

主讲老师：Fox老师

## 1. 什么是 Nacos

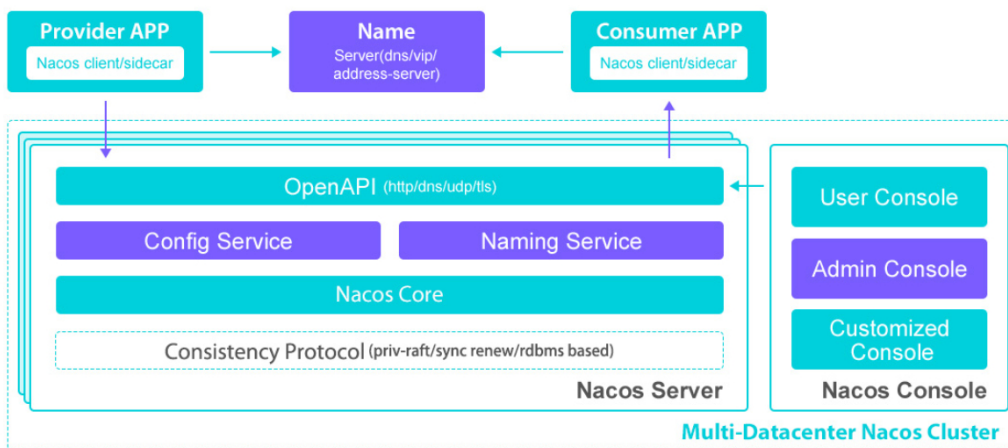
官方文档：<https://nacos.io/zh-cn/docs/what-is-nacos.html>

Nacos 致力于帮助您发现、配置和管理微服务。Nacos 提供了一组简单易用的特性集，帮助您快速实现动态服务发现、服务配置、服务元数据及流量管理。

Nacos 的关键特性包括：

- 服务发现和服务健康监测
- 动态配置服务
- 动态 DNS 服务
- 服务及其元数据管理

### 1.1 Nacos 架构



**NamingService:** 命名服务，注册中心核心接口

**ConfigService:** 配置服务，配置中心核心接口

OpenAPI文档：<https://nacos.io/zh-cn/docs/open-api.html>

nacos版本：v1.1.4 升级到v1.4.1

### 1.2 Nacos Server部署

#### 下载源码编译

源码下载地址：<https://github.com/alibaba/nacos/>

```
1 cd nacos/  
2 mvn -Prelease-nacos clean install -U  
3 cd nacos/distribution/target/
```

#### 下载安装包

下载地址：<https://github.com/alibaba/Nacos/releases>

#### 1.2.1 单机模式

官方文档：<https://nacos.io/zh-cn/docs/deployment.html>

解压，进入nacos目录

```
[root@redis nacos]# ls -l
总用量 32
drwxr-xr-x 4 root root    123 3月  2 21:07 bin
drwxr-xr-x 2 502 games   168 7月 30 14:53 conf
drwxr-xr-x 4 root root    38 3月  2 21:07 data
-rw-r--r-- 1 root root   722 4月 21 15:11 derby.log
-rw-r--r-- 1 502 games 17336 10月 11 2019 LICENSE
drwxr-xr-x 2 root root   4096 4月 21 15:54 logs
-rw-r--r-- 1 502 games  1305 10月 11 2019 NOTICE
drwxr-xr-x 2 root root    30 3月  2 21:01 target
drwxr-xr-x 3 root root    20 4月 21 15:11 work
```

单机启动nacos, 执行命令

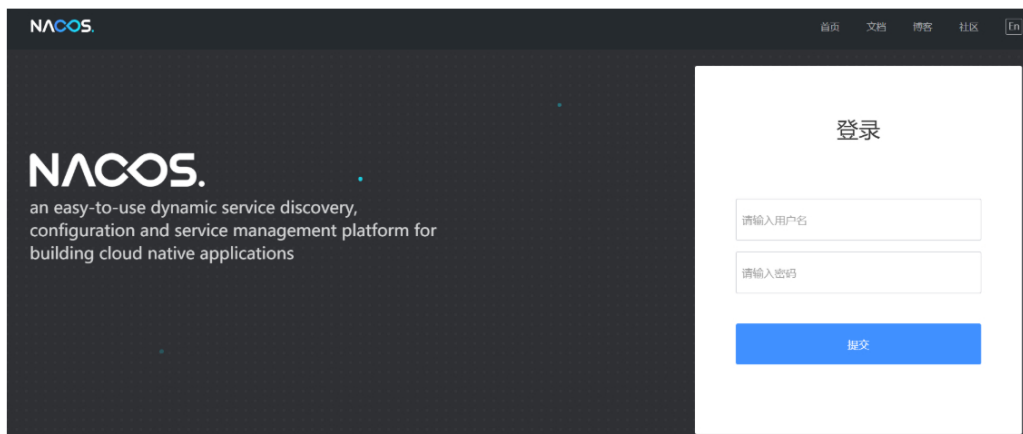
```
1 bin/startup.sh -m standalone
```

也可以修改默认启动方式

```
export SERVER="nacos-server"
export MODE="cluster"
export FUNCTION_MODE="all"
export MEMBER_LIST=""
export EMBEDDED_STORAGE=""
case $opt in
f)
```

