



用户手册

金蝶Apusic分布式配置中心 for Apollo V1.0

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

- 1 前言
 - 1.1 适用对象
 - 1.2 相关文档
 - 1.3 技术支持
- 2 名词解释
- 3 普通应用接入指南
 - 3.1 创建项目
 - 3.2 项目权限分配
 - 3.2.1 项目管理员权限
 - 3.2.2 配置编辑、发布权限
 - 3.3 添加配置项
 - 3.3.1 通过表格模式添加配置
 - 3.3.2 通过文本模式编辑
 - 3.4 发布配置
 - 3.5 应用读取配置
 - 3.6 回滚已发布配置
- 4 公共组件接入指南
 - 4.1 公共组件和普通应用的区别
 - 4.2 公共组件接入步骤
 - 4.2.1 创建Namespace
 - 4.2.2 添加配置项
 - 4.2.2.1 通过表格模式添加配置
 - 4.2.2.2 通过文本模式编辑
 - 4.2.3 发布配置
 - 4.2.4 应用读取配置
 - 4.3 应用覆盖公用组件配置步骤
 - 4.3.1 关联公共组件Namespace
 - 4.3.2 覆盖公用组件配置
 - 4.3.3 发布配置
- 5 集群独立配置说明
 - 5.1 创建集群
- 6 多个AppId使用同一份配置

•7 灰度发布使用指南

•7.1 场景介绍

•7.2 创建灰度

•7.3 灰度配置

•7.4 配置灰度规则

•7.5 灰度发布

•7.6 全量发布

•7.7 放弃灰度

•7.8 发布历史

•8 其它功能配置

•8.1 配置查看权限

•8.2 配置访问密钥

•9 最佳实践

•9.1 安全相关

•9.1.1 认证

•9.1.2 授权

•9.1.3 系统访问

1 前言

本文档为金蝶Apusic分布式配置中心for Apollo（ADCC for Apollo）V1.0的用户手册，详细介绍了 ADCC for Apollo产品的功能使用、配置方法及管理操作等内容。

1.1 适用对象

本文档适用于ADCC for Apollo产品运维工程师、IT系统运维工程师、开发工程师等人员。

1.2 相关文档

了解更多ADCC for Apollo V1.0产品相关的信息，请参阅以下ADCC for Apollo V1.0产品手册文档集：

序号	手册文档	说明
1	金蝶Apusic分布式配置中心for Apollo V1.0 安装手册	详细介绍如何在各操作系统上安装ADCC for Apollo，以及服务组件实例启停操作，产品的注册过程。
2	金蝶Apusic分布式配置中心for Apollo V1.0 用户手册	详细介绍 ADCC 相关功能的使用、配置、管理及配套工具的使用方法。
3	金蝶Apusic分布式配置中心for Apollo V1.0 开发手册	详细介绍基于各种开发语言进行ADCC for Apollo客户端应用开发的说明。
4	金蝶Apusic分布式配置中心for Apollo V1.0 API开放平台手册	详细介绍ADCC for Apollo的标准API。
5	金蝶Apusic分布式配置中心for Apollo V1.0 常见问题	详细介ADCC for Apollo的常见问题及处理方法。

1.3 技术支持

ADCC for Apollo产品提供全面的技术支持服务，您可以通过以下方式获得技术支持：

- 网址：www.apusic.com
- 电话：400-855-5800
- 邮箱：support@apusic.com
- 金蝶云社区：<https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技术支持时，请提供如下信息：

- 您的姓名
- 公司与联系方式
- 操作系统及其版本
- 产品版本号
- 出现异常及错误的日志、截图等详细信息

2 名词解释

- 普通应用
 - 普通应用指的是独立运行的程序，如
 - Web应用程序
 - 带有main函数的程序
- 公共组件
 - 公共组件指的是发布的类库、客户端程序，不会自己独立运行，如
 - Java的jar包
 - .Net的dll文件

3 普通应用接入指南

3.1 创建项目

要使用ADCC for Apollo，第一步需要创建项目。

1. 打开apollo-portal主页
2. 点击"创建项目"



3. 输入项目信息

- 部门：选择应用所在的部门
- 应用AppId：用来标识应用身份的唯一id，格式为string，需要和客户端app.properties中配置的app.id对应
- 应用名称：应用名，仅用于界面展示
- 应用负责人：选择的人默认会成为该项目的管理员，具备项目权限管理、集群创建、Namespace创建等权限

创建项目

* 部门

* 应用AppId
(CMS上申请的App Id)

* 应用名称
(建议格式 xx-yy-zz 例:apollo-server)

* 应用负责人
(负责人具有项目管理的最高权限,比如分配配置的修改权,发布权等)

4. 点击提交

创建成功后，会自动跳转到项目首页

环境列表 ?

DEV

项目信息

AppId: 100003806
应用名: hermes-portal
部门: 框架(FX)
负责人: jhliang
邮箱: jhliang@ctrip.com

管理项目

添加集群

添加Namespace

私有

application properties

发布 回滚 发布历史 授权 灰度 设置

表格 文本 更改历史 实例列表

过滤配置 同步配置 新增配置

Tips: 此namespace从来没有发布过，Apollo客户端将获取不到配置并记录404日志信息，请及时发布。

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
------	--------	-------	----	----------	-----------	----

3.2 项目权限分配

3.2.1 项目管理员权限

项目管理员拥有以下权限：

1. 可以管理项目的权限分配
2. 可以创建集群
3. 可以创建Namespace

创建项目时填写的应用负责人默认会成为项目的管理员之一，如果还需要其他人也成为项目管理员，可以按照下面步骤操作：

1. 点击页面左侧的"管理项目"



2. 搜索需要添加的成员并点击添加

权限管理(AppId:100003806)

管理员
(可以创建Namespace
可以创建集群
可以分配用户权限)

添加

song_s

song_s | vss宋顺 | song_s@ctrip.com



应用ID/应用名

✓ 添加成功

权限管理(AppId:100003806)

返回

管理员
(可以创建Namespace
可以创建集群
可以分配用户权限)

添加

jhliang



song_s



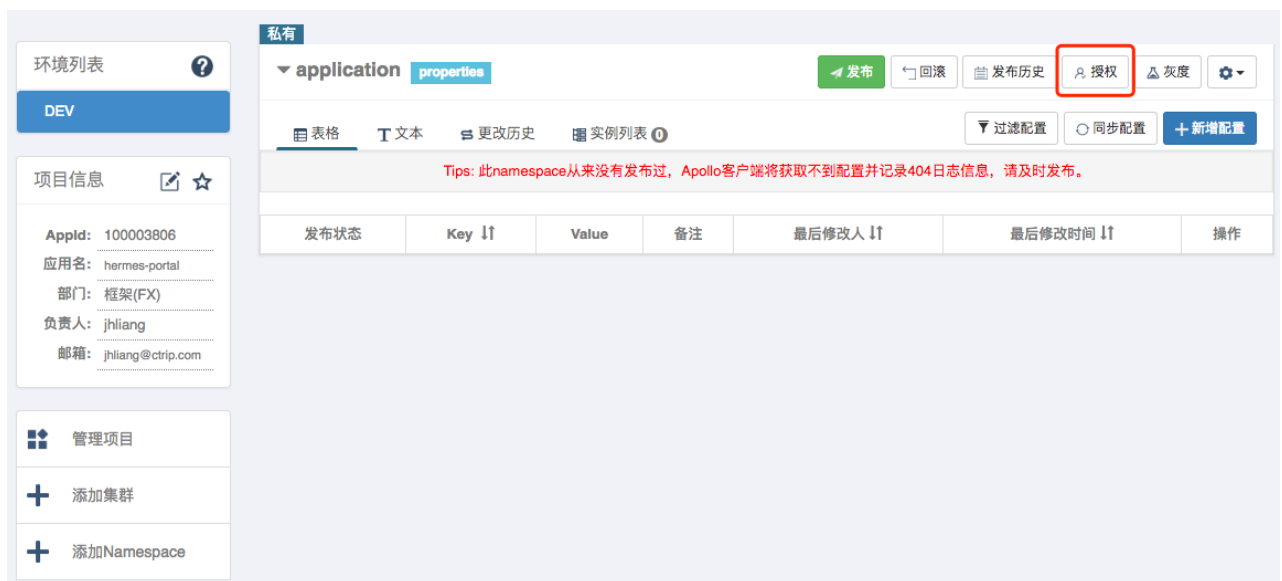
3.2.2 配置编辑、发布权限

配置权限分为编辑和发布：

- 编辑权限允许用户在Apollo界面上创建、修改、删除配置
 - 配置修改后只在Apollo界面上变化，不会影响到应用实际使用的配置
- 发布权限允许用户在Apollo界面上发布、回滚配置
 - 配置只有在发布、回滚动作后才会被应用实际使用到
 - Apollo在用户操作发布、回滚动作后实时通知到应用，并使最新配置生效

