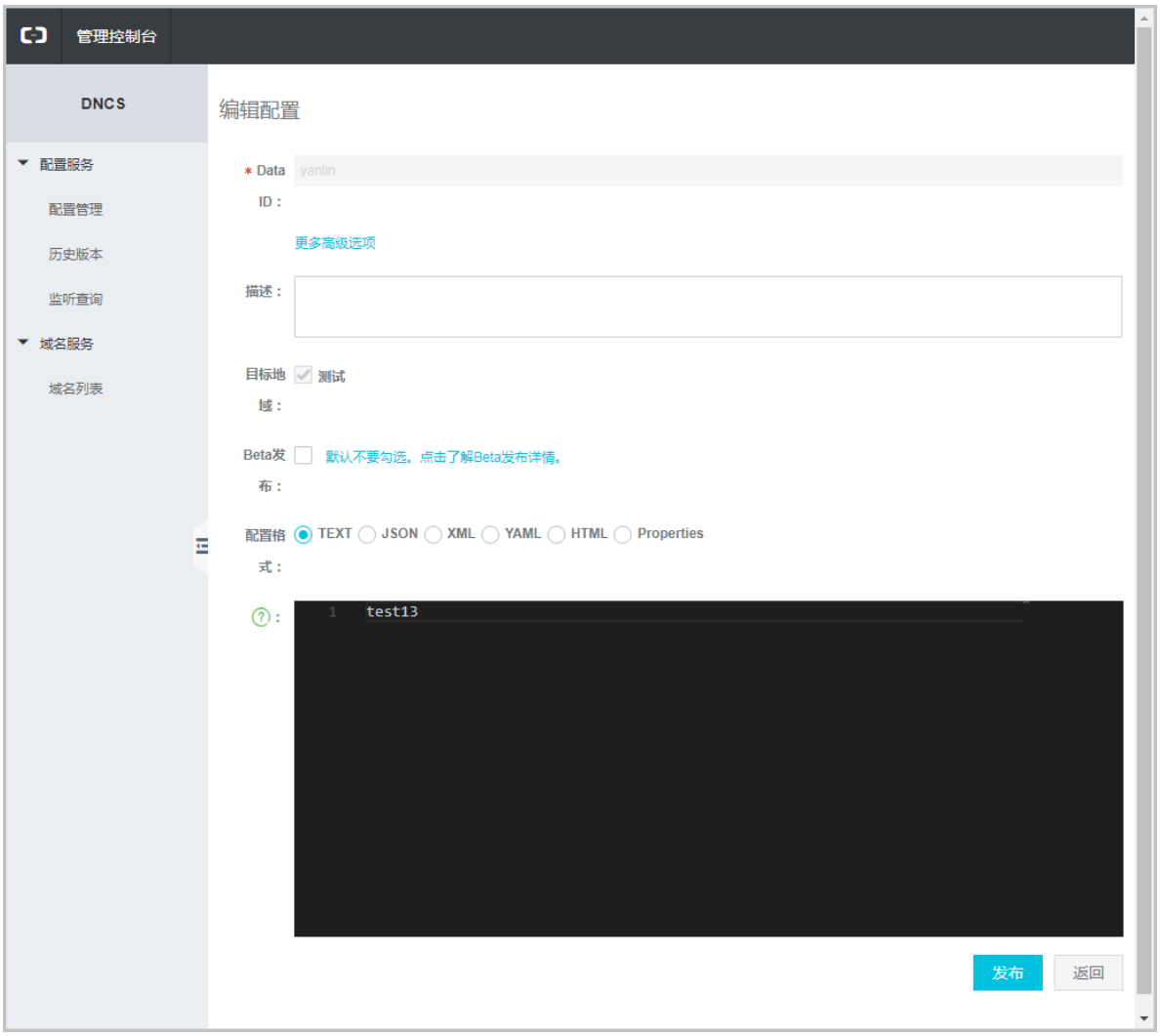


表 2-2: 编辑配置页面的高级特性

特性	简介	使用说明
格式校验	提供JSON、XML格式语法校验能力。在配置格式中选择不同的格式，会提供不同格式预发校验，减少语法格式导致的问题。	编辑配置前在**配置格式**选项中选择编辑文本的格式。
变更对比	修改完配置提交发布时，DNCS提供变更对比功能。变更前检查一次，可降低误操作概率。	修改完配置，单击**发布**按钮，弹出变更对比框。