可以改为standalone, 单机启动

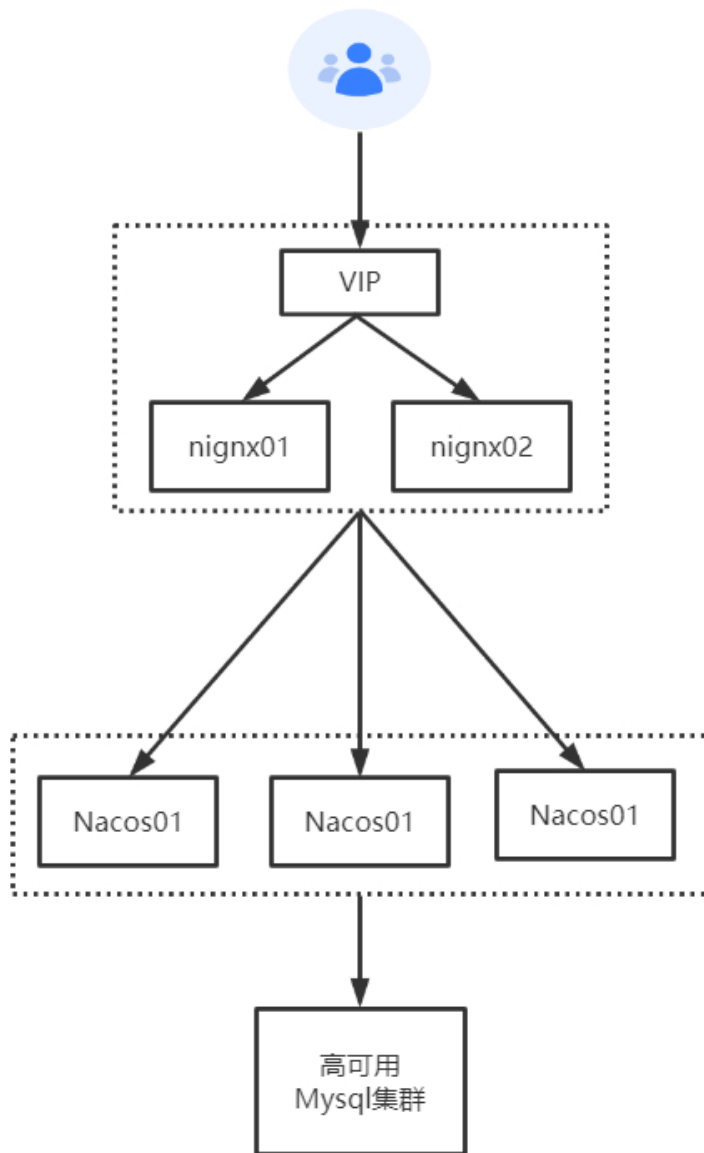
访问nacos的管理端: <http://192.168.3.14:8848/nacos>, 默认的用户名密码是 nocas/nocas



### 1.2.2 集群模式

官网文档: <https://nacos.io/zh-cn/docs/cluster-mode-quick-start.html>

集群部署架构图



1) 单机搭建伪集群，复制nacos安装包，修改为nacos8849, nacos8850, nacos8851

```
[root@redis nacos-cluster]# ls
nacos8849 nacos8850 nacos8851
```

2) 以nacos8849为例，进入nacos8849目录

2.1) 修改conf/application.properties的配置，使用外置数据源

```

1 #使用外置mysql数据源
2 spring.datasource.platform=mysql
3
4 ### Count of DB:
5 db.num=1
6
7 ### Connect URL of DB:
8 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
9 db.user.0=root
  
```

```
10 db.password.0=root
11
```

```
### 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
```

## 2.2) 将conf\cluster.conf.example改为cluster.conf,添加节点配置

```
1 # ip:port
2 192.168.3.14:8849
3 192.168.3.14:8850
4 192.168.3.14:8851
```

nacos8850, nacos8851 按同样的方式配置。

3) 创建mysql数据库,sql文件位置: conf\nacos-mysql.sql

4) 修改启动脚本 (bin\startup.sh) 的jvm参数

```
# JVM Configuration
#=====
=====
if [[ "${MODE}" == "standalone" ]]; then
  JAVA_OPTS="${JAVA_OPTS} -Xms512m -Xmx512m -Xmn256m"
  JAVA_OPTS="${JAVA_OPTS} -Dnacos.standalone=true"
else
```