项目创建完，默认没有分配配置的编辑和发布权限，需要项目管理员进行授权。

1. 点击application这个namespace的授权按钮



2. 分配修改权限

修改权
(可以修改配置)

song_s | vss宋顺 | song_s@ctrip.com

添加

3. 分配发布权限

发布权
(可以发布配置)

song_s | vss宋顺 | song_s@ctrip.com

添加

3.3 添加配置项

编辑配置需要拥有这个Namespace的编辑权限, 如果发现没有新增配置按钮, 可以找项目管理员授权。

3.3.1 通过表格模式添加配置

1. 点击新增配置

application

发布

回滚

发布历史

授权

同步配置

表格

文本

更改历史

+ 新增配置

2. 输入配置项

添加配置项

* Key

portal.elastic.cluster.name

* Value

hermes-es-fws

Comment

elastic cluster name

* 选择集群

☐ 环境

集群

☒ DEV

default

关闭

提交

3. 点击提交

application 有修改 1

发布

回滚

发布历史

授权

同步配置

表格

文本

更改历史

+ 新增配置

Key	Value	备注	最后修改人	最后修改时间	操作
portal.elastic.cluster.name	hermes-es-fws	elastic cluster name	song_s	2016-06-28 15:27:37	

3.3.2 通过文本模式编辑

Apollo除了支持表格模式，逐个添加、修改配置外，还提供文本模式批量添加、修改。这个对于从已有的properties文件迁移尤其有用。

1. 切换到文本编辑模式



2. 点击右侧的修改配置按钮



3. 输入配置项，并点击提交修改



3.4 发布配置

配置只有在发布后才会真的被应用使用到，所以在编辑完配置后，需要发布配置。

发布配置需要拥有这个Namespace的发布权限，如果发现没有发布按钮，可以找项目管理员授权。

1. 点击"发布按钮"



The screenshot shows the Apollo configuration interface. At the top, there is a navigation bar with 'application', 'properties', and '有修改 2'. To the right of the navigation bar is a green '发布' (Release) button, which is highlighted with a red box. Other buttons include '回滚' (Rollback), '发布历史' (Release History), '授权' (Authorization), and '同步配置' (Sync Configuration). Below the navigation bar, there are tabs for '表格' (Table), '文本' (Text), '更改历史' (Change History), and '实例列表' (Instance List). On the right side of the tabs, there are buttons for '过滤配置' (Filter Configuration) and '新增配置' (Add Configuration). The main content area displays a table with configuration items.

Key	Value	备注	最后修改人	最后修改时间	操作
portal.elastic.cluster.name	hermes-es-fws		song_s	2016-09-12 13:40:13	
portal.elastic.document.type	biz		song_s	2016-09-12 13:42:07	

2. 填写发布相关信息，点击发布



The screenshot shows the '发布' (Release) dialog box. It contains a table of changes, a 'Release Name' field, a 'Comment' field, and '关闭' (Close) and '发布' (Release) buttons.

Changes	Key	Old Value	New Value	最后修改人	最后修改时间
	portal.elastic.cluster.name		hermes-es-fws	song_s	2016-09-12 13:40:13
	portal.elastic.document.type		biz	song_s	2016-09-12 13:42:07

* Release Name: 发布demo

Comment: 增加es相关配置

关闭 发布

3.5 应用读取配置

配置发布成功后，应用就可以通过Apollo客户端读取到配置了。

Apollo目前提供Java客户端，具体信息请点击[Java客户端使用文档](#)：

如果应用使用了其它语言，也可以通过直接访问Http接口获取配置，具体可以参考[其它语言客户端接入指南](#)

3.6 回滚已发布配置

如果发现已发布的配置有问题，可以通过点击『回滚』按钮来将客户端读取到的配置回滚到上一个发布版本。

这里的回滚机制类似于发布系统，发布系统中的回滚操作是将部署到机器上的安装包回滚到上一个部署的版本，但代码仓库中的代码是不会回滚的，从而开发可以在修复代码后重新发布。

Apollo中的回滚也是类似的机制，点击回滚后是将发布到客户端的配置回滚到上一个已发布版本，也就是说客户端读取到的配置会恢复到上一个版本，但页面上编辑状态的配置是不会回滚的，从而开发可以在修复配置后重新发布。

4 公共组件接入指南

4.1 公共组件和普通应用的区别

公共组件是指那些发布给其它应用使用的客户端代码，比如CAT客户端、Hermes Producer客户端等。

虽然这类组件是由其他团队开发、维护，但是运行时是在业务实际应用内的，所以本质上可以认为是应用的一部分。

通常情况下，这类组件所用到的配置由原始开发团队维护，不过由于实际应用的运行时、环境各不一样，所以我们也允许应用在实际使用时能够覆盖公共组件的部分配置。

4.2 公共组件接入步骤

公共组件的接入步骤，和普通应用几乎一致，唯一的区别是公共组件需要创建自己唯一的Namespace。

所以，首先执行普通应用接入文档中的以下几个步骤，然后再按照本章节后面的步骤操作。

1. [创建项目](#)
2. [项目管理员权限](#)

4.2.1 创建Namespace

创建Namespace需要项目管理员权限，如果发现没有添加Namespace按钮，可以找项目管理员授权。

1. 点击页面左侧的添加Namespace

项目信息

AppId: 100003806

应用名: hermes-portal

部门: 框架(FX)

负责人: jhliang

邮箱: jhliang@ctrip.com

管理项目

添加集群

添加Namespace

○

2. 点击"创建新的Namespace"

新建Namespace

关联已存在的Namespace

创建新的Namespace

应用ID

100003806

* 名称

FX.

FX.

备注

提交

○

3. 输入公共组件的Namespace名称，需要注意的是Namespace名称全局唯一

- Apollo会默认把部门代号添加在最前面

新建Namespace

关联已存在的Namespace

创建新的Namespace

应用ID

100003806

* 名称

FX.

Hermes.Producer

FX.Hermes.Producer

备注

Hermes producer相关配置

提交

o

4. 点击提交后，页面会自动跳转到关联Namespace页面

- o 首先，选中所有需要有这个Namespace的环境和集群，一般建议全选
- o 其次，选中刚刚创建的namespace
- o 最后，点击提交

新建Namespace

关联已存在的Namespace

创建新的Namespace

应用ID

100003806

* 选择集群

☒ 环境

集群

☒ DEV

default

* namespace

FX.Hermes.Producer

提交

o

5. 关联成功后，页面会自动跳转到Namespace权限管理页面

1. 分配修改权限

修改权
(可以修改配置)

vss

song_s | vss宋顺 | song_s@ctrip.com

添加

2. 分配发布权限

发布权
(可以发布配置)

vss

song_s | vss宋顺 | song_s@ctrip.com

添加

6. 点击"返回"回到项目页面

4.2.2 添加配置项

编辑配置需要拥有这个Namespace的编辑权限，如果发现没有新增配置按钮，可以找项目管理员授权。

4.2.2.1 通过表格模式添加配置

1. 点击新增配置

FX.Hermes.Producer

发布

回滚

发布历史

授权

同步配置

表格

文本

更改历史

新增配置

添加配置项

×

* Key

sender.batchSize

* Value

500

Comment

批量发送大小

* 选择集群

☐ 环境
 集群

☒ DEV
 default

关闭

提交

2. 输入配置项

3. 点击提交

FX.Hermes.Producer 有修改 1		发布	回滚	发布历史	授权	同步配置
表格	文本	更改历史	+ 新增配置			
Key	Value	备注	最后修改人	最后修改时间	操作	
sender.batchSize	500	批量发送大小	song_s	2016-06-28 16:18:31	✎ ✕	

4.2.2.2 通过文本模式编辑

这部分和普通应用一致，具体步骤请参见[通过文本模式编辑](#)。

4.2.3 发布配置

配置只有在发布后才会真的被应用使用到，所以在编辑完配置后，需要发布配置。

发布配置需要拥有这个Namespace的发布权限，如果发现没有发布按钮，可以找项目管理员授权。

1. 点击"发布按钮"

FX.Hermes.Producer 有修改 1 发布 回滚 发布历史 授权 同步配置

表格 文本 更改历史 新增配置

Key	Value	备注	最后修改人	最后修改时间	操作
sender.batchSize	500	批量发送大小	song_s	2016-06-28 16:21:33	

2. 填写发布相关信息，点击发布

发布 ×

Changes:	Key	Old Value	New Value	最后修改人	最后修改时间
	sender.batchSize		500	song_s	2016-06-28 16:21:33

• Release Name:

Comment:

关闭 发布

4.2.4 应用读取配置

配置发布成功后，应用就可以通过Apollo客户端读取到配置了。

Apollo目前提供Java客户端，具体信息请点击[Java客户端使用文档](#)：

如果应用使用了其它语言，也可以通过直接访问Http接口获取配置，具体可以参考[其它语言客户端接入指南](#)

对于公共组件的配置读取，可以参考上述文档中的"获取公共Namespace的配置"部分。

4.3 应用覆盖公用组件配置步骤

前面提到，通常情况下，公共组件所用到的配置由原始开发团队维护，不过由于实际应用的运行时、环境各不一样，所以我们也允许应用在实际使用时能够覆盖公共组件的部分配置。

这里就讲一下应用如何覆盖公用组件的配置，简单起见，假设apollo-portal应用使用了hermes producer客户端，并且希望调整hermes的批量发送大小。

4.3.1 关联公共组件Namespace

1. 进入使用公共组件的应用项目首页，点击左侧的添加Namespace按钮

- 所以，在这个例子中，我们需要进入apollo-portal的首页。
- （添加Namespace需要项目管理员权限，如果发现没有添加Namespace按钮，可以找项目管理员授权）



