

芋道源码 —— 纯源码解析博客

愿半生编码，如一生老友！



芋道 Nacos 极简入门

📖 总阅读量:38192次

☆☆☆ Spring Boot 项目实战

☆☆☆ Spring Cloud 项目实战

《Dubbo 实现原理与源码解析 —— 精品合集》

《Netty 实现原理与源码解析 —— 精品合集》

《Spring 实现原理与源码解析 —— 精品合集》

《MyBatis 实现原理与源码解析 —— 精品合集》

《Spring MVC 实现原理与源码解析 —— 精品合集》

《数据库实体设计合集》

《Spring Boot 实现原理与源码解析 —— 精品合集》

《Java 面试题 + Java 学习指南》

摘要: 原创出处 <http://www.iocoder.cn/Nacos/install/> 「芋道源码」欢迎转载，保留摘要，谢谢！

- 1. 概述
- 2. 单机部署（最简模式）
- 3. 单机部署（基于 MySQL 数据库）
- 4. 集群部署
- 5. 集成到 Spring Boot
- 6. 集成到 Spring Cloud
- 7. 集成到 Dubbo
- 666. 彩蛋

推荐阅读如下 Nacos 文章：

- 《芋道 Spring Cloud Alibaba 配置中心 Nacos 入门》
- 《芋道 Spring Boot 注册中心 Nacos 入门》
- 《芋道 Spring Cloud Alibaba 配置中心 Nacos 入门》
- 《芋道 Spring Cloud Alibaba 注册中心 Nacos 入门》

1. 概述

Nacos 是什么？其官方文档自我介绍如下：

FROM <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 帮助您更敏捷和容易地构建、交付和管理微服务平台。Nacos 是构建以“服务”为中心的现代应用架构（例如微服务范式、云原生范式）的服务基础设施。

看起来介绍看起来有点高大上，嘿嘿。Nacos 是 Dynamic Naming and Configuration Service 的缩写，动态命名和配置服务。正如其名，Nacos 是阿里开源的**注册中心 + 配置中心**服务。

目前，Nacos 对 Spring、Spring Boot、Spring Cloud、Dubbo 提供了非常好的支持，已经有蛮多大型互联网公司正在**使用**。

下面，我们先来一起对 Nacos 进行快速入门，美滋滋。

2. 单机部署（最简模式）

本小节，我们来最小化部署一个 Nacos 服务，适合**演示**或**学习**之用。

① 下载 Nacos

在 <https://github.com/alibaba/nacos/releases> 中，我们可下载 Nacos 的安装包。

- Windows 下载 `.zip` 包。

- Linux / Unix / Mac 下载 `.tar.gz` 包。

后面，进行解压。

② 启动 Nacos

在命令行中，可以输入如下命令启动：

- Windows 启动命令：`cmd startup.cmd`。
- Linux / Unix / Mac 启动命令：`sh startup.sh -m standalone`。

如下，是芳芳使用 Mac 启动成功的日志：

```
2019-02-17 11:50:37,169 INFO Nacos started successfully in stand alone
```

```
2019-02-17 11:50:40,993 INFO Initializing Servlet 'dispatcherServlet'
```

```
2019-02-17 11:50:41,003 INFO Completed initialization in 10 ms
```

