## 一站式微服务解决方案Spring Cloud Alibaba

主讲：Cat 老师

北京动力节点教育科技有限公司  
2020 • 北京

动力节点•版权所有•禁止传播

## 快速回顾

1、Spring家族开源项目梳理

2、Spring Cloud下的开源项目梳理

3、Spring Cloud Alibaba下的开源及商业项目梳理

4、微服务的基础模型：服务消费者-注册中心-服务提供者

5、What is Nacos？

6、Nacos的运行环境部署（Java写的，springboot项目）

7、Nacos的后台web管控台

8、Nacos作为注册中心注册服务

9、Nacos作为注册中心发现/订阅服务

10、服务消费者负载均衡调用服务提供者（ribbon）restTemple、feign

11、Nacos宕机时服务消费者缓存注册中心信息

12、Nacos作为配置中心存储项目各种配置

13、Nacos作为配置中心支持自动配置刷新（不需要重启应用）

14、Nacos配置中心DataId+Group+Properties/yaml+配置内容（比较灵活）

15、Nacos配置中心多环境配置（profile）

即${spring.application.name}-${profile}.${file-extension:properties}

## Nacos服务配置数据模型

这里面涉及到三个概念

1. 命名空间 （java package： com.bjpowernode.service / controller）

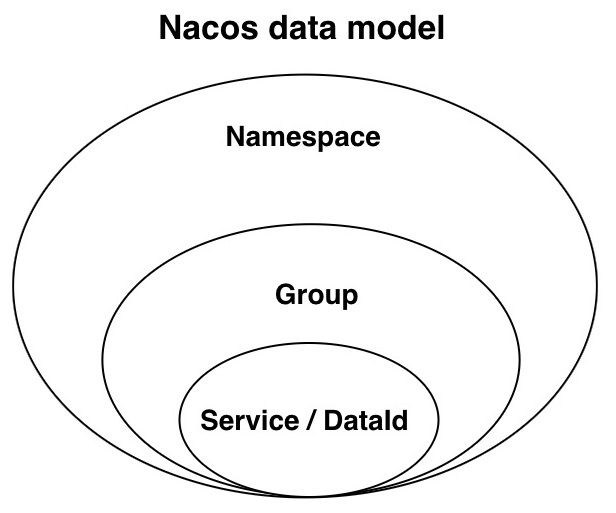
用于配置隔离，不同命名空间下，可以存在相同的Group或Data ID配置，Namespace的常用场景之一是不同环境的配置进行区分隔离，例如开发环境、测试环境和生产环境的资源（如配置、服务）隔离等；

1. Group

Nacos中的一组配置集合，是组织配置的维度之一，通过一个有意义的字符串（如 Buy 或 Trade）对一组配置集合进行分组，从而区分Data ID相同的配置集合，当在 Nacos 上创建一个配置时，如果未填写配置分组的名称，则配置分组的名称默认采用 DEFAULT\_GROUP，配置分组的常见场景：不同的应用或组件使用了相同的配置类型，如 database\_url配置和MQ\_topic配置；

1. Data Id

Nacos中的某个配置集合的ID，配置集合ID是组织划分配置的维度之一，Data ID通常用于组织划分系统的配置集合，一个系统或者应用可以包含多个配置集合，每个配置集都可以被一个有意义的名称标识；



第一种方案：

默认命名空间(public) + 默认Group分组(DEFAULT\_GROUP) + 自定义Data Id（没有默认值）；

默认命名空间：public

默认Group：DEFAULT\_GROUP

自定义Data Id：

${spring.application.name}-${profile}.${file-extension:properties}

比如：nacos-config-dev.yaml

第二种方案：

默认命名空间 + 自定义Group分组 + 自定义的Data Id（没有默认值）；

默认命名空间：public

默认Group：MY\_GROUP

自定义Data Id：

${spring.application.name}-${profile}.${file-extension:properties}

比如：nacos-config-dev.yaml

第三种方案：

自定义命名空间 + 自定义Group分组 + 自定义的Data Id（没有默认值）；

默认命名空间：public

默认Group：MY\_GROUP

自定义Data Id：

${spring.application.name}-${profile}.${file-extension:properties}

比如：nacos-config-dev.yaml

## Nacos数据持久化

Nacos默认情况下是采用apache derby内嵌数据库进行数据存储，在单机模式时可以使用nacos嵌入式数据库实现数据存储，但是derby数据库不方便观察数据存储的基本情况，从nacos 0.7版本开始增加了支持mysql数据源能力；