2. 找到hermes producer的namespace，并选择需要关联到哪些环境和集群

新建Namespace

关联已存在的Namespace 创建新的Namespace

应用ID 100003173

* 选择集群

☒	环境	集群
☒	DEV	default

* namespace

FX.Hermes.Producer

提交

3. 关联成功后，页面会自动跳转到Namespace权限管理页面

1. 分配修改权限

修改权
(可以修改配置)

vss

song_s | vss宋顺 | song_s@ctrip.com

添加

2. 分配发布权限

发布权
(可以发布配置)

vss

song_s | vss宋顺 | song_s@ctrip.com

添加

4. 点击"返回"回到项目页面

4.3.2 覆盖公用组件配置

1. 点击新增配置

FX.Hermes.Producer

发布

回滚

发布历史

授权

同步配置

表格

文本

更改历史

+ 新增配置

2. 输入要覆盖的配置项

添加配置项 ✕

Key

sender.batchSize

Value

1000

Comment

apollo-portal自定义的hermes批量发送大小

选择集群

☐ 环境

集群

☒ DEV

default

关闭

提交

3. 点击提交

FX.Hermes.Producer

有修改 1

发布

回滚

发布历史

授权

同步配置

表格

文本

更改历史

新增配置

Key	Value	备注	最后修改人	最后修改时间	操作
sender.batchSize	1000	apollo-portal自定义的hermes批量发送大小	song_s	2016-06-28 16:35:33	

4.3.3 发布配置

配置只有在发布后才会真的被应用使用到，所以在编辑完配置后，需要发布配置。

发布配置需要拥有这个Namespace的发布权限，如果发现没有发布按钮，可以找项目管理员授权。

23

1. 点击"发布按钮"

FX.Hermes.Producer 有修改 1 发布 回滚 发布历史 授权 同步配置

表格 文本 更改历史 新增配置

Key	Value	备注	最后修改人	最后修改时间	操作
sender.batchSize	1000	apollo-portal自定义的hermes批量发送大小	song_s	2016-06-28 16:35:33	 

2. 填写发布相关信息，点击发布

发布

✕

Changes:

Key	Old Value	New Value	最后修改人	最后修改时间
sender.batchSize		1000	song_s	2016-06-28 16:35:33

Release Name:

2016-06-28 16:34:04 覆盖公共组件发布Demo

Comment:

apollo-portal覆盖hermes发送批量大小

关闭

发布

3. 配置发布成功后，hermes producer客户端在apollo-portal应用里面运行时读取到的sender.batchSize的值就是1000。

5 集群独立配置说明

在有些特殊情况下，应用有需求对不同的集群做不同的配置，比如部署在A机房的应用连接的es服务器地址和部署在B机房的应用连接的es服务器地址不一样。

在这种情况下，可以通过在Apollo创建不同的集群来解决。

5.1 创建集群

创建集群需要项目管理员权限，如果发现没有添加集群按钮，可以找项目管理员授权。

1. 点击页面左侧的"添加集群"按钮



2. 输入集群名称，选择环境并提交

创建集群

* 应用AppId 100003173

* 集群名称

SHAJQ

(部署集群如:SHAJQ,SHAOY 或自定义集群如:SHAJQ-xx,SHAJQ-yy)

* 选择环境



DEV

提交

o

3. 切换到对应的集群，修改配置并发布即可

DEV

default

SHAJQ

应用信息

AppId: 100003173
应用名: apollo-portal
部门: 框架(FX)

application

表格 文本 更改历史

Key	Value
requestTimeout	200

o

4. 通过上述配置，部署在SHAJQ机房的应用就会读到SHAJQ集群下的配置

5. 如果应用还在其它机房部署了应用，那么在上述的配置下，会读到default集群下的配置。

6 多个Appld使用同一份配置

在一些情况下，尽管应用本身不是公共组件，但还是需要在多个Appld之间共用同一份配置，比如同一个产品的不同项目：XX-Web, XX-Service, XX-Job等。

这种情况下如果希望实现多个Appld使用同一份配置的话，基本概念和公共组件的配置是一致的。

具体来说，就是在其中一个Appld下创建一个namespace，写入公共的配置信息，然后在各个项目中读取该namespace的配置即可。

如果某个Appld需要覆盖公共的配置信息，那么在该Appld下关联公共的namespace并写入需要覆盖的配置即可。

具体步骤可以参考[公共组件接入指南](#)。

7 灰度发布使用指南

通过灰度发布功能，可以实现：

1. 对于一些对程序有较大影响的配置，可以先在一个或者多个实例生效，观察一段时间没问题后再全量发布配置。
2. 对于一些需要调优的配置参数，可以通过灰度发布功能来实现A/B测试。可以在不同的机器上应用不同的配置，不断调整、测评一段时间后找出较优的配置再全量发布配置。

下面将结合一个实际例子来描述如何使用灰度发布功能。

7.1 场景介绍

100004458(apollo-demo)项目有两个客户端：

1. 10.32.21.19
2. 10.32.21.22

Apollo 配置中心

搜索项目(项目ID、项目名) 帮助 song_s

环境列表

- FAT
- UAT
- PRO
- default
- SHAOY
- SHAJQ

项目信息

AppId: 100004458
应用名: apollo-demo

私有

application properties

发布 回滚 发布历史 授权 创建灰度

表格 文本 更改历史 实例列表

使用最新配置的实例 2 使用非最新配置的实例 1 所有实例 2

初始发布

App ID	Cluster Name	Data Center	IP	配置获取时间
100004458	default		10.32.21.19	2016-11-24 14:22:47
100004458	default		10.32.21.22	2016-11-24 14:22:47

灰度目标：

- 当前有一个配置timeout=2000，我们希望对10.32.21.22灰度发布timeout=3000，对10.32.21.19仍然是timeout=2000。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助

song_s

私有

application properties

发布 回滚 发布历史 授权 创建灰度

表格 文本 更改历史 实例列表 2

过滤配置 同步配置 新增配置

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout	2000	Demo程序会用，勿删	song_s	2016-11-23 18:38:05	✎ ✕

7.2 创建灰度

首先点击application namespace右上角的 创建灰度 按钮。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助

song_s

私有

application properties

发布 回滚 发布历史 授权 创建灰度

过滤配置 同步配置 新增配置

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout	2000	Demo程序会用，勿删	song_s	2016-11-23 18:38:05	✎ ✕

点击确定后，灰度版本就创建成功了，页面会自动切换到 灰度版本 Tab。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助

song_s

私有

主版本 灰度版本

application properties

灰度发布 全量发布 放弃灰度

配置 灰度规则 灰度实例列表 1 更改历史

灰度的配置 新增灰度配置

发布状态	Key ↑↓	主版本的值	灰度的值	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout	2000		Demo程序会用，勿删	song_s	2016-11-23 18:38:05	✎ ✕

主版本的配置

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout	2000	Demo程序会用，勿删	song_s	2016-11-23 18:38:05	✎ ✕

7.3 灰度配置

点击 主版本的配置 中，timeout配置最右侧的 对此配置灰度 按钮

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助 song_s

环境列表

- FAT
- UAT
- PRO
- default
- SHAOY
- SHAJQ

项目信息

AppId: 100004458
应用名: apollo-demo
部门: 框架(FX)

私有

主版本 灰度版本

application properties

灰度发布 全量发布 放弃灰度

配置 灰度规则 灰度实例列表 更改历史

灰度的配置

+ 新增灰度配置

发布状态	Key ↑↓	主版本的值	灰度的值	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
主版本的配置							
已发布	timeout	2000		Demo程序会用，勿删	song_s	2016-11-23 18:38:05	✕
已发布	kibana.url	http://1.1.1.2:5601			song_s	2016-10-18 19:57:29	✕

对此配置灰度

在弹出框中填入要灰度的值：3000，点击提交。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助 song_s

环境列表

- FAT
- UAT
- PRO
- default
- SHAOY
- SHAJQ

项目信息

AppId: 100004458
应用名: apollo-demo
部门: 框架(FX)