5) 分别启动nacos8849, nacos8850, nacos8851

以nacos8849为例, 进入nacos8849目录, 启动nacos

```
1 bin/startup.sh
```

```
[root@redis nacos8850]# bin/startup.sh
/usr/local/jdk1.8.0_181/bin/java -server -Xms512m -Xmx512m -Xmn256m -XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m -XX:-OmitStackTraceInFastThrow -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/usr/local/soft/nacos-cluster/nacos8850/logs/java_heapdump.hprof -XX:-UseLargePages -Djava.ext.dirs=/usr/local/jdk1.8.0_181/jre/lib/ext:/usr/local/jdk1.8.0_181/lib/ext:/usr/local/soft/nacos-cluster/nacos8850/plugins/cmdb:/usr/local/soft/nacos-cluster/nacos8850/plugins/mysql -Xloggc:/usr/local/soft/nacos-cluster/nacos8850/logs/nacos_gc.log -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M -Dnacos.home=/usr/local/soft/nacos-cluster/nacos8850 -Dloader.path=/usr/local/soft/nacos-cluster/nacos8850/plugins/health -jar /usr/local/soft/nacos-cluster/nacos8850/target/nacos-server.jar --spring.config.location=classpath:/,classpath:/config/,file:./,file:./config/,file:/usr/local/soft/nacos-cluster/nacos8850/conf/ --logging.config=/usr/local/soft/nacos-cluster/nacos8850/conf/nacos-logback.xml --server.max-http-header-size=524288
nacos is starting with cluster
nacos is starting, you can check the /usr/local/soft/nacos-cluster/nacos8850/logs/start.out
```

6) 测试

登录 <http://192.168.3.14:8849/nacos> , 用户名和密码都是nacos

NACOS 1.1.4 public

节点列表 | public

节点ip  查询

节点ip	节点状态	集群任期	Leader任期(ms)	心跳间隔(ms)
192.168.3.14-8850	FOLLOWER	18	15867	2500
192.168.3.14-8851	FOLLOWER	0	17359	3000
192.168.3.14-8849	LEADER	18	17382	2500

## 7) 官方推荐, nginx反向代理

```

upstream nacoscluster {
    server 127.0.0.1:8851;
    server 127.0.0.1:8849;
    server 127.0.0.1:8850;
}

server {
    listen 8847;
    server_name localhost ;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location /nacos/ {
        proxy_pass http://nacoscluster/nacos/;
    }

```

访问: <http://192.168.3.14:8847/nacos>

## 1.3 prometheus+grafana监控Nacos

<https://nacos.io/zh-cn/docs/monitor-guide.html>

Nacos 0.8.0版本完善了监控系统, 支持通过暴露metrics数据接入第三方监控系统监控Nacos运行状态。

### 1. nacos暴露metrics数据

```
1 management.endpoints.web.exposure.include=*
```

测试: <http://localhost:8848/nacos/actuator/prometheus>

```
# HELP tomcat_global_error_total
# TYPE tomcat_global_error_total counter
tomcat_global_error_total{name="http-nio-8848",} 24.0
# HELP system_cpu_count The number of processors available to the Java virtual machine
# TYPE system_cpu_count gauge
system_cpu_count 12.0
# HELP process_start_time_seconds Start time of the process since unix epoch.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.615290084082E9
# HELP nacos_monitor
# TYPE nacos_monitor gauge
nacos_monitor{module="config",name="longPolling",} 0.0
nacos_monitor{module="config",name="configCount",} 32.0
nacos_monitor{module="naming",name="failedPush",} 0.0
nacos_monitor{module="naming",name="leaderStatus",} 0.0
nacos_monitor{module="config",name="publish",} 0.0
nacos_monitor{module="naming",name="tcpHealthCheck",} 0.0
nacos_monitor{module="config",name="dumpTask",} 0.0
nacos_monitor{module="config",name="notifyTask",} 0.0
nacos_monitor{module="naming",name="totalPush",} 7.0
nacos_monitor{module="naming",name="avgPushCost",} -1.0
nacos_monitor{module="config",name="getConfig",} 6.0
nacos_monitor{module="naming",name="ipCount",} 2.0
nacos_monitor{module="naming",name="mysqlhealthCheck",} 0.0
nacos_monitor{module="naming",name="serviceCount",} 2.0
nacos_monitor{module="naming",name="httpHealthCheck",} 0.0
nacos_monitor{module="naming",name="maxPushCost",} -1.0
# HELP tomcat_global_sent_bytes_total
# TYPE tomcat_global_sent_bytes_total counter
tomcat_global_sent_bytes_total{name="http-nio-8848",} 3.4530029E7
```

## 2. prometheus采集Nacos metrics数据

### 启动prometheus服务

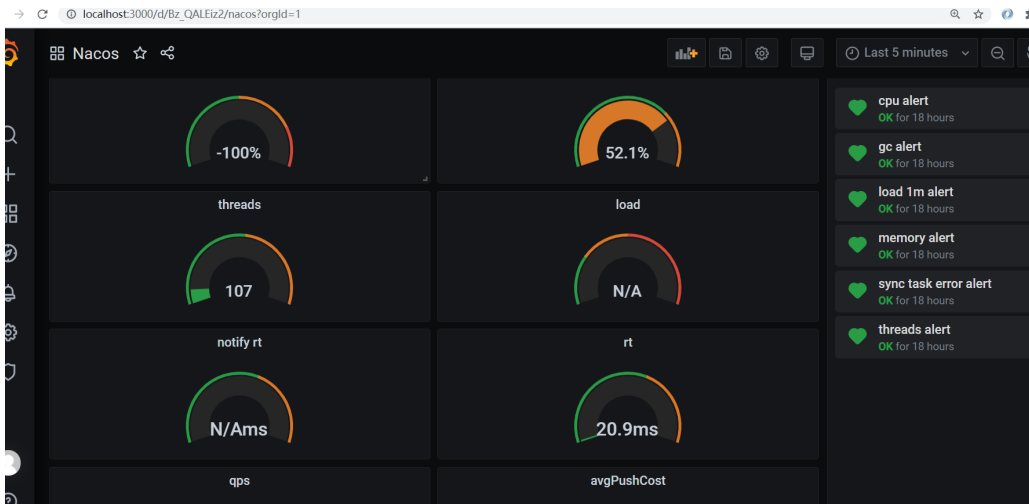
```
1 prometheus.exe --config.file=prometheus.yml
```