具体操作步骤：

第一步：安装数据库，版本要求：5.6.5+；

第二步：初始化MySQL数据库，数据库初始化文件： nacos-mysql.sql，该文件可以在Nacos程序包下的 conf目录下获得；

第三步：修改 conf/application.properties文件，增加支持MySQL数据源配置，添加（目前只支持mysql）数据源的url、用户名和密码；

然后启动nacos（还是单机启动），此时发现nacos所有写嵌入式数据库的数据都写到了mysql；

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Config Module Related Configurations \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

### If use MySQL as datasource:

spring.datasource.platform=mysql

### Count of DB:

db.num=1

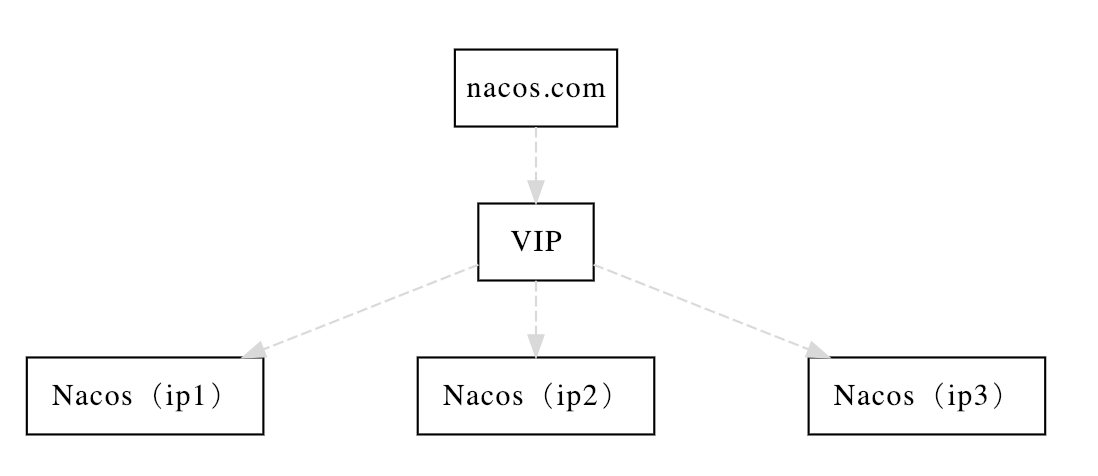
### Connect URL of DB:

db.url.0=jdbc:mysql://127.0.0.1:3306/nacos\_config?characterEncoding=utf8&connectTimeout=1000&socketTimeout=3000&autoReconnect=true&useUnicode=true&useSSL=false&serverTimezone=UTC

db.user=root

db.password=123456

## Ncaos集群部署



具体配置步骤：

1、准备三个nacos，3个或3个以上Nacos节点才能构成集群；

2、在Nacos的conf目录下有一个cluster.conf.example，可以直接把example扩展名去掉来使用，也可以单独创建一个cluster.conf文件，然后在该文件中每行配置一个ip:port；

192.168.172.128:8801

192.168.172.128:8802

192.168.172.128:8803

1. 按照上面nacos持久化的方式配置好数据持久化到MySQL，生产使用建议至少主备模式，或者采用高可用数据库；
2. 启动测试；

集群下客户端怎么连？

第一种：可以在单机的基础上加多个IP和端口中间用逗号隔开；

spring.cloud.nacos.discovery.server-addr=192.168.172.128:8801,192.168.172.128:8802,192.168.172.128:8803

第二种：配合Nginx代理我们的Nacos集群，配置里就直接写Nginx的IP和端口即可；

Nacos集群节点有三种角色状态：leader、follower、candidate；

当leader宕机，会从剩下的follower中投票选举出一个新的leader，选举算法是基于Raft算法实现；

经测试，发现有一点与三个角色不符，部署3个nacos节点，其中宕机2台，只剩下一个节点，此节点将变为candidate角色，但是此时该nacos集群仍然可以注册服务，订阅服务，（按照正确的理论应该是：如果nacos集群中没有leader角色的节点就不能注册服务，因为leader角色处理事务性请求），这比较匪夷所思，有待研究；