为什么要看源码：

1、**提升技术功底：**学习源码里的优秀设计思想，比如一些疑难问题的解决思路，还有一些优秀的设计模式，整体提升自己的技术功底

2、**深度掌握技术框架：**源码看多了，对于一个新技术或框架的掌握速度会有大幅提升，看下框架demo大致就能知道底层的实现，技术框架更新再快也不怕

3、**快速定位线上问题：**遇到线上问题，特别是框架源码里的问题(比如bug)，能够快速定位，这就是相比其他没看过源码的人的优势

4、**对面试大有裨益：**面试一线互联网公司对于框架技术一般都会问到源码级别的实现

5、**知其然知其所以然：**对技术有追求的人必做之事，使用了一个好的框架，很想知道底层是如何实现的

6、**拥抱开源社区：**参与到开源项目的研发，结识更多大牛，积累更多优质人脉

看源码方法：

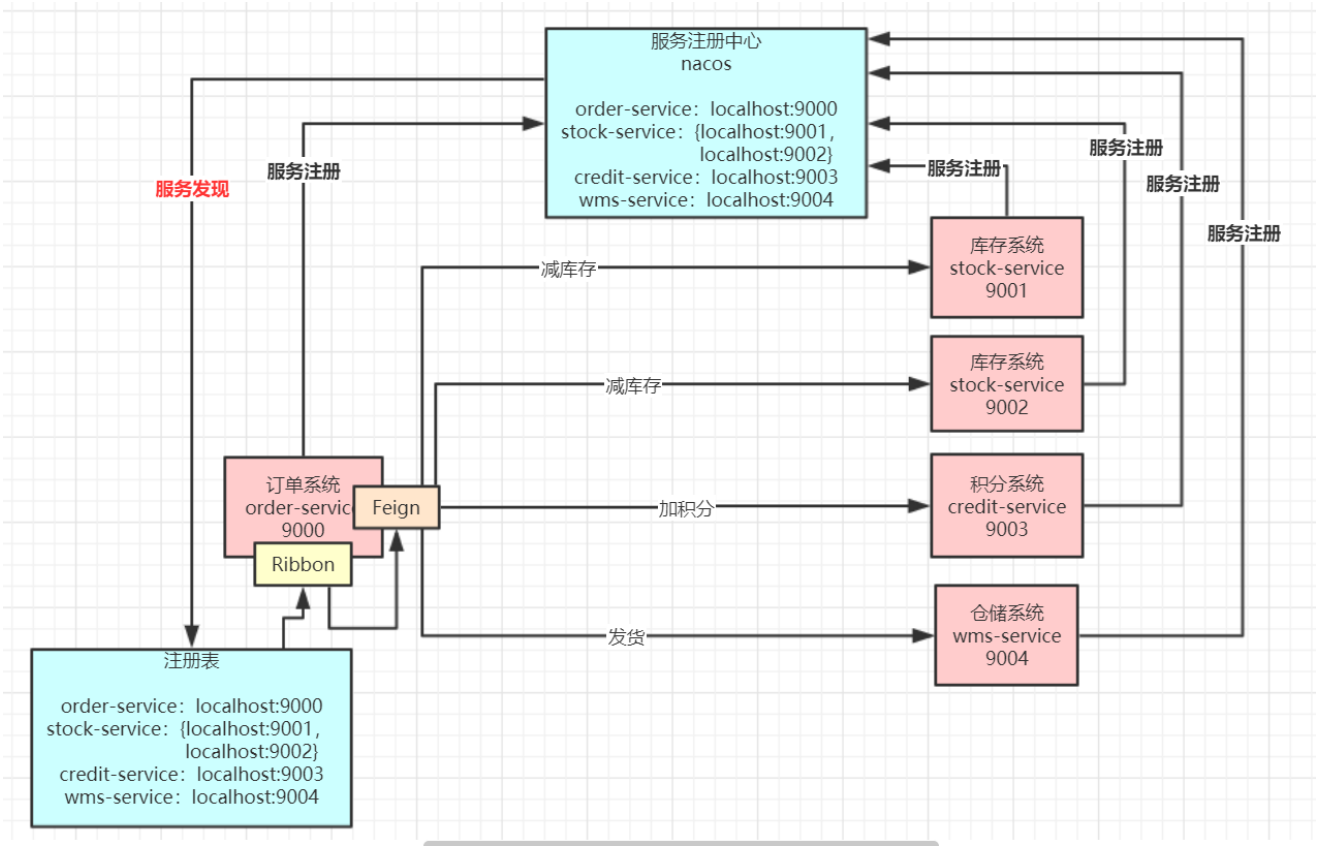
1、**先使用：**先看官方文档快速掌握框架的基本使用