测试: <http://localhost:9090/graph>

The screenshot shows the Prometheus web interface. At the top, there are navigation links: Prometheus, Alerts, Graph, Status, Help, and Classic UI. Below these are three checkboxes: "Enable query history" (unchecked), "Use local time" (unchecked), and "Enable autocomplete" (checked). A search bar contains the text "nacos\_monitor". Below the search bar, there are two tabs: "Table" and "Graph", with "Table" selected. A "Table" header shows "Evaluation time" with left and right navigation arrows. The table contains 11 rows of metrics, each with a label like "nacos\_monitor{instance='localhost:8848', job='prometheus', module='config', name='configCount'}".

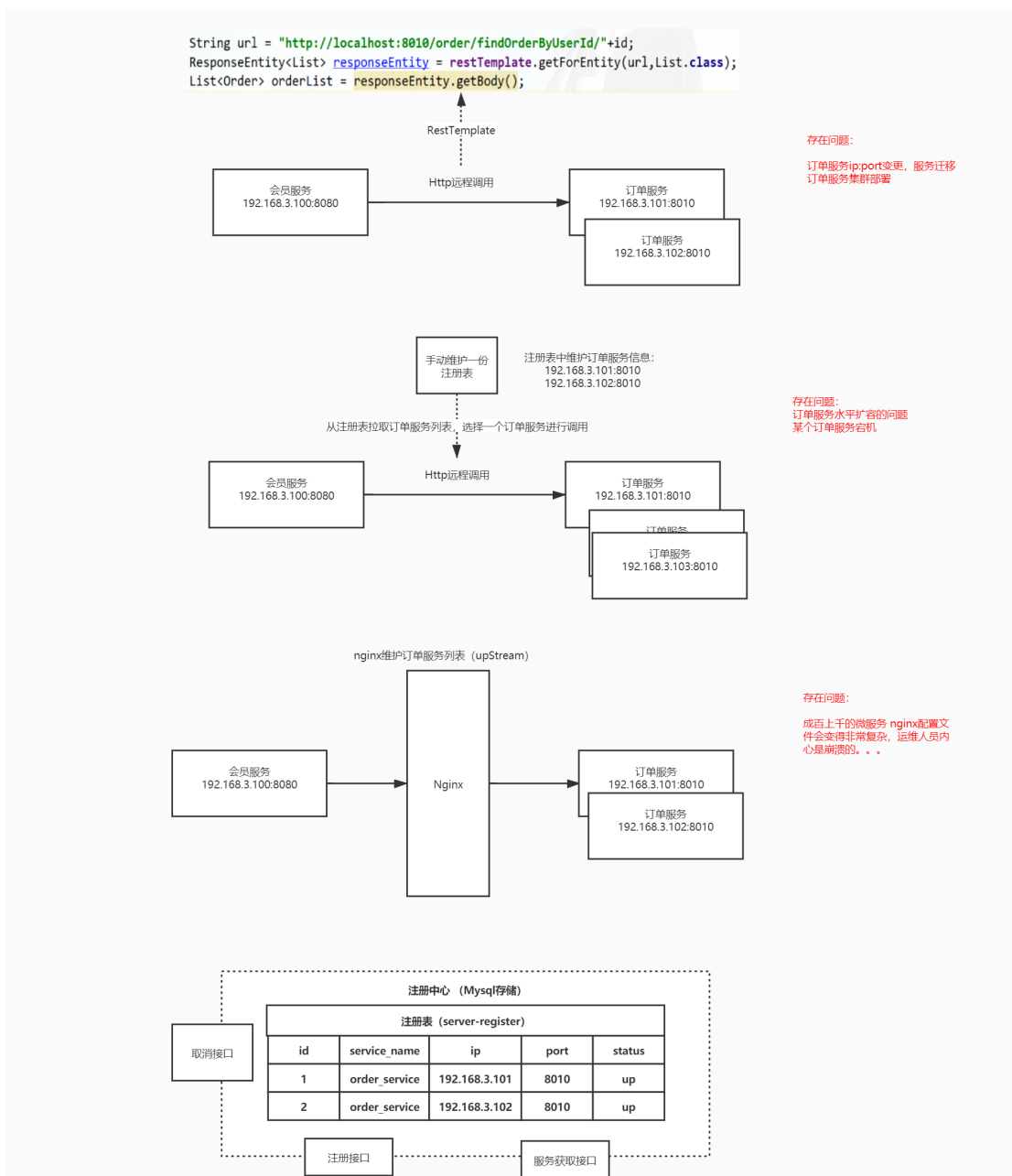
## 3. grafana展示metrics数据

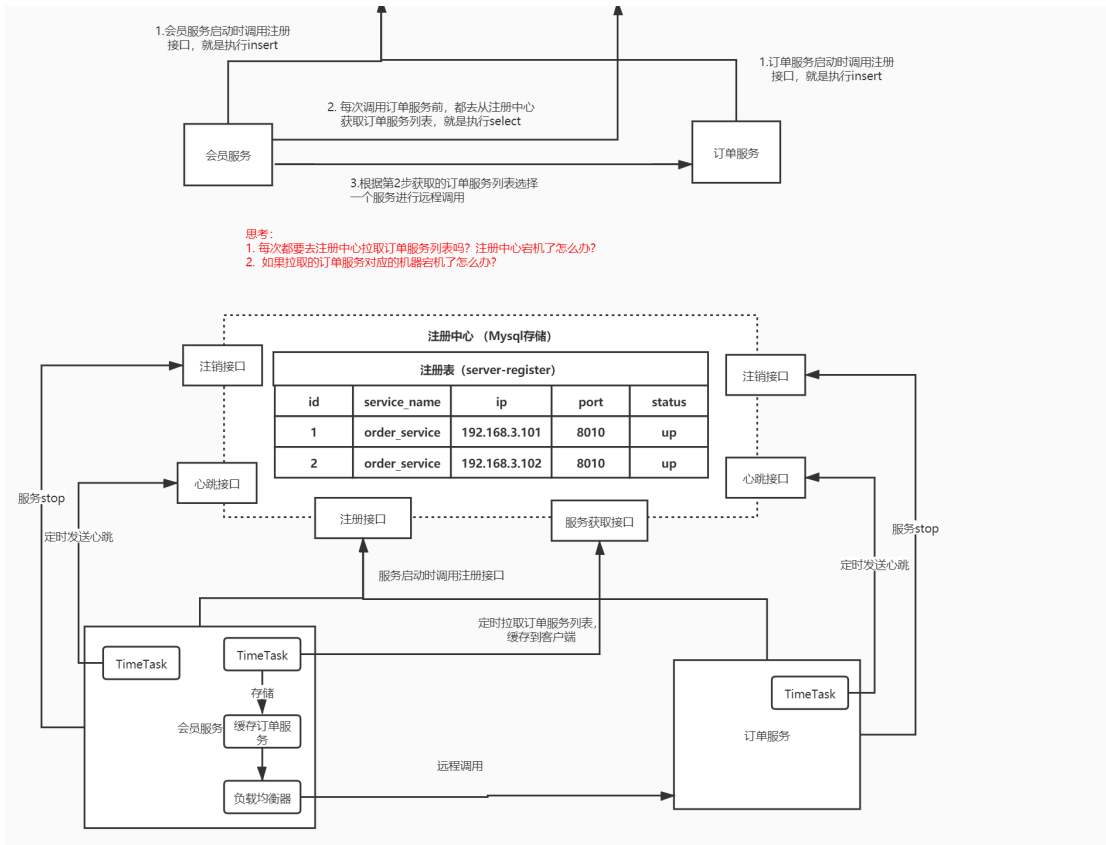
测试: <http://localhost:3000/>



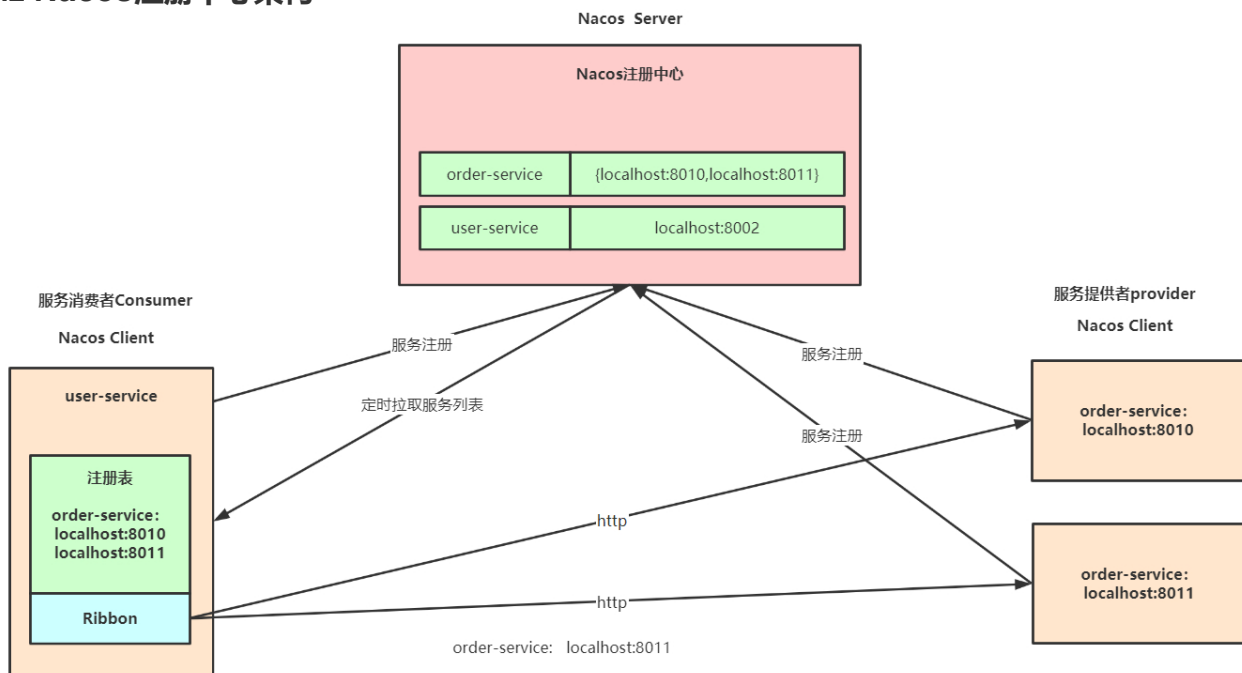