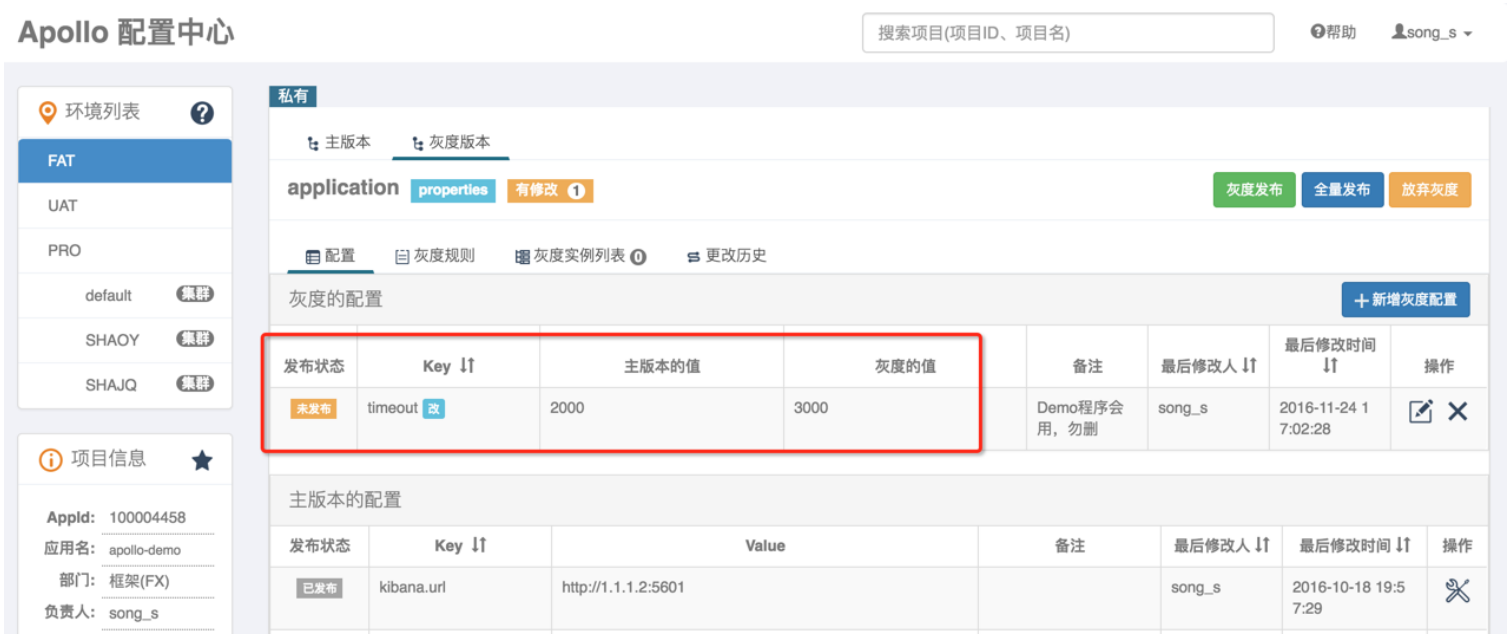
添加灰度配置项

* Key timeout

* Value 3000

Comment Demo程序会用，勿删

取消 提交

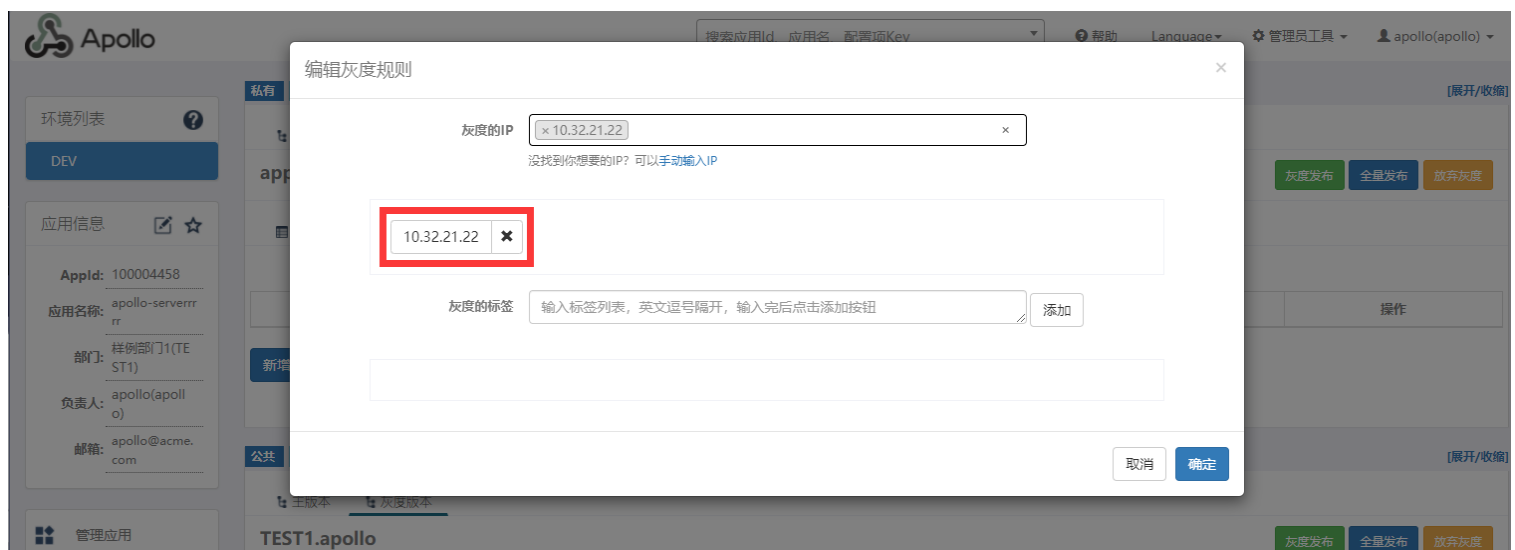
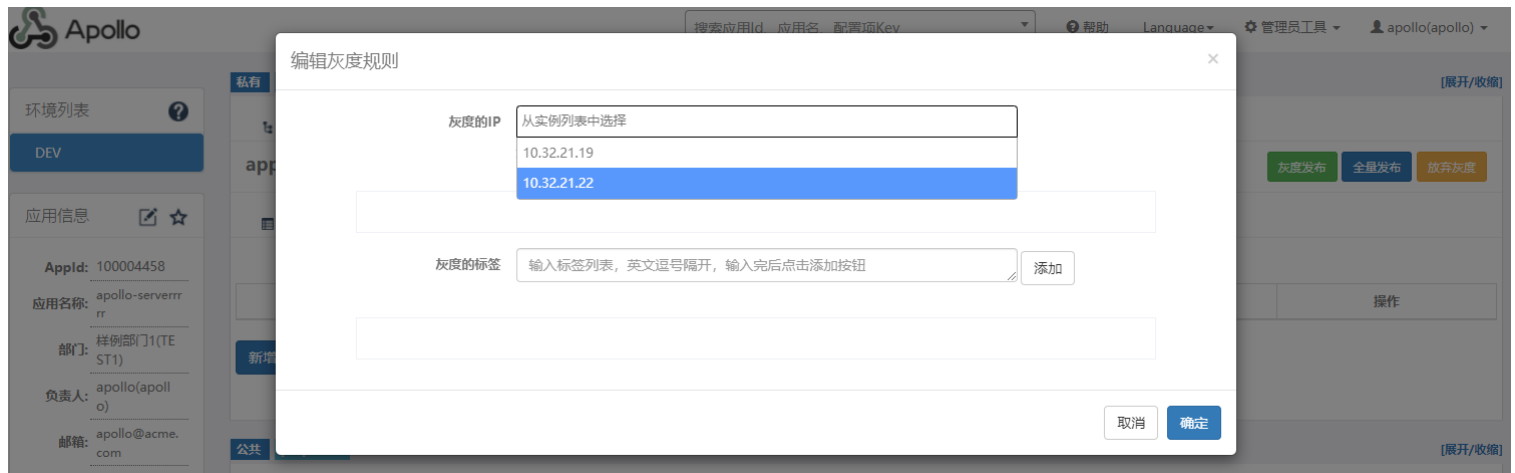