特性	简介	使用说明
Beta发布	对于重要的配置变更，一个变更错误可能导致巨大故障，因此需要将该配置发到几台机器上先验证一下。如果没有问题再全部推送，降低错误变更影响。	勾选**Beta发布**，填写需要Beta发布的机器IP（本机测试注意填写公网IP）。

4. 在**编辑配置**按需编辑相关属性，然后单击**发布**。

2.3.2 查看监听状态

修改配置后，需要查看一下修改后的配置信息是否已推送到监听该配置的机器上。此查询只对使用了监听配置接口监听配置的客户端有效。

操作步骤

1. 在控制台左侧菜单栏选择**配置服务** > **监听查询**。

   进入**监听查询**页面。

2. 在**监听查询**页面，按需执行以下操作。

   - 如需查询配置推送的机器及推送状态，在**查询维度**下拉框中选择**配置**，输入其他查询条件，并单击**查询**。
   - 如需查询机器监听的配置列表，在**查询维度**下拉框中选择**IP**，输入其他查询条件，并单击**查询**。

2.3.3 管理配置版本

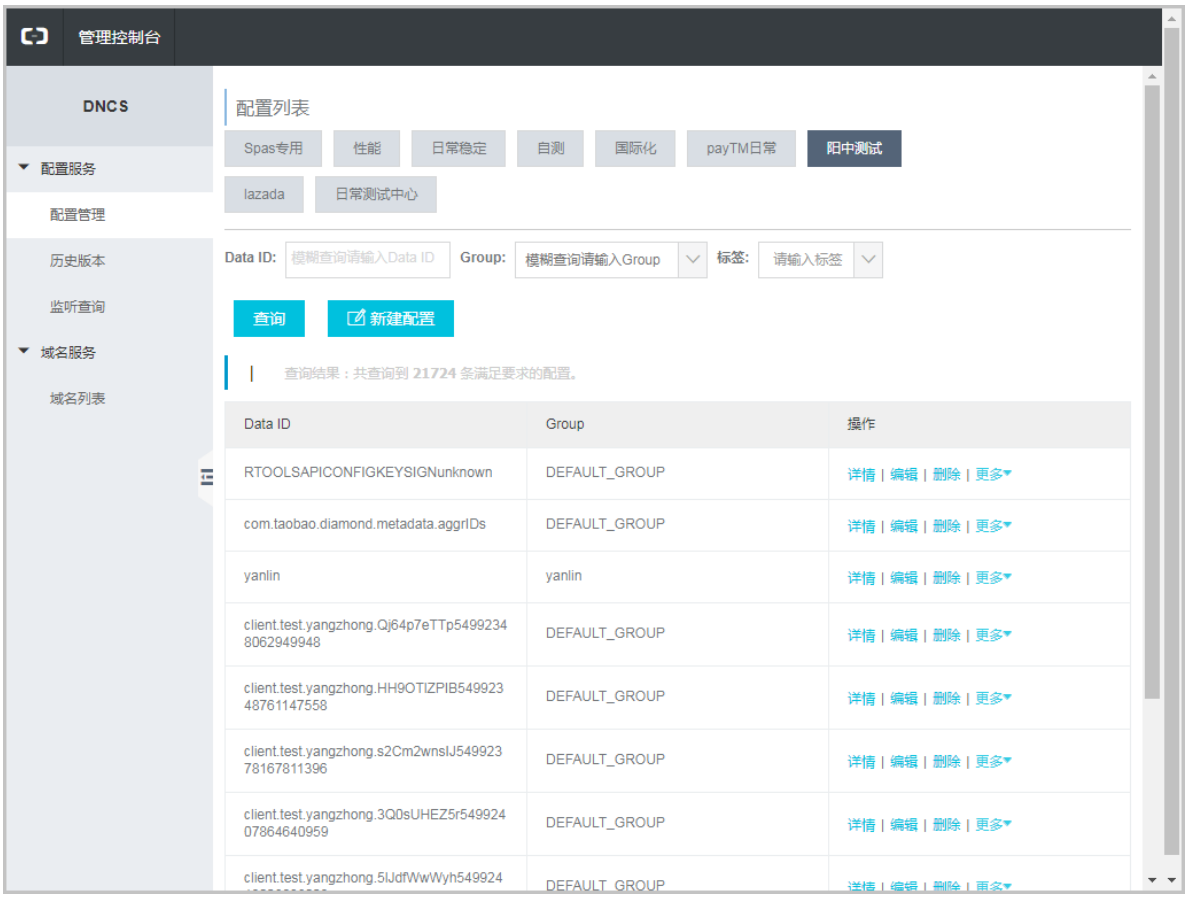
如果改错配置，需要立即回滚解决。DNCS提供了配置变更历史查询和回滚功能。

操作步骤

1. 在控制台左侧菜单栏选择**配置服务** > **配置管理**。

   进入**配置列表**页面。

2. 在**配置列表**页面输入搜索条件，并单击**查询**。



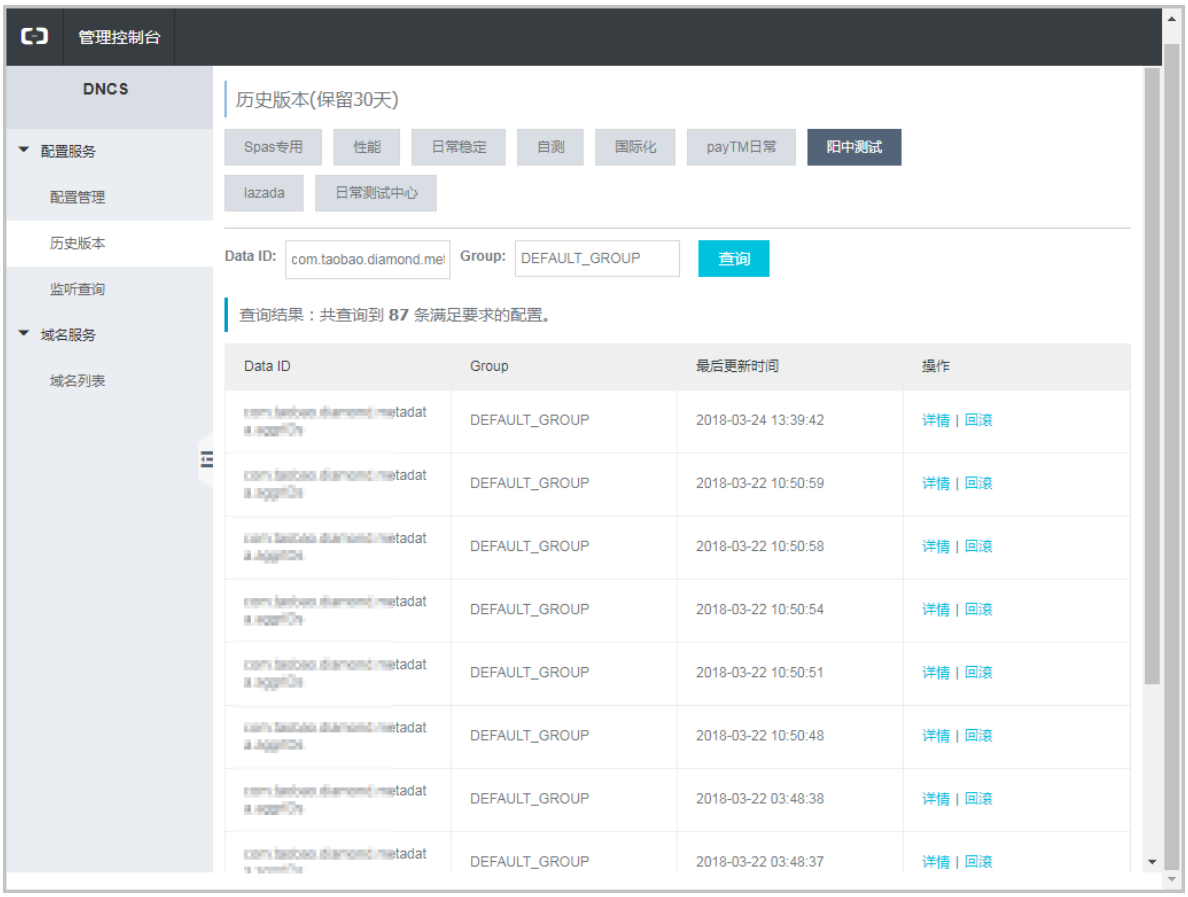
   搜索到配置之后，即可对相应配置进行查看详情、编辑或删除操作。