2、**抓主线：**找一个demo入手，顺藤摸瓜快速静态看一遍框架的主线源码，画出源码主流程图，切勿一开始就陷入源码的细枝末节，否则会把自己绕晕，凭经验猜

3、**画图做笔记：**总结框架的一些核心功能点，从这些功能点入手深入到源码的细节，**边看源码边画源码走向图**，并对关键源码的理解做笔记，把源码里的闪光点都记录下来，后续借鉴到工作项目中，理解能力强的可以直接看静态源码，也可以边看源码边debug源码执行过程，观察一些关键变量的值

4、**整合总结：**所有功能点的源码都分析完后，回到主流程图再梳理一遍，争取把自己画的所有图都在脑袋里做一个整合

Nacos&Ribbon&Feign核心微服务架构图



架构原理

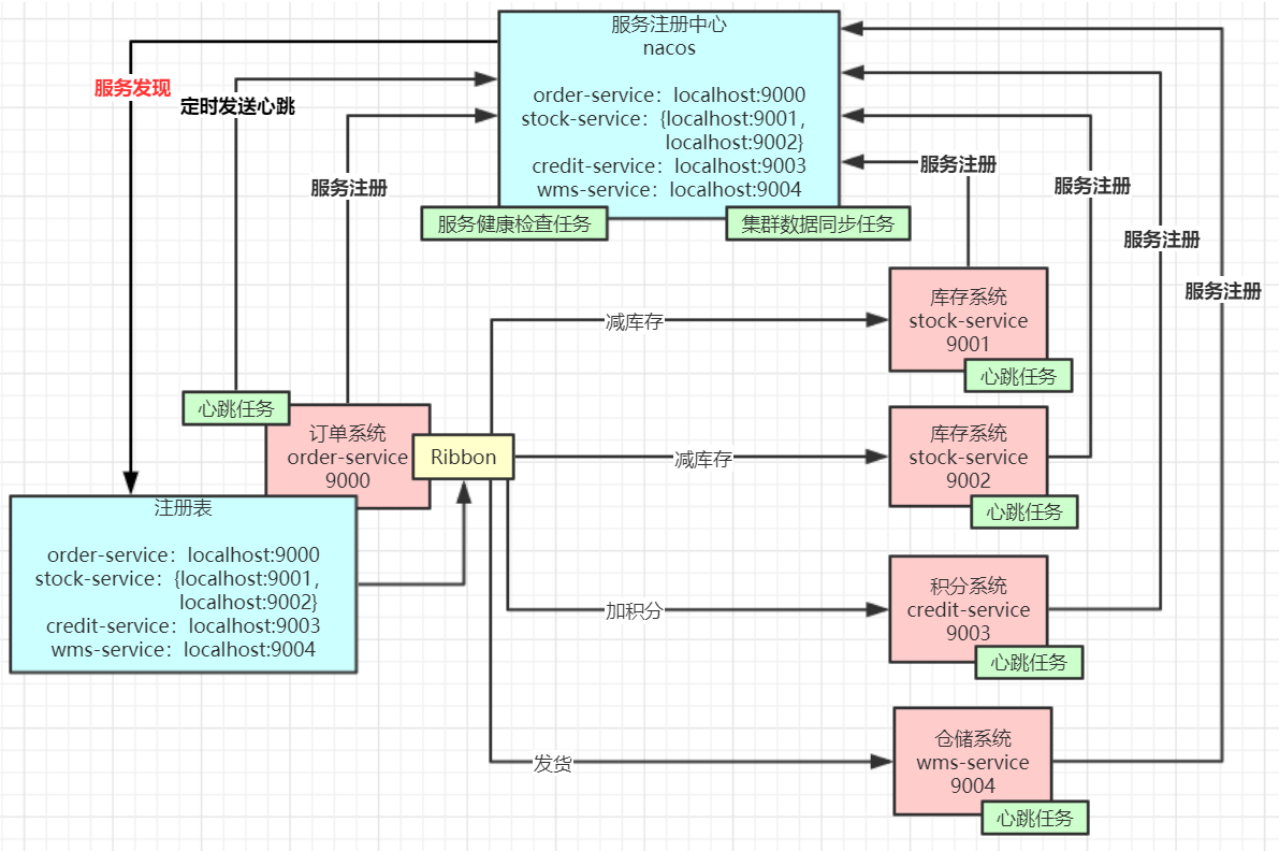
1、微服务系统在启动时将自己注册到服务注册中心，同时外发布 Http 接口供其它系统调用(一般都是基于Spring MVC)