7.4 配置灰度规则

切换到 灰度规则 Tab，点击 新增规则 按钮

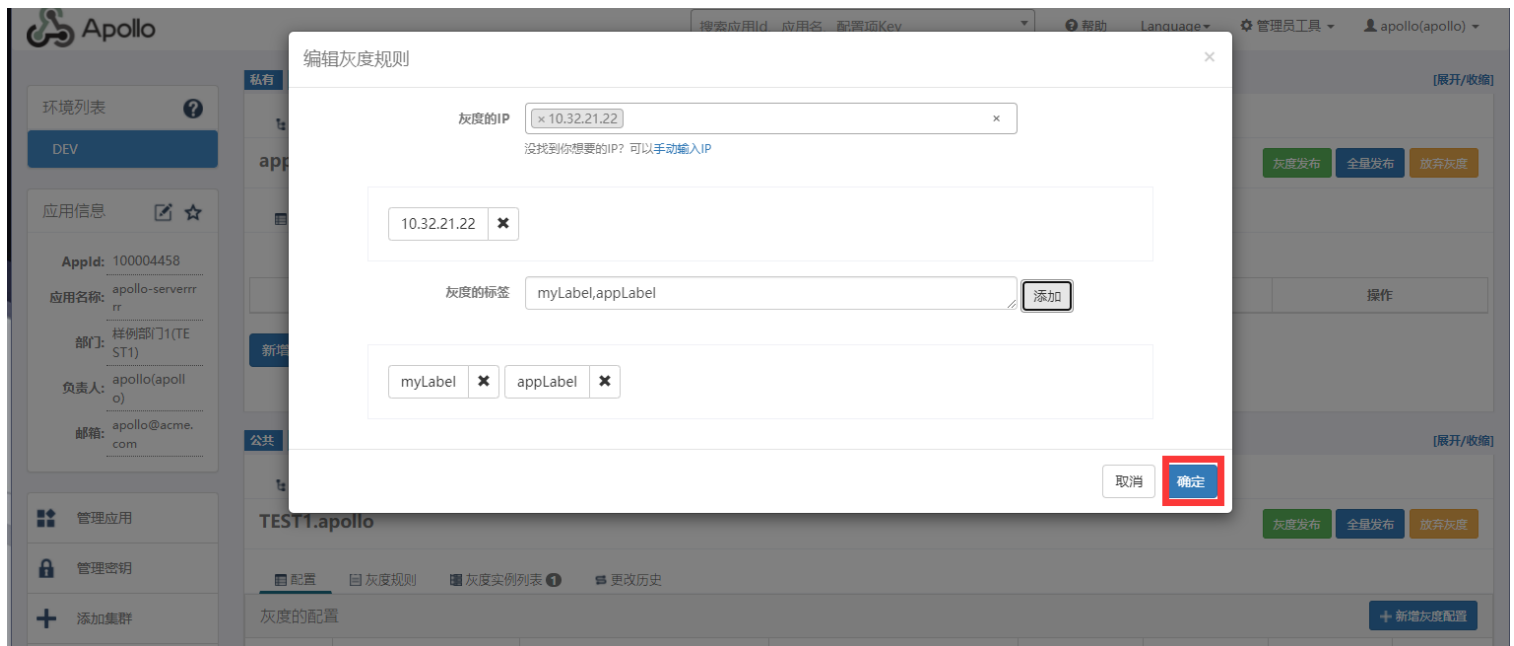
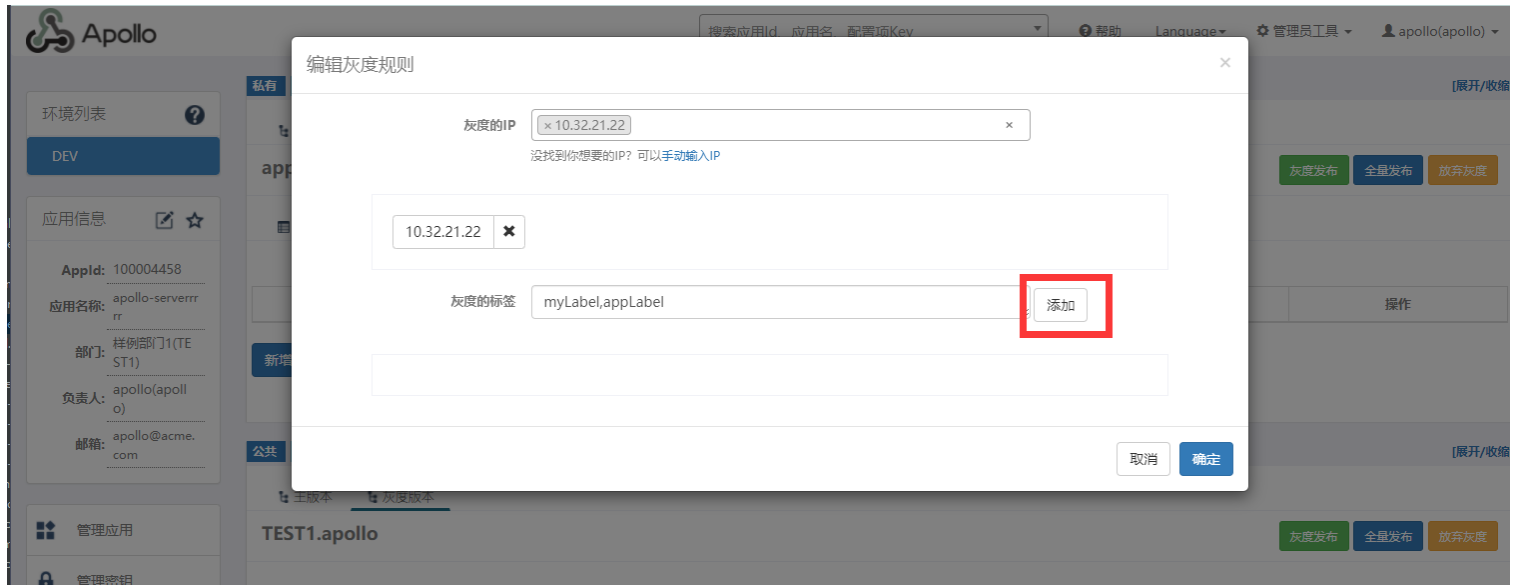


在弹出框中 灰度的IP 下拉框会默认展示当前使用配置的机器列表，选择我们要灰度的IP。



除了IP维度以外，从2.0.0版本开始还支持通过label来标识灰度的实例列表，适用于IP不固定的场景如 Kubernetes。

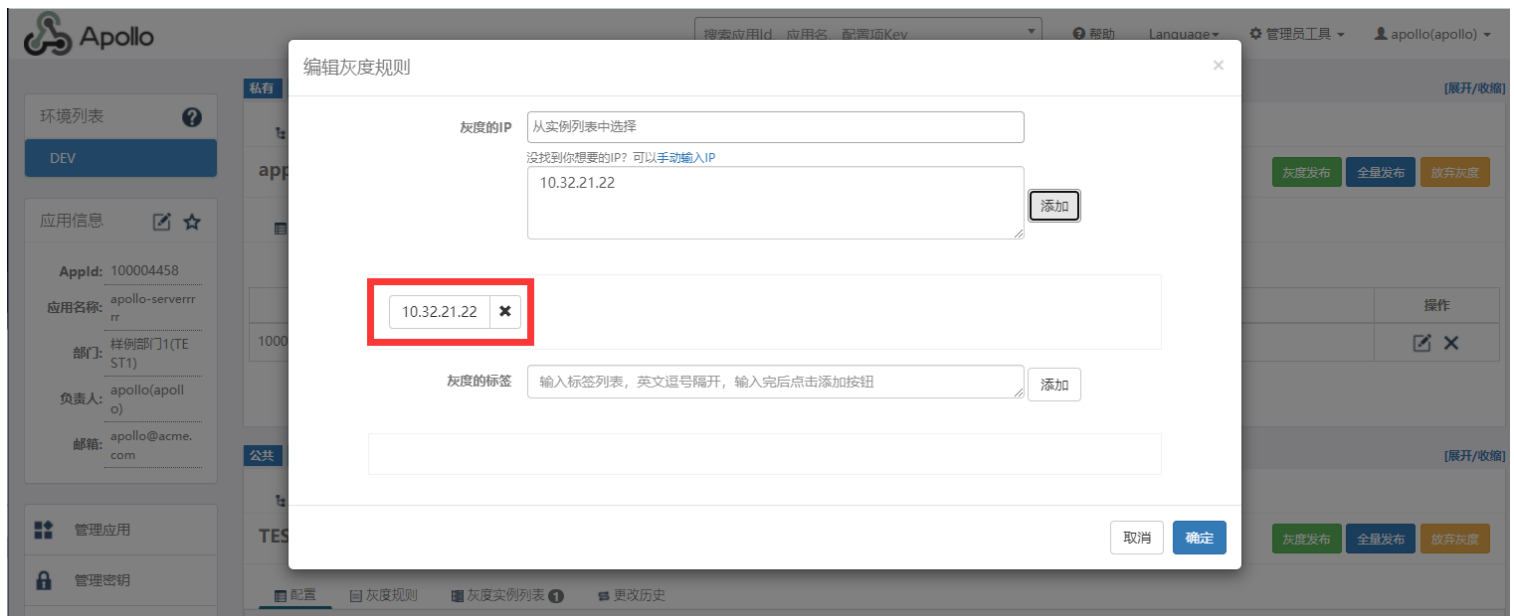
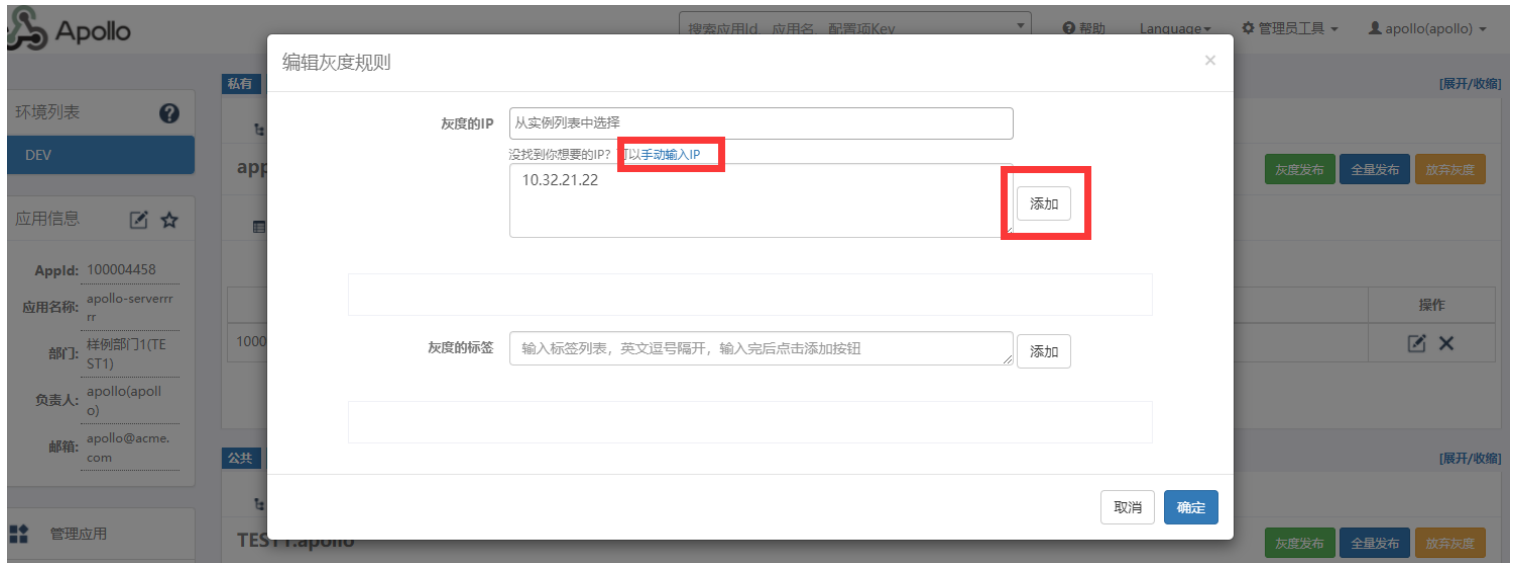
手动输入想要设置的label标签，输入完成后点击添加按钮。