3. 在查询结果列表中，对目标配置单击**更多** > **历史版本**。

   说明：

   DNCS只保存30天的变更记录。

   进入**历史版本**页面。



4. 在**历史版本**页面，根据需要执行以下操作。
   - 如需查看配置变更前的值，单击**操作**栏中的**详情**，然后在**历史详情**页面查看详细信息。
   - 如需回滚到以前的版本，单击**操作**栏中的**回滚**，然后在**配置回滚**页面单击**回滚**。

3 SDK接入

3.1 域名服务SDK接入

3.1.1 Java SDK接入

环境准备

运行Java代码前，请先准备好环境。

操作步骤

引入依赖

```
<dependency>
 <groupId>com.taobao.vipserver</groupId>
 <artifactId>vipserver-client</artifactId>
 <version>[最新版本]</version>
</dependency>
```

查询域名

Java SDK提供的查询接口较多，可以根据需要查询域名下的一个或多个IP，详情参考Java SDK文档。此处举例说明如何查询域名下的一个IP。

代码示例

```
func () {
 ...
 Host host = VIPClient.srvHost("www.taobao.com");
String ip = host.getIP();
HttpClient.get("http://ip:port/index.htm?"); port是服务提供方开放的http端口
...
}
```

参数说明

参数	含义	是否必选
dom	域名名称	是

监听域名

通过监听域名接口，用户可持续获取域名的更新通知。当域名有更新时，用户传入的一个回调方法将会被调用。

代码示例

```
func () {
 ...
Host host = VIPClient.listen( “www.taobao.com” , new HostListener() {
 @Override
 public void ipChanged(List<Host> hosts) {
 System.out.println(hosts);
 }
});
...
}
```

参数说明

参数	含义	是否必选
dom	域名名称	是
listener	域名更新的回调接口	是

3.1.2 Spring Cloud SDK接入

环境准备

与Java SDK类似，需要先引入maven依赖：

操作步骤

引入依赖

```
<dependency>
<groupId>com.alibaba.middleware.vipserver</groupId>
<artifactId>spring-cloud-vipserver-client</artifactId>
<version>[最新版本]</version>
 </dependency>
```

注册域名

Spring Cloud体系下，域名注册是隐式的、自动化的，域名将会在服务启动时，自动将配置好的域名注册到DNCS服务端。类似地，在服务下线时，也会自动注销本机注册的域名信息注销。详见以下步骤。