2、服务消费者基于 Feign 调用服务提供者对外发布的接口，先对调用的本地接口加上注解@FeignClient，Feign会针对加了该注解的接口生成动态代理，服务消费者针对 Feign 生成的动态代理去调用方法时，会在底层生成Http协议格式的请求，类似 /stock/deduct?productId=100

3、Feign 最终会调用Ribbon从本地的Nacos注册表的缓存里根据服务名取出服务提供在机器的列表，然后进行负载均衡并选择一台机器出来，对选出来的机器IP和端口拼接之前生成的url请求，生成调用的Http接口地址

http://192.168.0.60:9000/stock/deduct?productId=100，最后基于HTTPClient调用请求

Nacos架构图



Nacos核心功能点

服务注册：Nacos Client会通过发送REST请求的方式向Nacos Server注册自己的服务，提供自身的元数据，比如ip地址、端口等信息。Nacos Server接收到注册请求后，就会把这些元数据信息存储在一个双层的内存Map中。

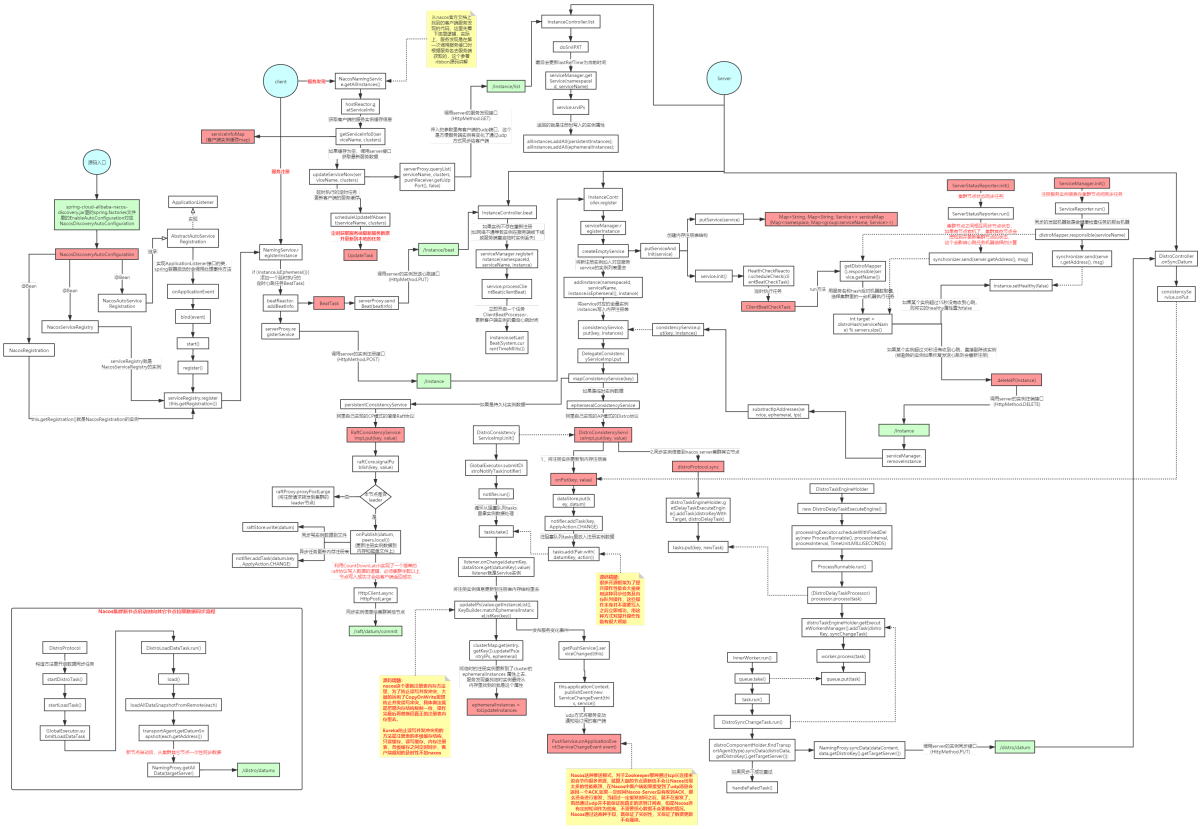
服务心跳：在服务注册后，Nacos Client会维护一个定时心跳来持续通知Nacos Server，说明服务一直处于可用状态，防止被剔除。默认5s发送一次心跳。