上述规则配置后，灰度配置会对AppId为 100004458，IP为 10.32.21.22 或者 Label 标记为 myLabel 或 appLabel 的实例生效。

关于 Label 如何标记, 可以参考[ApolloLabel](#)的配置说明。

如果下拉框中没找到需要的IP, 说明机器还没从Apollo取过配置, 可以点击手动输入IP来输入, 输入完后点击添加按钮



注: 对于公共Namespace的灰度规则, 需要先指定要灰度的appid, 然后再选择IP和Label。

7.5 灰度发布

配置规则已经生效, 不过灰度配置还没有发布。切换到 配置 Tab。

再次检查灰度的配置部分, 如果没有问题, 点击 灰度发布 。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助

song_s

环境列表

FAT

UAT

PRO

default

SHAQY

SHAQJ

项目信息

私有

主版本

灰度版本

application

properties

有修改 1

灰度发布

继续灰度发布

灰度的配置

发布状态

Key

主版本的值

灰度的值

备注

最后修改人

最后修改时间

操作

主版本的配置

发布状态

Key

Value

备注

最后修改人

最后修改时间

操作

在弹出框中可以看到主版本的值是2000，灰度版本即将发布的值是3000。填入其它信息后，点击发布。

环境列表

FAT

UAT

PRO

default

SHAQY

SHAQJ

项目信息

灰度发布

Changes

Key

主版本值

灰度版本发布的值

灰度版本未发布的值

修改人

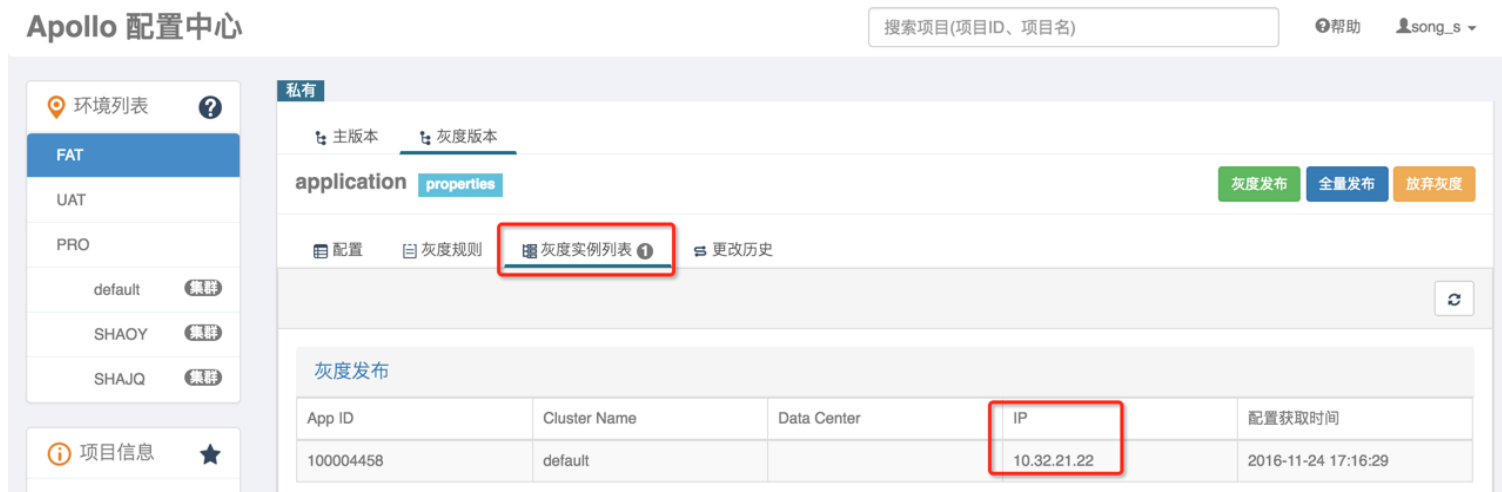
修改时间

Release Name

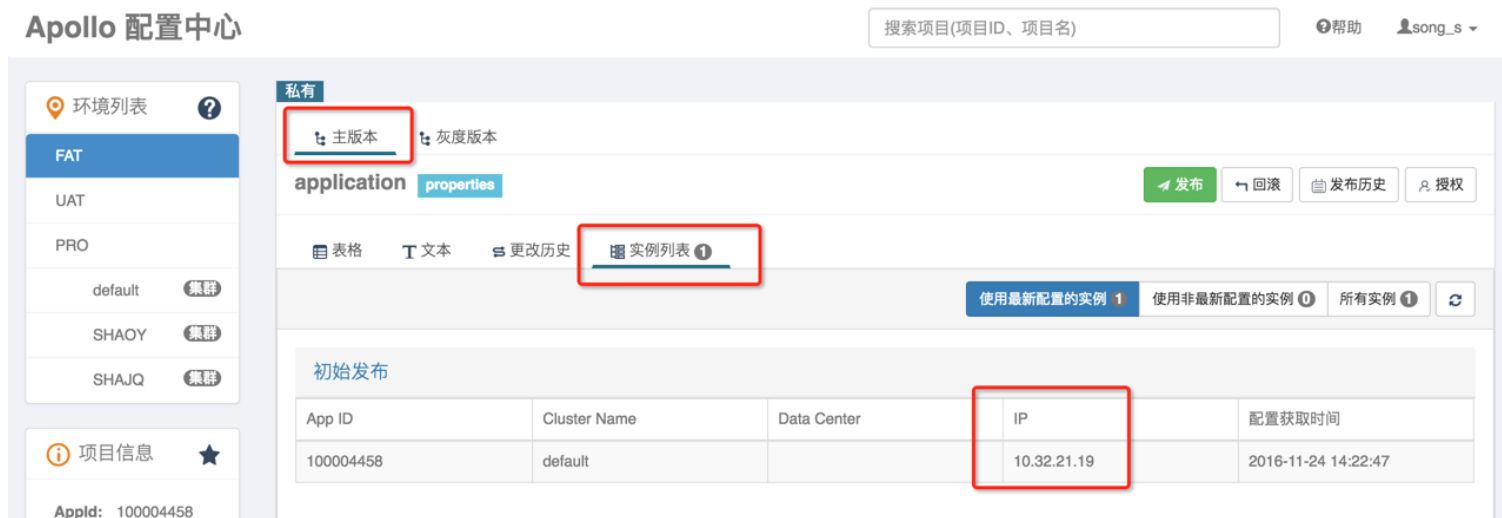
Comment

发布

发布后，切换到 灰度实例列表 Tab，就能看到10.32.21.22已经使用了灰度发布的值。



切换到 主版本 的 实例列表，会看到主版本配置只有10.32.21.19在使用了。



后面可以继续配置的修改或规则的更改。配置的修改需要点击灰度发布后才会生效，规则的修改在规则点击完成后就会实时生效。

7.6 全量发布

如果灰度的配置测试下来比较理想，符合预期，那么就可以操作 全量发布。

全量发布的效果是：

1. 灰度版本的配置会合并回主版本，在这个例子中，就是主版本的timeout会被更新成3000
2. 主版本的配置会自动进行一次发布
3. 在全量发布页面，可以选择是否保留当前灰度版本，默认为不保留。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助

song_s

环境列表

FAT

UAT

PRO

default

SHAQY

SHAJQ

项目信息

AppId: 100004458

应用名: apollo-demo

部门: 框架(FX)

负责人: song_s

私有

主版本 灰度版本

application properties

灰度发布 全量发布 放弃灰度

合并到主版本并发布主版本配置

配置 灰度规则 灰度实例列表 更改历史

灰度的配置

+ 新增灰度配置

发布状态	Key ↑↓	主版本的值	灰度的值	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout 改	2000	3000	Demo程序会用, 勿删	song_s	2016-11-24 17:02:28	

主版本的配置

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	kibana.url	http://1.1.1.2:5601		song_s	2016-10-18 19:57:29	

环境列表

FAT

UAT

PRO

default

SHAQY

SHAJQ

项目信息

AppId: 100004458

应用名: apollo-demo

部门: 框架(FX)

负责人: song_s

私有

主版本 灰度版本

application properties

灰度发布 全量发布 放弃灰度

合并到主版本并发布主版本配置

配置 灰度规则 灰度实例列表 更改历史

灰度的配置

+ 新增灰度配置

发布状态	Key ↑↓	主版本的值	灰度的值	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout 改	2000	3000	Demo程序会用, 勿删	song_s	2016-11-24 17:02:28	