## 2. Nacos注册中心

### 2.1 注册中心演变及其设计思想





## 2.2 Nacos注册中心架构



## 2.3 核心功能

**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。

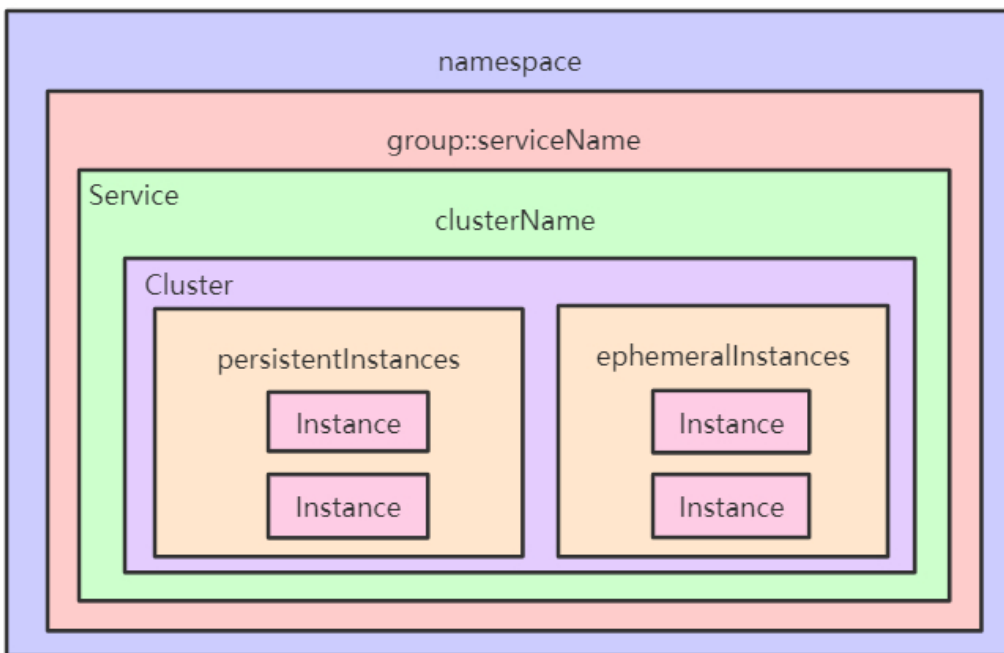
**服务心跳:** 在服务注册后, Nacos Client会维护一个定时心跳来持续通知Nacos Server, 说明服务一直处于可用状态, 防止被剔除。默认5s发送一次心跳。

**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。

**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)

## 2.4 服务注册表结构



## 2.4 服务领域模型



## 2.6 服务实例数据

```

{
  - hosts: [
    - {
      ip: "192.168.65.18",
      port: 8020,
      valid: true,
      healthy: true,
      marked: false,
      instanceId: "192.168.65.18#8020#DEFAULT#DEFAULT_GROUP@@mall-order",
      - metadata: {
          preserved.register.source: "SPRING_CLOUD"
        },
      enabled: true,
      weight: 1,
      clusterName: "DEFAULT",
      serviceName: "mall-order",
      ephemeral: true