操作步骤

1. 参数配置

   参数需要配置到应用的application.properties文件里。

参数	含义	示例值
address.server.domain	地址服务器域名	jmenv.tbsite.net
address.server.port	地址服务器端口	8080
vipserver.register.enabled	是否自动注册域名	true
vipserver.register.ip	注册IP，不配置该属性时，将调用Java SDK获取本机IP	1.1.1.1
vipserver.register.doms	注册的域名，多个域名以英文逗号分隔	1.dom,2.dom
vipserver.register.port	注册的端口	8080
spring.application.name	若vipserver.register.doms属性为空，则使用该属性作为域名注册	app1

2. 使用注解

   在应用启动入口添加注解。

代码示例

```
@EnableVipserverClient
@SpringBootApplication
public class ProviderApplication {
public static void main(String[] args) {
 SpringApplication.run(ProviderApplication.class, args);
}
 }
```

订阅域名

在应用启动入口，添加@EnableVipserverClient和@EnableFeignClients注解。

代码示例

```
@EnableFeignClients
@SpringBootApplication
@EnableVipserverClient
public class ConsumerApplication {
public static void main(String[] args) {
 SpringApplication.run(ConsumerApplication.class, args);
}
 }
```

使用RestTemplate访问域名

操作步骤

1. 定义基于域名服务提供的HttpClientFactory属性的RestTemplate bean。

```
<bean id="httpClientFactory" class="com.alibaba.middleware.vipserver.sub.templated.
VipserverHttpRequestFactory">
</bean>
<!--RestTemplate-->
<bean id="restTemplate" class="org.springframework.web.client.RestTemplate">
 <constructor-arg ref="httpClientFactory"/>
</bean>
```

2. 代码调用

   例如注册的VIPServer域名为liaoyuan.http.vipserver，下面挂了一个IP“11.162.127.242”，该IP提供http服务“11.162.127.242/liaoyuan/api/pids”，那么可以通过如下代码使用RestTemplate访问域名。

代码示例

```
@Autowired
private RestTemplate restTemplate;
void func() {
HttpHeaders headers = new HttpHeaders();
headers.add("User-Agent", "LiaoYuan");
HttpEntity<String> entity = new HttpEntity<String>("parameters", headers);
ResponseEntity<String> result = restTemplate.exchange("http://liaoyuan.http.vipserver/liaoyuan
/api/pids", HttpMethod.GET, entity, String.class);
System.out.println(result.getBody());
 }
```

使用FeignClient访问域名

假设订阅的域名是test.dom.a，那么只需要在FeignClient上加上该域名的FeignClient注解即可。

代码示例

```
@FeignClient(name = "test.dom.a")
public interface ITestApi {
@RequestMapping(value = "/api/hello", method = RequestMethod.GET)
String hello();
 }
```

3.1.3 DNS-F接入

环境准备

DNS-F是指根据DNS协议的原理，在客户机器上安装一个监听DNS查询的服务，这样即可将用户查询的域名重定向到域名服务指定的IP下。DNS-F的安装依赖一些系统环境变量：

变量	含义	示例值
dncs_server_list	DNCS服务端IP列表，DNS-F用该变量寻址DNCS服务端，多个服务端IP以英文逗号分隔	1.1.1.1,2.2.2.2
DNSF_SESSION_STICKY	“会话保持”是否打开。这里的“会话保持”特指每一次使用DNS-F查询某域名时，域名将会返回一个固定的IP。	8080

操作步骤

1. 下载RPM包

   点击下载t-middeware-vipserver-dnsclient包【到时候替换为对外暴露的下载地址】

2. 安装RPM包

   运行以下命令安装：

   `rpm –ivh t-middeware-vipserver-dnsclient-1.1.0-1557292.alios7.noarch.rpm`

查询域名

使用DNS-F查询域名无需再接入任何代码，直接在机器上ping域名接口得到指定的IP结果。

3.2 配置服务SDK

3.2.1 Java SDK接入

3.2.1.1 环境准备

本文介绍DNCS API的接入步骤，包含SDK的获取及示例代码。

SDK获取

直接填写以下POM的配置，依赖API的SDK即可。

```
```
<dependency>
    <groupId>com.alibaba.edas.acm</groupId>
    <artifactId>acm-sdk</artifactId>
    <version>1.0.4</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>
```
```

```
```