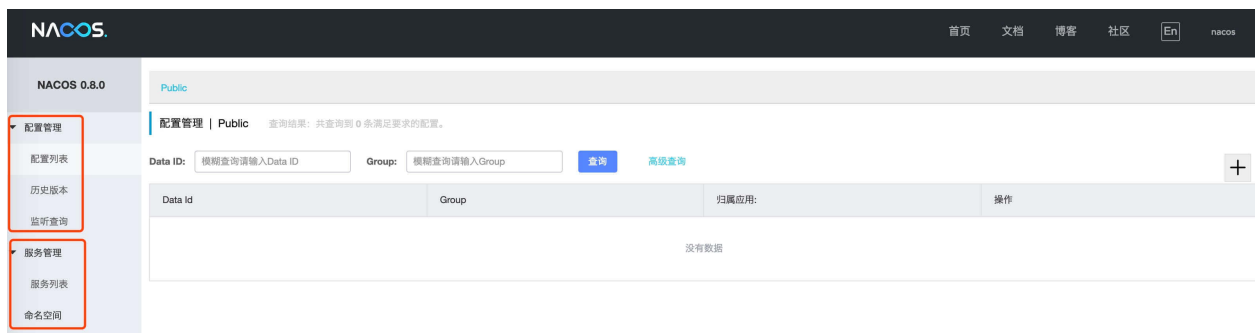


③ 管理界面

浏览器中，打开 `http://127.0.0.1:8848/nacos`，进行登录：

- 账号：`nacos`
- 密码：`nacos`

登录成功后，我们可以看到如下界面：



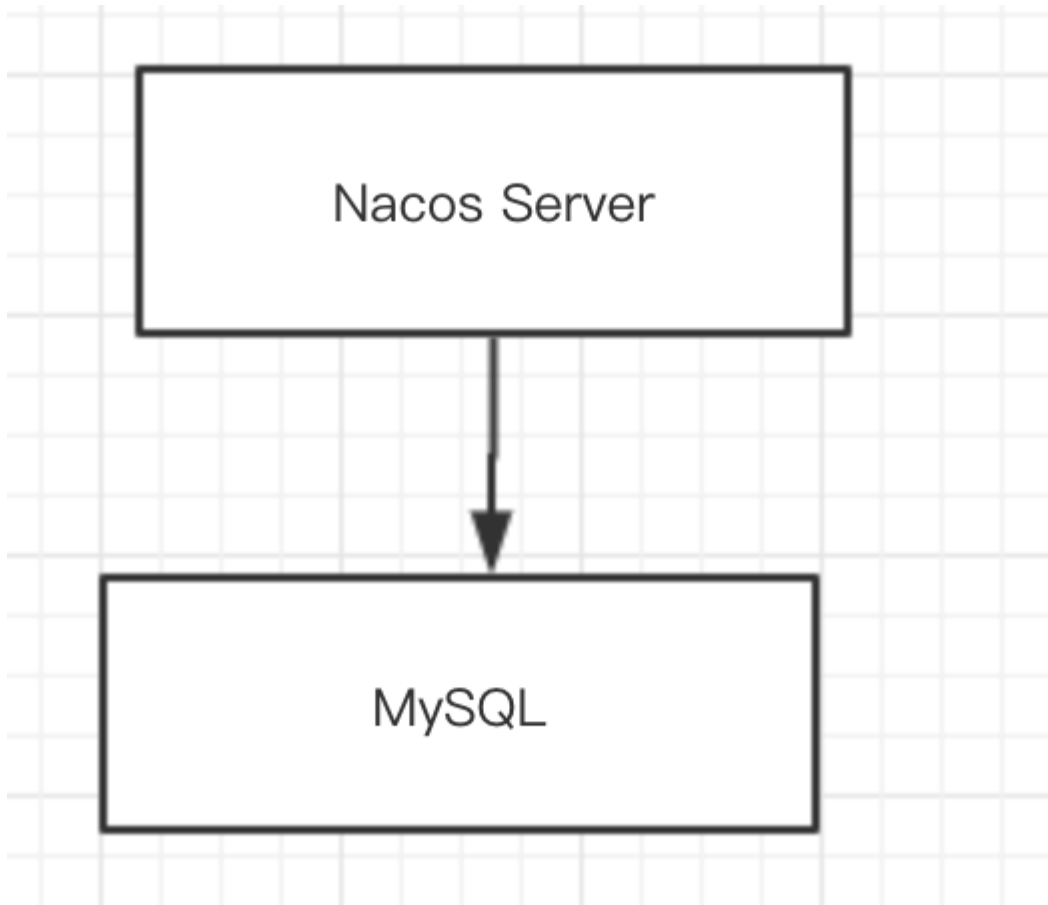
- 🐱 还是蛮好看的。

3. 单机部署（基于 MySQL 数据库）

操作系统：macOS 10.14

其它系统，基本一致的。

本小节，我们来单机部署一个 Nacos 服务，适合**测试**环境。如下图所示：



- 其中，MySQL 用于 Nacos 对配置的持久化。

友情提示：在「[3. 单机部署（最简模式）](#)」小节，Nacos 使用内嵌数据库 [Derby](#)。

3.1 MySQL 安装

相信大家都会，茈茈就不瞎哔哔了。嘿嘿~

3.2 Nacos 安装

① 下载 Nacos

打开 [Nacos 下载页面](#)，选择想要的 Nacos 版本。这里，我们选择 `1.1.4` 最新版本。

```
# 创建目录
$ mkdir -p /Users/yunai/Nacos
$ cd /Users/yunai/Nacos
```

```
# 下载
```

```
$ wget https://github.com/alibaba/nacos/releases/download/1.1.4/nacos-  
  
# 解压  
$ tar -zxvf nacos-server-1.1.4.tar.gz  
$ cd nacos  
  
# 查看目录  
$ ls -ls  
40 -rw-r--r--@ 1 yunai  staff  17336 Oct 11 14:09  LICENSE  
8 -rw-r--r--@ 1 yunai  staff   1305 Oct 11 14:09  NOTICE  
0 drwxr-xr-x  6 yunai  staff    192 Jan 20 22:26  bin # 执行脚本  
0 drwxr-xr-x@ 8 yunai  staff    256 Nov  4 10:26  conf # 配置文件  
0 drwxr-xr-x  3 yunai  staff     96 Jan 20 22:26  target # jar 包（目前
```