    },
    + {...]
  ],
  dom: "mall-order",
  name: "DEFAULT_GROUP@@mall-order",
  cacheMillis: 3000,
  lastRefTime: 1616423527818,
  checksum: "b252f6aea678ed326f94fa8174121225",
  useSpecifiedURL: false,
  clusters: "",
  env: "",
  metadata: { }
}

```

### 3. Spring Cloud Alibaba Nacos快速开始

#### 3.1 Spring Cloud Alibaba版本选型

##### 组件版本关系

Spring Cloud Alibaba Version	Sentinel Version	Nacos Version	RocketMQ Version	Dubbo Version	Seata Version
2.2.4.RELEASE	1.8.0	1.4.1	4.4.0	2.7.8	1.3.0
2.2.3.RELEASE or 2.1.3.RELEASE or 2.0.3.RELEASE	1.8.0	1.3.3	4.4.0	2.7.8	1.3.0
2.2.1.RELEASE or 2.1.2.RELEASE or 2.0.2.RELEASE	1.7.1	1.2.1	4.4.0	2.7.6	1.2.0
2.2.0.RELEASE	1.7.1	1.1.4	4.4.0	2.7.4.1	1.0.0
2.1.1.RELEASE or 2.0.1.RELEASE or 1.5.1.RELEASE	1.7.0	1.1.4	4.4.0	2.7.3	0.9.0
2.1.0.RELEASE or 2.0.0.RELEASE or 1.5.0.RELEASE	1.6.3	1.1.1	4.4.0	2.7.3	0.7.1