示例代码

```
import java.util.Properties;
 import com.alibaba.edas.acm.ConfigService;
 import com.alibaba.edas.acm.exception.ConfigException;
 import com.alibaba.edas.acm.listener.ConfigChangeListener;
 import com.alibaba.edas.acm.listener.PropertiesListener;
 // 示例代码，仅用于示例测试
 public class ACMTest {
    // 属性/开关
    private static String config = "DefaultValue";
    private static Properties acmProperties = new Properties();
    public static void main(String[] args) {
        try {
            // 本地调试模式。本机测试无法连接ACM服务集群，因此设置debug模式（该模式下获取值为null，不会进入回调函数）。
            ConfigService.setDebug(true);
            // 初始化配置服务，控制台通过示例代码自动获取下面参数
            ConfigService.init("${endpoint}", "${namespace}", "${accessKey}", "${secretKey}");
            // 主动获取配置
            String content = ConfigService.getConfig("${dataId}", "${group}", 6000);
            System.out.println(content);
            // 初始化的时候，给配置添加监听，配置变更会回调通知
            ConfigService.addListener("${dataId}", "${group}", new ConfigChangeListener() {
                public void receiveConfigInfo(String configInfo) {
                    // 当配置更新后，通过该回调函数将最新值返回给用户。
                    // 注意回调函数中不要做阻塞操作，否则阻塞通知线程。
                    config = configInfo;
                    System.out.println(configInfo);
                }
            });
            /**
             * 如果配置值的内容为properties格式（key=value），可使用下面监听器。以便一个配置管理多个配置项
             */
            /**
            ConfigService.addListener("${dataId}", "${group}", new PropertiesListener() {
                @Override
                public void innerReceive(Properties properties) {
                    // TODO Auto-generated method stub
                    acmProperties = properties;
                    System.out.println(properties);
                }
            });
            **/
        } catch (ConfigException e) {
            e.printStackTrace();
        }
        // 测试让主线程不退出，因为订阅配置是守护线程，主线程退出守护线程就会退出。 正式代码中无需下面代码
        while (true) {
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
```

```
    }
    // 通过get接口把配置值暴露出去使用
    public static String getConfig() {
        return config;
    }
    // 通过get接口把配置值暴露出去使用
    public static Object getPorpertiesValue(String key) {
        if (acmProperties != null) {
            return acmProperties.get(key);
        }
        return null;
    }
}
```

3.2.1.2 监听配置

描述

如果希望DNCS推送配置变更，可以使用DNCS动态监听配置接口来实现。

```
public static void addListener(String dataId, ConfigChangeListenerAdapter listener)
```

请求参数

参数名	参数类型	描述
dataId	string	配置ID，采用类似package.class（如com.taobao.tc.refund.log.level）的命名规则保证全局唯一性，class部分建议是配置的业务含义。 全部字符小写。只允许英文字符和4种特殊字符（"."、":"、"-"、"_"）。不超过256字节。
group	string	配置分组，建议填写产品名#模块名（如DNCS:Test）保证唯一性。 只允许英文字符和4种特殊字符（"."、":"、"-"、"_"），不超过128字节。
listener	ConfigChangeListener	监听器，配置变更进入监听器的回调函数。

返回值

参数类型	描述
string	配置值，初始化或者配置变更的时候通过回调函数返回该值。

请求示例

```
// 初始化配置服务，控制台通过示例代码自动获取下面参数
ConfigService.init("${endpoint}", "${namespace}", "${accessKey}", "${secretKey}");
// 初始化的时候，给配置添加监听,配置变更会回调通知
ConfigService.addListener("${dataId}", "${group}", new ConfigChangeListener() {
    public void receiveConfigInfo(String configInfo) {
        // 当配置更新后，通过该回调函数将最新值返回给用户
        System.out.println(configInfo);
    }
});
// 测试让主线程不退出，因为订阅配置是守护线程，主线程退出守护线程就会退出。 正式代码中无需下面代码
while (true) {
    try {
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

3.2.1.3 获取配置

描述

用于服务启动的时候从DNCS获取配置。

