

Apollo

Apollo是由携程开源的分布式配置中心。

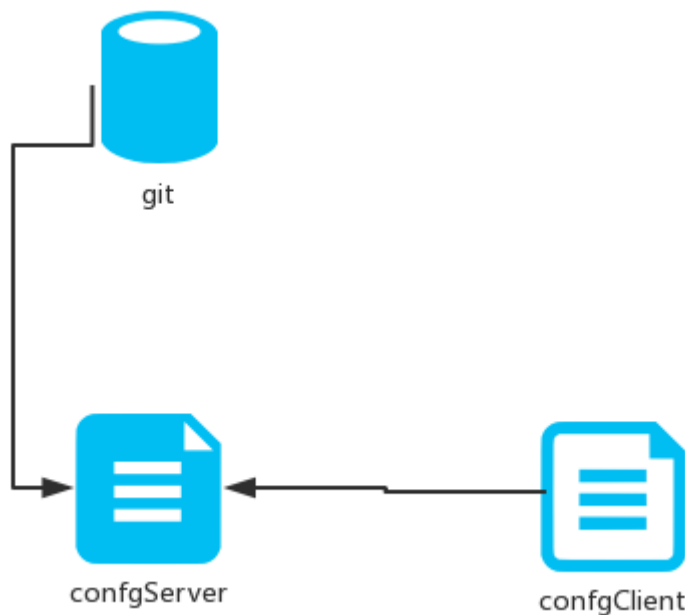
Apollo的特点有很多，比如：配置更新之后可以实时生效，还可以支持灰度发布功能。并且能对所有的配置进行版本管理、操作审计等功能，提供开放平台API。另外由于Apollo使用的人很多，所以网上的资料也非常的丰富，并且github上资料也写的很详细。

Apollo是由携程开源的分布式配置中心。

Apollo的特点有很多，比如：配置更新之后可以实时生效，还可以支持灰度发布功能。并且能对所有的配置进行版本管理、操作审计等功能，提供开放平台API。另外由于Apollo使用的人很多，所以网上的资料也非常的丰富，并且github上资料也写的很详细。

Spring Cloud Config

看名字就知道，这是Spring Cloud中带的配置中心组件。也正是这个原因，所以它和Spring是无缝集成，使用起来非常方便。并且它的配置存储支持Git，不过它没有可视化的操作界面，配置的生效也不是实时的，需要重启或去刷新。所以比较适用于小型项目快速上手。



Config Client和Config Server两部分，Config Server 实现配置文件的存储，对外以接口的形式提供获取配置文件，然后Config Client通过这些接口获取数据。