## 毕业版本依赖关系(推荐使用)

Spring Cloud Version	Spring Cloud Alibaba Version	Spring Boot Version
Spring Cloud Hoxton.SR8	2.2.4.RELEASE	2.3.2.RELEASE
Spring Cloud Greenwich.SR6	2.1.3.RELEASE	2.1.13.RELEASE
Spring Cloud Hoxton.SR3	2.2.1.RELEASE	2.2.5.RELEASE
Spring Cloud Hoxton.RELEASE	2.2.0.RELEASE	2.2.X.RELEASE
Spring Cloud Greenwich	2.1.2.RELEASE	2.1.X.RELEASE
Spring Cloud Finchley	2.0.3.RELEASE	2.0.X.RELEASE
Spring Cloud Edgware	1.5.1.RELEASE(停止维护, 建议升级)	1.5.X.RELEASE

### 3.2 搭建Nacos-client服务

#### 1) 引入依赖

父Pom中支持spring cloud&spring cloud alibaba, 引入依赖

```
1 <dependencyManagement>
2 <dependencies>
3 <!--引入springcloud的版本-->
4 <dependency>
5 <groupId>org.springframework.cloud</groupId>
6 <artifactId>spring-cloud-dependencies</artifactId>
7 <version>Hoxton.SR3</version>
8 <type>pom</type>
9 <scope>import</scope>
10 </dependency>
11
12 <dependency>
13 <groupId>com.alibaba.cloud</groupId>
14 <artifactId>spring-cloud-alibaba-dependencies</artifactId>
15 <version>2.2.1.RELEASE</version>
16 <type>pom</type>
17 <scope>import</scope>
18 </dependency>
19 </dependencies>
20
21 </dependencyManagement>
```

当前项目pom中引入依赖

```
1 <dependency>
2 <groupId>com.alibaba.cloud</groupId>
3 <artifactId>spring-cloud-starter-alibaba-nacos-discovery</artifactId>
4 </dependency>
```

#### 2) application.properties中配置

```

1 server.port=8002
2 #微服务名称
3 spring.application.name=service-user
4 #配置 Nacos server 的地址
5 spring.cloud.nacos.discovery.server-addr=localhost:8848

```

更多配置：<https://github.com/alibaba/spring-cloud-alibaba/wiki/Nacos-discovery>

配置项	Key	默认值	说明
服务端地址	<code>spring.cloud.nacos.discovery.server-addr</code>	无	Nacos Server 启动监听的ip地址和端口
服务名	<code>spring.cloud.nacos.discovery.service</code>	<code>\${spring.application.name}</code>	给当前的服务命名
服务分组	<code>spring.cloud.nacos.discovery.group</code>	DEFAULT_GROUP	设置服务所处的分组
权重	<code>spring.cloud.nacos.discovery.weight</code>	1	取值范围 1 到 100，数值越大，权重越大
网卡名	<code>spring.cloud.nacos.discovery.network-interface</code>	无	当IP未配置时，注册的IP为此网卡所对应的IP地址，如果此项也未配置，则默认取第一块网卡的地址
注册的IP地址	<code>spring.cloud.nacos.discovery.ip</code>	无	优先级最高
注册的端口	<code>spring.cloud.nacos.discovery.port</code>	-1	默认情况下不用配置，会自动探测
命名空间	<code>spring.cloud.nacos.discovery.namespace</code>	无	常用场景之一是不同的环境的注册的区分隔离，例如开发测试环境和生产环境的资源（如配置、服务）隔离等。
AccessKey	<code>spring.cloud.nacos.discovery.access-key</code>	无	当要上阿里云时，阿里云上面的一个云账号名
SecretKey	<code>spring.cloud.nacos.discovery.secret-key</code>	无	当要上阿里云时，阿里云上面的一个云账号密码
Metadata	<code>spring.cloud.nacos.discovery.metadata</code>	无	使用Map格式配置，用户可以根据自己的需要自定义一些和服务相关的元数据信息
日志文件名	<code>spring.cloud.nacos.discovery.log-name</code>	无	
集群	<code>spring.cloud.nacos.discovery.cluster-name</code>	DEFAULT	配置成Nacos集群名称
接入点	<code>spring.cloud.nacos.discovery.endpoint</code>	UTF-8	地域的某个服务的入口域名，通过此域名可以动态地拿到服务端地址
是否集成			默认配置集成，即可

Ribbon	<code>ribbon.nacos.enabled</code>	true	一般默认是true即可
是否开启 Nacos Watch	<code>spring.cloud.nacos.discovery.watch.enabled</code>	true	可以设置成false来关闭 watch

© 2020 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

[Help](#)



[Contact GitHub](#)

[Pricing](#)

### 3) 启动springboot应用, nacos管理端界面查看是否成功注册

历史版本	服务名	分组名称	集群数目	实例数	健康实例数	触发保护阈值	操作
监听查询 服务管理 服务列表 订阅者列表	service-order	DEFAULT_GROUP	1	1	1	false	<a href="#">详情</a>   <a href="#">示例代码</a>   <a href="#">删除</a>