```
public static String getConfig(String dataId, String group, long timeoutMs) throws ConfigExce
ption
```

请求参数

参数名	参数类型	描述
dataId	string	配置ID，采用类似package.class（如com.taobao.tc.refund.log.level）的命名规则保证全局唯一性，class部分建议是配置的业务含义。 全部字符小写。只允许英文字符和4种特殊字

参数名	参数类型	描述
		符（".”、":”、"-"、"_"）。不超过256字节。
group	string	配置分组，建议填写产品名#模块名（如DNCS:Test）保证唯一性。 只允许英文字符和4种特殊字符（".”、":”、"-"、"_"），不超过128字节。
timeout	string	读取配置超时时间，单位ms，推荐值3000。

返回值

参数类型	描述
string	配置值

请求示例

```
try {
    // 初始化配置服务，控制台通过示例代码自动获取下面参数
    ConfigService.init("${endpoint}", "${namespace}", "${accessKey}", "${secretKey}");
    // 主动获取配置
 String content = ConfigService.getConfig("${dataId}", "${group}", 3000);
 System.out.println(content);
} catch (ConfigException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

异常说明

读取配置超时或网络异常，抛出ConfigException异常。

3.2.2 Spring Cloud SDK接入

3.2.2.1 环境准备

Spring Cloud SDK的使用步骤如下。

1. 增加Maven依赖。

```
<dependency>
    <groupId>com.alibaba.cloud</groupId>
    <artifactId>spring-cloud-starter-acm</artifactId>
    <version>1.0.2</version>
```

```
</dependency>
```

2. 配置应用名和应用组。

   在Spring Boot应用中配置application.properties，配置spring.application.group和spring.application.name：

```
spring.application.group=com.alibaba.cloud.acm
spring.application.name=sample-app
```

3. 配置DNCS环境和认证信息。

   在Spring Boot应用中配置application.properties，配置alibaba.acm.endpoint、alibaba.acm.namespace、alibaba.acm.accessKey和alibaba.acm.secretKey：

```
spring.application.group=com.alibaba.cloud.acm
spring.application.name=sample-app

alibaba.acm.endpoint=xxx
alibaba.acm.namespace=xxx
alibaba.acm.accessKey=xxx
alibaba.acm.secretKey=xxx
```

4. 在控制台添加应用配置。

   前往控制台，在相应的空间（namespace）下新建配置。

   - Data ID按照如下约定格式编写：

     ${spring.application.group}:${spring.application.name}.properties

     例如：com.alibaba.cloud.acm:sample-app.properties

   - 配置格式选择**TEXT**，配置内容即为想要注入到应用中的具体key-value：

```
user.id = 001
user.name = juven2
user.age = 88
```

3.2.2.2 配置注入

使用Spring MVC注解注入配置，降低使用配置成本。

可以直接使用@Value注入配置：

```
@Component
class SampleRunner implements ApplicationRunner {

    @Value("${user.id}")
    String userId;

    @Value("${user.name}")
    String userName;
```

```
    @Value("${user.age}")
    int userAge;

    @Override
    public void run(ApplicationArguments args){
        System.out.println(userId);
        System.out.println(userName);
        System.out.println(userAge);
    }
}
```

说明：

如果同时在Spring Boot应用的application.properties和DNCS的${spring.application.group}:${spring.application.name}.properties中配置了同一个key，DNCS中的value会覆盖应用默认的value。