服务健康检查：Nacos Server会开启一个定时任务用来检查注册服务实例的健康情况，对于超过15s没有收到客户端心跳的实例会将它的healthy属性置为false(客户端服务发现时不会发现)，如果某个实例超过30秒没有收到心跳，直接剔除该实例(被剔除的实例如果恢复发送心跳则会重新注册)

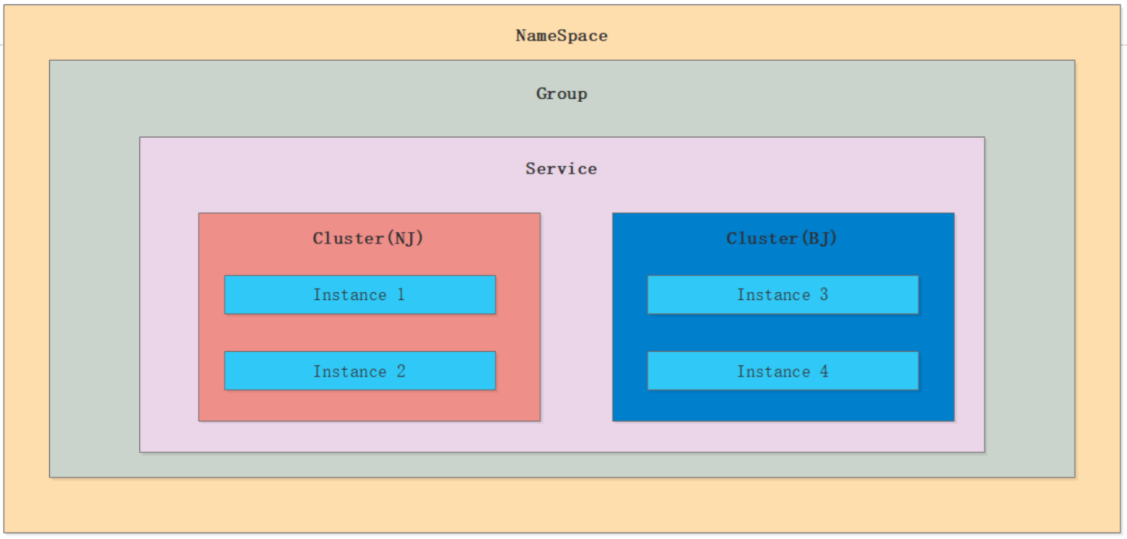
服务发现：服务消费者（Nacos Client）在调用服务提供者的服务时，会发送一个REST请求给Nacos Server，获取上面注册的服务清单，并且缓存在Nacos Client本地，同时会在Nacos Client本地开启一个定时任务定时拉取服务端最新的注册表信息更新到本地缓存