### 4) 测试

使用RestTemplate进行服务调用, 可以使用微服务名称 (spring.application.name)

```
1 String url = "http://service-order/order/findOrderByUserId/"+id;
2 List<Order> orderList = restTemplate.getForObject(url, List.class);
```

注意: 需要添加@LoadBalanced注解

```
1 @Bean
2 @LoadBalanced
3 public RestTemplate restTemplate() {
4     return new RestTemplate();
5 }
```

## 4. Nacos源码编译

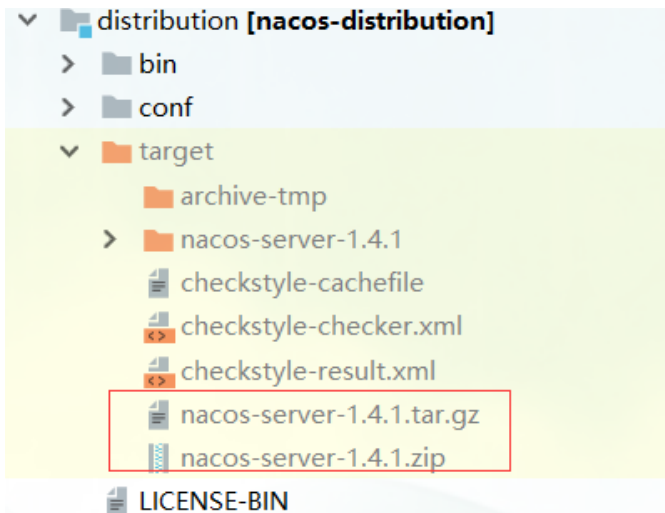
源码下载地址: <https://github.com/alibaba/nacos/>

版本: Nacos 1.4.1

### 1) 进入nacos目录, 执行编译命令

```
1 # 编译1.4.1 增加 -Drat.skip=true 参数, 跳过licensing 检查
2 mvn -Prelease-nacos -Dmaven.test.skip=true -Drat.skip=true clean install -U
```

编译成功后会在distribution/target目录下生成nacos-server-1.4.1.tar.gz



## 2) 启动nacos

进入console模块，找到启动类 `com.alibaba.nacos.Nacos`，执行main方法

```
@SpringBootApplication(scanBasePackages = "com.alibaba.nacos")
@EnableScheduling
public class Nacos {

    public static void main(String[] args) {
        SpringApplication.run(Nacos.class, args);
    }
}
```

## 3) 配置启动参数

单机模式执行需要指定`nacos.standalone=true`

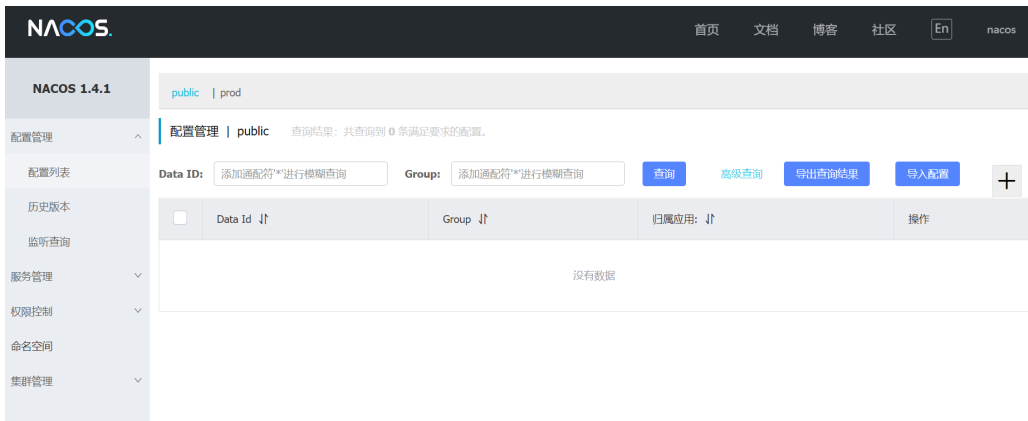
```
1 -Dnacos.standalone=true -Dnacos.home=D:\code\java_yuanma\nacos-1.4.1
```

创建`nacos_config`数据库 (`distribution/conf/nacos-mysql.sql`)

在`application.properties`中开启mysql配置

```
1 ### If use MySQL as datasource:
2 spring.datasource.platform=mysql
3
4 ### Count of DB:
5 db.num=1
6
7 ### Connect URL of DB:
8 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
9 db.user.0=root
10 db.password.0=root
```

4) 进入<http://localhost:8848/nacos>，用户名和密码默认nacos



文档: 01 Alibaba微服务组件Nacos注册中心实...

链接: [http://note.youdao.com/noteshare?](http://note.youdao.com/noteshare?id=ff9ab83ebe09e367dc598cc844b5bb13&sub=WEBde1cef584766ad525f409465f8001898)

[id=ff9ab83ebe09e367dc598cc844b5bb13&sub=WEBde1cef584766ad525f409465f8001898](http://note.youdao.com/noteshare?id=ff9ab83ebe09e367dc598cc844b5bb13&sub=WEBde1cef584766ad525f409465f8001898)