主版本的配置

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	kibana.url	http://1.1.1.2:5601		song_s	2016-10-18 19:57:29	

全量发布

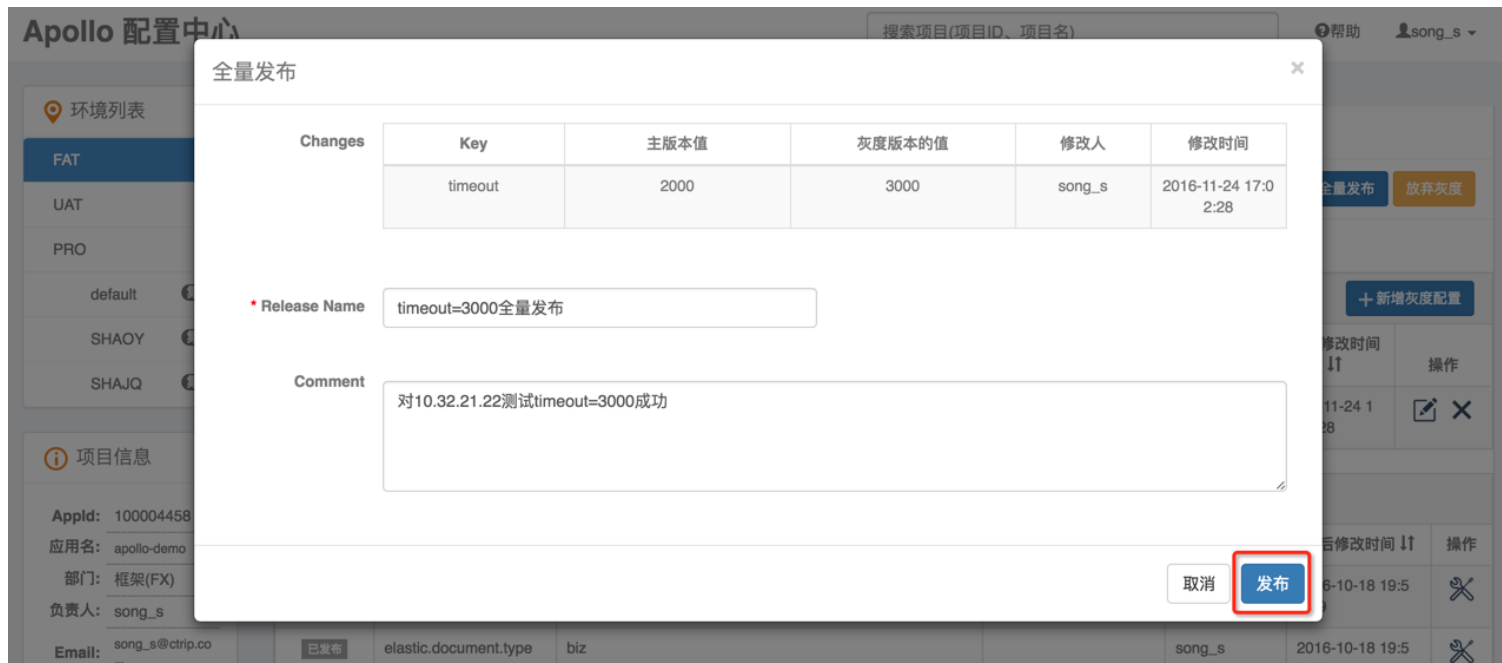
全量发布将会把灰度版本的配置合并到主分支, 并发布。

全量发布后, 您希望

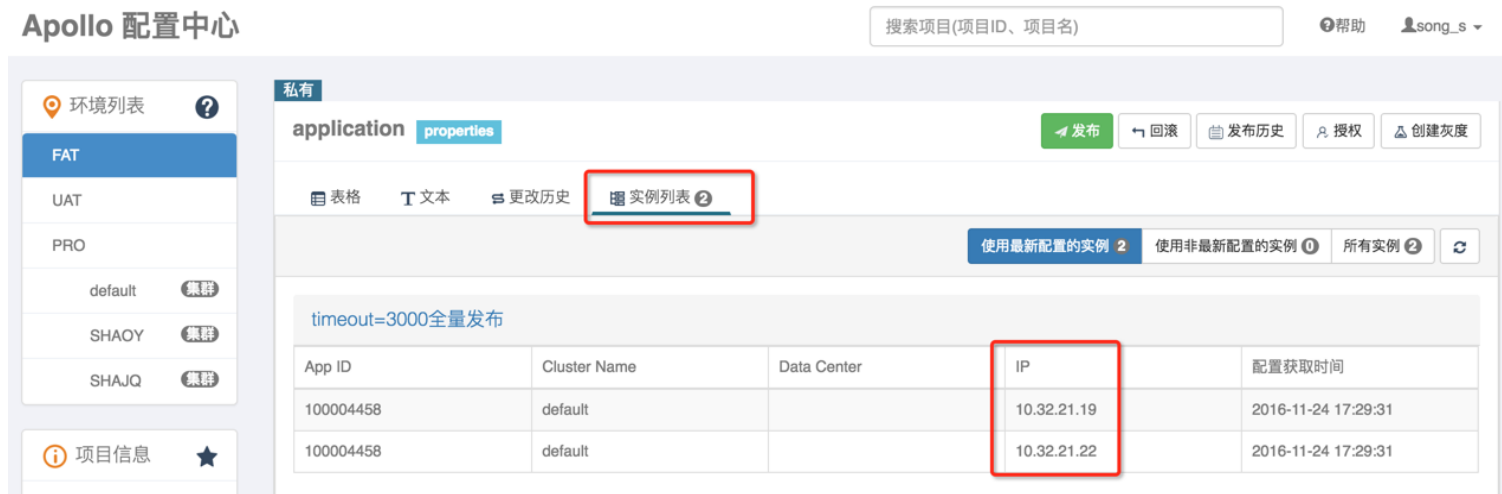
☒ 删除灰度版本

☐ 保留灰度版本

取消 确定



我选择了不保留灰度版本，所以发布完的效果就是主版本的配置更新、灰度版本删除。点击主版本的实例列表，可以看到10.32.21.22和10.32.21.19都使用了主版本最新的配置。



7.7 放弃灰度

如果灰度版本不理想或者不需要了，可以点击 **放弃灰度**。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助

song_s

环境列表

FAT

UAT

PRO

default 集群

SHAOY 集群

SHAJQ 集群

项目信息

AppId: 100004458

应用名: apollo-demo

部门: 框架(FX)

负责人: song_s

私有

主版本

灰度版本

application

properties

灰度发布

全量发布

放弃灰度

废弃灰度版本

配置

灰度规则

灰度实例列表 1

更改历史

灰度的配置

+ 新增灰度配置

发布状态	Key ↑↓	主版本的值	灰度的值	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout 改	2000	3000	Demo程序会用，勿删	song_s	2016-11-24 17:02:28	 

主版本的配置

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	kibana.url	http://1.1.1.2:5601		song_s	2016-10-18 19:57:29	

7.8 发布历史

点击主版本的 发布历史 按钮，可以看到当前namespace的主版本以及灰度版本的发布历史。

Apollo 配置中心

搜索项目(项目ID、项目名)

帮助

song_s

环境列表

FAT

UAT

PRO

default 集群

SHAOY 集群

SHAJQ 集群

私有

application

properties

发布

回滚

发布历史

授权

创建灰度

表格

文本

更改历史

实例列表 2

查看发布历史

同步配置

+ 新增配置

发布状态	Key ↑↓	Value	备注	最后修改人 ↑↓	最后修改时间 ↑↓	操作
已发布	timeout	3000	Demo程序会用，勿删	song_s	2016-11-24 17:29:31	 
已发布	kibana.url	http://1.1.1.2:5601		song_s	2016-10-18 19:57:29	 

发布历史

(AppId:100004458, ENV:FAT, Cluster:default, Namespace:application)

返回到项目首页

主版本发布 主版本回滚 灰度操作

song_s

删除灰度(全量发布)