服务同步：Nacos Server集群之间会互相同步服务实例，用来保证服务信息的一致性。

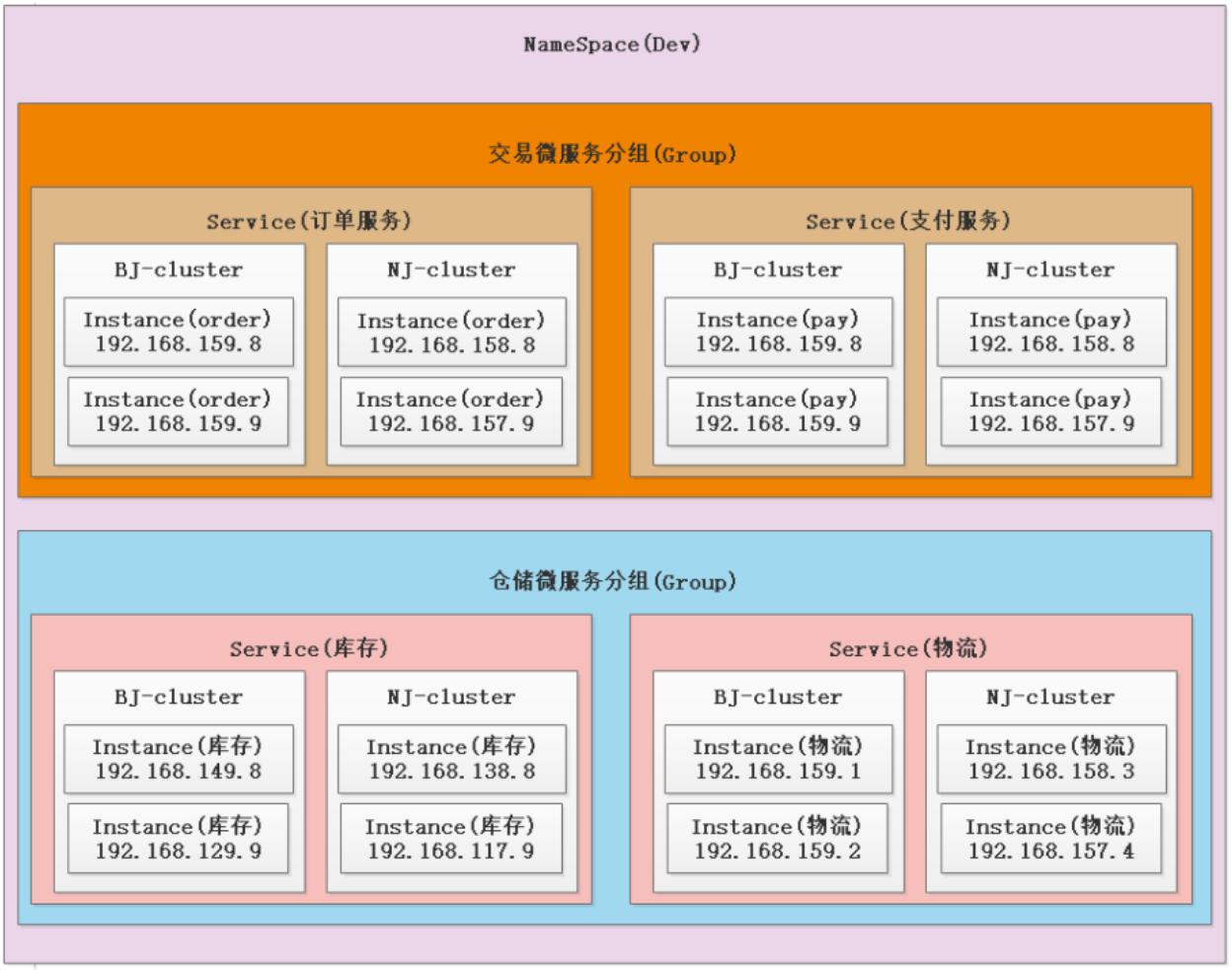
Nacos核心功能源码架构图



Nacos服务注册表结构：Map<namespace, Map<group::serviceName, Service>>



举例说明：

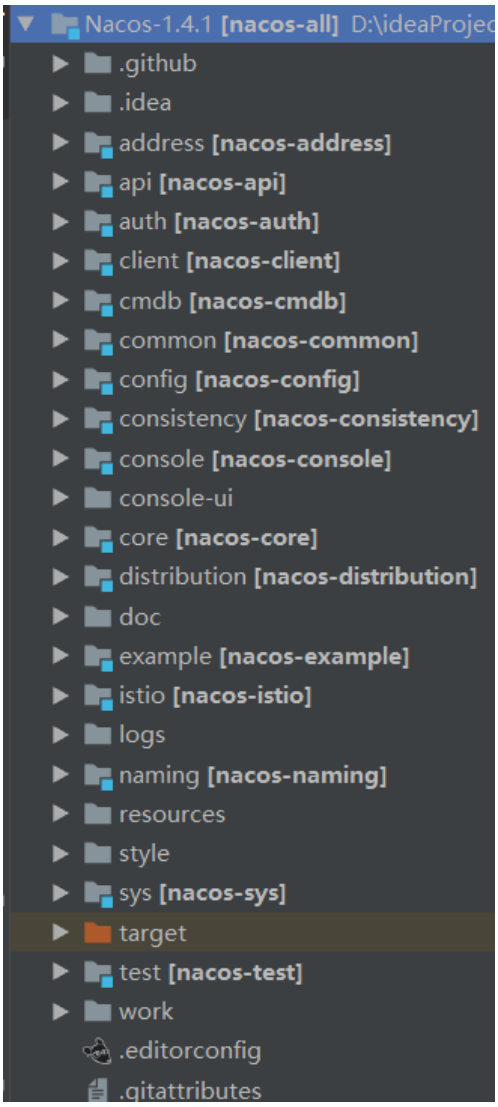


Nacos服务端源码单机运行

```
# 下载nacos源码
git clone https://github.com/alibaba/nacos.git
```