nacos

测试环境的地址：

<http://cdh1:8848/nacos/#/login>

Nacos 是阿里开源的，可以方便地集成 Spring Cloud 框架。nacos现在用的越来越多，以后也会是一个大的趋势，但是现在可能还没那么的普及

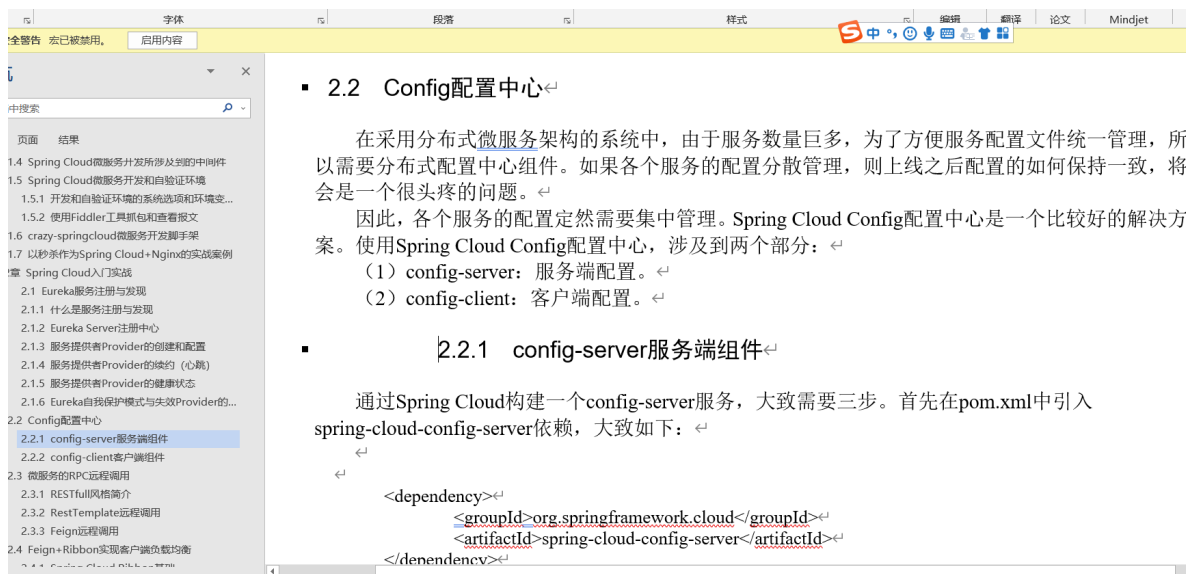
Nacos是基于raft算法的CP模型，同时也支持配置成类似eureka的AP。其实CP或者AP也都行，CP就是偶尔可能短时间不可用，AP就是可能数据不一致，两个都有问题，但是在生产环境下，无论CP还是AP其实都用的很多

除了服务注册和发现之外，Nacos 还提供了配置管理、元数据管理和流量管理等功能，并且提供了一个可视化的控制台管理界面。

具体，请参考：

<https://www.cnblogs.com/crazymakercircle/p/14231815.html>

Spring Cloud Config



The screenshot shows a document editor with a table of contents on the left and a main text area on the right. The table of contents includes sections like '2.2 Config配置中心' and '2.2.1 config-server服务端组件'. The main text area contains a paragraph about distributed configuration management and a code snippet for a Maven dependency.

2.2 Config配置中心

在采用分布式微服务架构的系统中，由于服务数量巨大，为了方便服务配置文件统一管理，所以需要分布式配置中心组件。如果各个服务的配置分散管理，则上线之后配置的如何保持一致，将会是一个很头疼的问题。

因此，各个服务的配置定然需要集中管理。Spring Cloud Config配置中心是一个比较好的解决方案。使用Spring Cloud Config配置中心，涉及到两个部分：

- (1) config-server: 服务端配置。
- (2) config-client: 客户端配置。

2.2.1 config-server服务端组件

通过Spring Cloud构建一个config-server服务，大致需要三步。首先在pom.xml中引入spring-cloud-config-server依赖，大致如下：

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-config-server</artifactId>
</dependency>
```

Spring Cloud Config的打包

fatjar (肥包)

生产环境，主要使用docker镜像的模式，推送到镜像仓库

Spring Cloud Config的部署

```
[root@cdh1 work]# tar -zxvf cloud-config-1.0-SNAPSHOT-Alpha.tar.gz
cloud-config-1.0-SNAPSHOT/bin/
cloud-config-1.0-SNAPSHOT/bin/deploy.bat
cloud-config-1.0-SNAPSHOT/bin/deploy.sh
```

```
cloud-config-1.0-SNAPSHOT/lib/cloud-config-1.0-SNAPSHOT.jar
cloud-config-1.0-SNAPSHOT/logs/
```

```
[root@cdh1 work]# sh /work/cloud-config-1.0-SNAPSHOT/bin/deploy.sh stop
PORT:7788
JVM:-server -Xms64m -Xmx256m
Shutting down boot: cloud-config-1.0-SNAPSHOT.jar is stopped
```

```
[root@cdh1 work]# sh /work/cloud-config-1.0-SNAPSHOT/bin/deploy.sh start
PORT:7788
JVM:-server -Xms64m -Xmx256m
cloud-config-1.0-SNAPSHOT.jar is running,pid is 22175
```

Spring Cloud Config的配置访问

测试环境的参考地址:

<http://cdh1:7788/crazymaker/redis/dev>

微服务如何使用配置信息