3.2.2.3 Spring Boot集成

健康检查

Spring Cloud ACM集成了Spring Boot的*Health Check*。访问health endpoint可以看到Spring Boot应用是否正确连接了ACM服务器：

```
{
  "status": "UP",
  "acm": {
    "status": "UP",
    "dataIds": [
      "com.alibaba.cloud.acm:sample-app.properties"
    ]
  },
  "diskSpace": {
    "status": "UP",
    "total": 1000240963584,
    "free": 858827423744,
    "threshold": 10485760
  },
  "refreshScope": {
    "status": "UP"
  }
}
```

@RefreshScope

Spring Cloud ACM也支持Spring Cloud的*Refresh Scope*特性。

标记了@RefreshScope注解的Bean会自动监听ACM服务端的变更，并在运行时自动更新配置。代码示例如下：

```
@RestController
@RequestMapping("/sample")
```

```
@RefreshScope
class SampleController {

    @Value("${user.name}")
    String userName;

    @RequestMapping("/acm")
    public String simple() {
        return "Hello Spring Cloud ACM!" + " Hello " + userName + "!";
    }
}
```

使用了@ConfigurationProperties注解的Bean默认就是Refresh Scope的。

ACM Endpoint

Spring Cloud ACM内置了一个名为acm的endpoint，您可以通过访问Spring Boot应用management.port（默认为8080）下的/acm访问，如：http://localhost:8080/acm

```
{
  "runtime": {
    "sources": [
      {
        "dataId": "com.alibaba.cloud.acm:sample-app.properties",
        "lastSynced": "2017-10-10 10:46:27"
      }
    ],
    "refreshHistory": [
      {
        "timestamp": "2017-10-10 10:46:24",
        "dataId": "com.alibaba.cloud.acm:sample-app.properties",
        "md5": "8692ae986ec7bc345b3f0f4de602ff13"
      }
    ]
  },
  "config": {
    "group": "DEFAULT_GROUP",
    "timeOut": 3000,
    "endpoit": "xxx",
    "namespace": "xxx",
    "accessKey": "xxx",
    "secretKey": "xxx"
  }
}
```

4 常见问题

4.1 DNS-F无法启动

1. 检查jmenv.tbsite.net解析是否正确。可以通过命令curl "http://jmenv.tbsite.net:8080/vipserver/serverlist"检查返回的结果是否和对应环境下的机器列表一致。
2. 检查/etc/resolv.con文件内容是否正确。

   查看在该文件中，首选的nameserver是否为本机。如果不是，则在文件中添加“nameserver 127.0.0.1”，且确保该行在所有包含“nameserver”的行中的首行。
3. 查看端口是否被占用。

   运行命令sudo netstat -anup | grep :53，查看端口是否被占用。如果被占用，则需要停止占用进程。
4. 查看是否已安装依赖。

   执行命令sudo /home/tops/bin/python -u /home/a/vipsrv-dns/main.py，如果提示如下信息: Traceback (most recent call last): File "/home/a/vipsrv-dns/main.py", line 11, in <module> from vipsrv import VIPSrv File "/home/a/vipsrv-dns/vipsrv.py", line 7, in <module> import requests ImportError: No module named requests

   则需要运行以下命令安装tops-python27-simplejson与tops-python27-requests

   sudo yum install tops-python27-simplejson -y

   sudo yum install tops-python27-requests -y

4.2 域名已在vipserver ops上注册，Java客户端无法获取IP

首先，查看client的日志，看是否报错，检查jmenv解析是否正确，检查方法同*DNS-F无法启动*。如果执行上述操作仍然不能解决问题，请联系开发人员获取帮助。

4.3 vipserver的容灾策略是怎样的？

vipserver的容灾策略分为客户端和服务端两部分。

在客户端，vipserver会把请求过的域名信息写入磁盘中，客户端启动时会首先将这些磁盘中的文件加载到内存中。这样可以保证在服务器不可用时，客户端仍然能够提供服务。

在服务端，容灾主要通过Diamond实现，即所有域名的信息都会以文件的形式存储在本地磁盘。此外，当某个域名下挂载的IP只剩一个时，不可以把该IP从域名中删掉，这也是容灾策略之一。