选择Tag 1.4.1版本

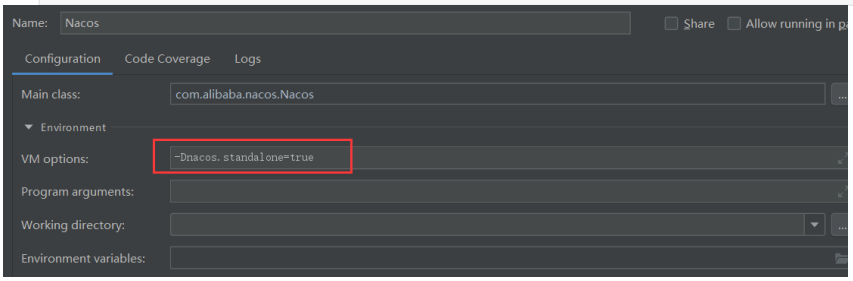
源码整体结构(注意，**nacos源码导入要求maven 3.2.5以上版本**)：



1、源码单机运行：

直接运行console模块里的 com.alibaba.nacos.Nacos.java

```
# 增加启动vm参数
-Dnacos.standalone=true
```



2、源码集群运行(启动流程参见视频)：

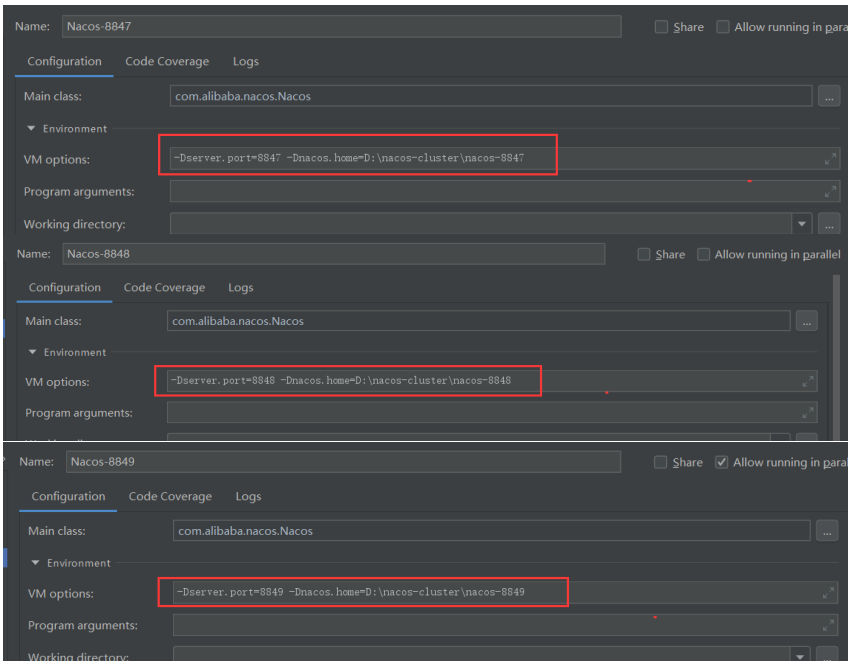
nacos集群需要配置mysql存储，需要先创建一个数据，名字随便取，然后执行 distribution/conf 目录下的 nacos-mysql.sql 脚本，然后修改 console\src\main\resources 目录下的 application.properties 文件里的mysql配置，如下所示

```
### 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?characterEncoding=utf8&connectTimeout=1000&socketTimeout=3000&autoReconnect=true&useUnicode=true&useSSL=false&serverTimezone=UTC
db.user.0=root
```

```
db.password.0=root
```

运行console模块里的 com.alibaba.nacos.Nacos.java，需要增加启动vm参数端口号和实例运行路径nacos.home(对应的目录需要自己提前创建好)，每台server的nacos.home目录里需要创建一个conf文件夹，里面放一个cluster.conf文件，文件里需要把所有集群机器ip和端口写入进去，见下图：

```
192.168.65.10:8847
192.168.65.10:8848
192.168.65.10:8849
```




```
文档：04-VIP-Nacos服务发现与注册源码剖析
链接：http://note.youdao.com/noteshare?id=17c68958637d60582e9c473f69f04aa5&sub=9C41C5161F864CC0BAA1DDE9B2CDCDE6
```