5分钟前

song_s

灰度全量发布

5分钟前

song_s

灰度发布

18分钟前

song_s

更新灰度规则

24分钟前

song_s

更新灰度规则

25分钟前

timeout=3000全量发布

对10.32.21.22测试timeout=3000成功

2016-11-24 17:29:31

变更的配置

全部配置

变更的配置

Type	Key	Old Value	New Value
修改	timeout	2000	3000

8 其它功能配置

8.1 配置查看权限

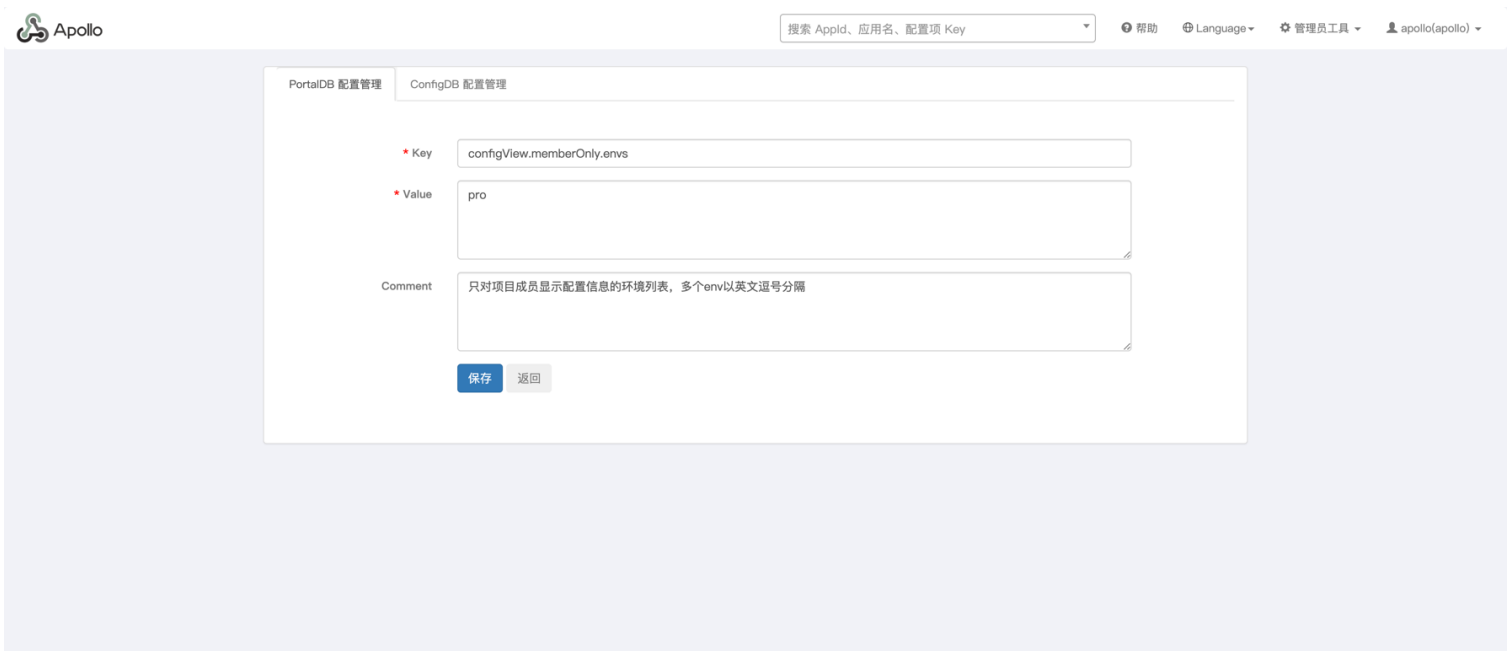
从1.1.0版本开始，apollo-portal增加了查看权限的支持，可以支持配置某个环境只允许项目成员查看私有Namespace的配置。

这里的项目成员是指：

1. 项目的管理员
2. 具备该私有Namespace在该环境下的修改或发布权限

配置方式很简单，用超级管理员账号登录后，进入 **管理员工具 - 系统参数** 页面新增或修改

`configView.memberOnly.envs` 配置项即可。



The screenshot shows the Apollo Portal interface. At the top, there's a search bar with the text '搜索 AppId、应用名、配置项 Key'. To the right of the search bar are links for '帮助', 'Language', '管理员工具', and a user profile 'apollo(apollo)'. Below the search bar, there are two tabs: 'PortalDB 配置管理' and 'ConfigDB 配置管理'. The 'ConfigDB 配置管理' tab is active. Inside this tab, there are three input fields: 'Key' with the value 'configView.memberOnly.envs', 'Value' with the value 'pro', and 'Comment' with the text '只对项目成员显示配置信息的环境列表，多个env以英文逗号分隔'. At the bottom of the form, there are two buttons: '保存' (Save) and '返回' (Return).

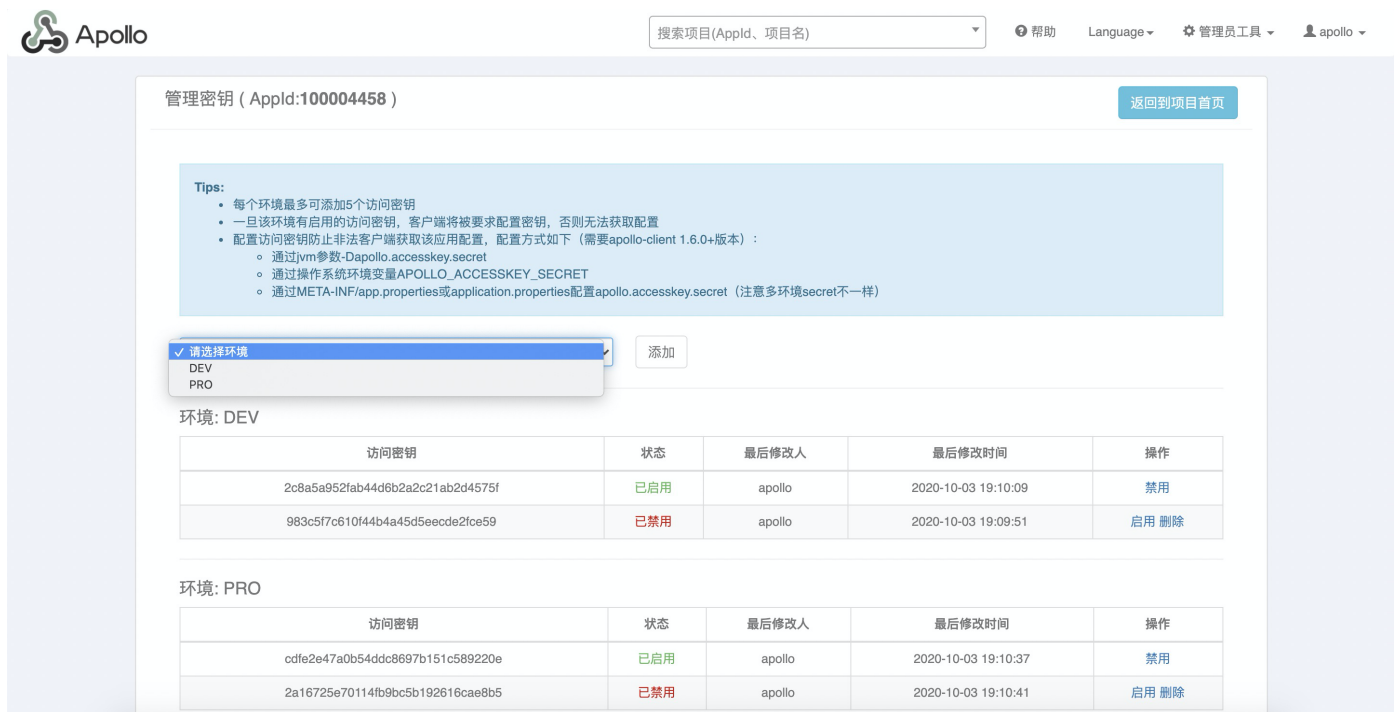
8.2 配置访问密钥

Apollo从1.6.0版本开始增加访问密钥机制，从而只有经过身份验证的客户端才能访问敏感配置。如果应用开启了访问密钥，客户端需要配置密钥，否则无法获取配置。

1. 项目管理员打开管理密钥页面



2. 为项目的每个环境生成访问密钥，注意默认是禁用的，建议在客户端都配置完成后再开启



3. 客户端侧配置访问密钥

9 最佳实践

9.1 安全相关

配置中心作为基础服务，存储着公司非常重要的配置信息，所以安全因素需要大家重点关注，下面列举了一些注意事项供大家参考，也欢迎大家分享自己的实践案例。

9.1.1 认证

建议接入公司统一的身份认证系统，如 SSO、LDAP 等，接入方式可以参考[Portal 实现用户登录功能](#)

如果使用Apollo提供的Spring Security简单认证，务必记得要修改超级管理员apollo的密码

9.1.2 授权

Apollo 支持细粒度的权限控制，请务必根据实际情况做好权限控制：

1. [项目管理员权限](#)

- Apollo 默认允许所有登录用户创建项目，如果只允许部分用户创建项目，可以开启[创建项目权限控制](#)

2. [配置编辑、发布权限](#)

- 配置编辑、发布权限支持按环境配置，比如开发环境开发人员可以自行完成配置编辑和发布的过程，但是生产环境发布权限交由测试或运维人员
- 生产环境建议同时开启[发布审核](#)，从而控制一次配置发布只能由一个人修改，另一个人发布，确保配置修改得到充分检查

3. [配置查看权限](#)

- 可以指定某个环境只允许项目成员查看私有Namespace的配置，从而避免敏感配置泄露，如生产环境

9.1.3 系统访问

除了用户权限，在系统访问上也需要加以考虑：

1. `apollo-configservice` 和 `apollo-adminservice` 是基于内网可信网络设计的，所以出于安全考虑，禁止 `apollo-configservice` 和 `apollo-adminservice` 直接暴露在公网
2. 对敏感配置可以考虑开启[访问秘钥](#)，从而只有经过身份验证的客户端才能访问敏感配置
3. 1.7.1及以上版本可以考虑为 `apollo-adminservice` 开启[访问控制](#)，从而只有[受控的](#) `apollo-portal` 才能访问对应接口，增强安全性
4. 2.1.0及以上版本可以考虑为 `eureka` 开启[访问控制](#)，从而只有受控的 `apollo-configservice` 和 `apollo-adminservice` 可以注册到 `eureka`，增强安全性

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。



云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网: www.apusic.com