② 初始化数据库

在 `conf` 目录下，提供了 MySQL 数据库初始化脚本 `nacos-mysql.sql`。

我们可以根据自己的需要，创建一个库。例如说，茈茈这里创建一个 `nacos-example` 库。然后，使用该脚本进行数据库的初始化。最终表结构，如下图所示：

Name	Rows	Data Length	Engine	Created Date	Modified Date	Collation	Comment
config_info	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	config_info
config_info_aggr	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	增加租户字段
config_info_beta	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	config_info_beta
config_info_tag	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	config_info_tag
config_tags_relation	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	config_tag_relation
group_capacity	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	集群、各Group容量信息表
his_config_info	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	多租户改造
roles	1	16.00 KB	InnoDB	2020-01-20 22:58:38	2020-01-20 22:58:38	utf8mb4_0900_ai_ci	
tenant_capacity	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	租户容量信息表
tenant_info	0	16.00 KB	InnoDB	2020-01-20 22:58:38		utf8_bin	tenant_info
users	1	16.00 KB	InnoDB	2020-01-20 22:58:38	2020-01-20 22:58:38	utf8mb4_0900_ai_ci	

③ 配置数据库连接

修改 `conf/application.properties` 配置文件，在尾部额外增加 MySQL 数据库配置如下：

```
# 数据源为 MySQL
spring.datasource.platform=mysql

# 数据源的数量。因为这里我们只配置一个数据源，所以设置为 1。
db.num=1
# 第 0 个数据源的配置
db.url.0=jdbc:mysql://127.0.0.1:3306/nacos-example?characterEncoding=u
# 数据库的账号密码
db.user=root
db.password=
```

④ MySQL8 的支持

如果胖友使用的 MySQL 是 8.X 版本，则需要看该步骤。否则，可以直接跳过。

默认情况下，Nacos 连接 MySQL 5.6.5+ 版本，所以使用的 MySQL JDBC 驱动。如果胖友和芳芳一样，准备使用 MySQL 8.X 版本，则需要下载对应的新版本的 JDBC 驱动。Nacos 提供插件机制，允许我们无需修改源码，直接可以支持，美滋滋，操作步骤如下：

```
# 创建插件目录
$ mkdir -p plugins/mysql

# 下载 MySQL 8.X JDBC 驱动
$ cd plugins/mysql
$ wget https://repo1.maven.org/maven2/mysql/mysql-connector-java/8.0.1
```

⑤ 启动 Nacos 服务

执行 `sh bin/startup.sh -m standalone` 命令，启动 Nacos 服务。其中，`-m standalone` 表示单机模式。

启动完成后，可以在 `logs/start.out` 日志文件中，查看启动日志。如果看到如下内容，说明启动是成功的。

```
2020-01-20 23:34:08,358 INFO Nacos started successfully in stand alone
2020-01-20 23:34:09,695 INFO Initializing Servlet 'dispatcherServlet'
```

```
2020-01-20 23:34:09,706 INFO Completed initialization in 11 ms
```

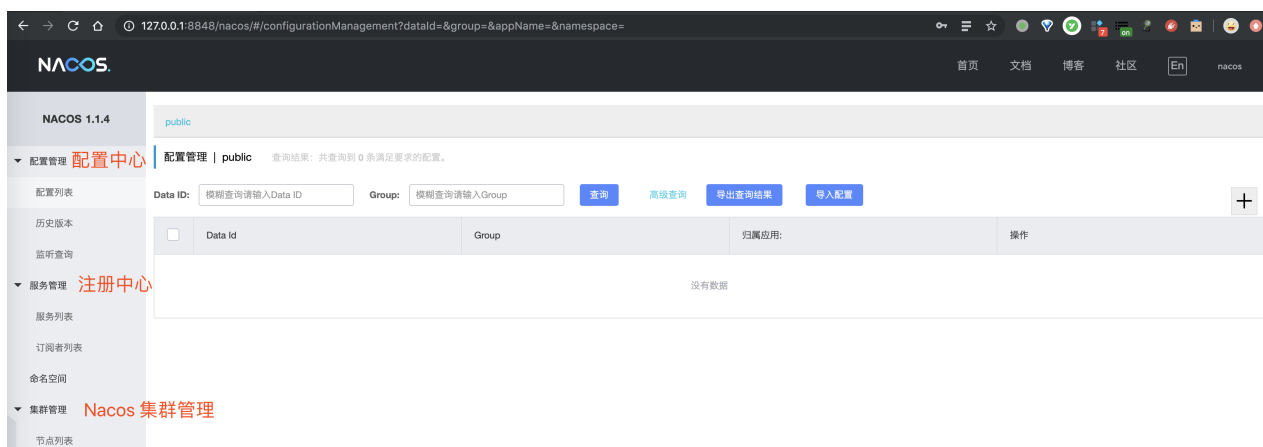
默认配置下，Nacos 服务使用 **8848** 端口。😈 嘿嘿，8848 的含义，胖友是否知道。

⑥ 访问 UI 界面

Nacos 服务，内置了 UI 界面。使用浏览器中，打开 <http://127.0.0.1:8848/nacos> 地址，使用默认的用户「nacos/nacos」进行登录。

友情提示：生产环境下，一定要记得修改默认的用户密码噢。

登录成功后，我们可以看到如下界面：

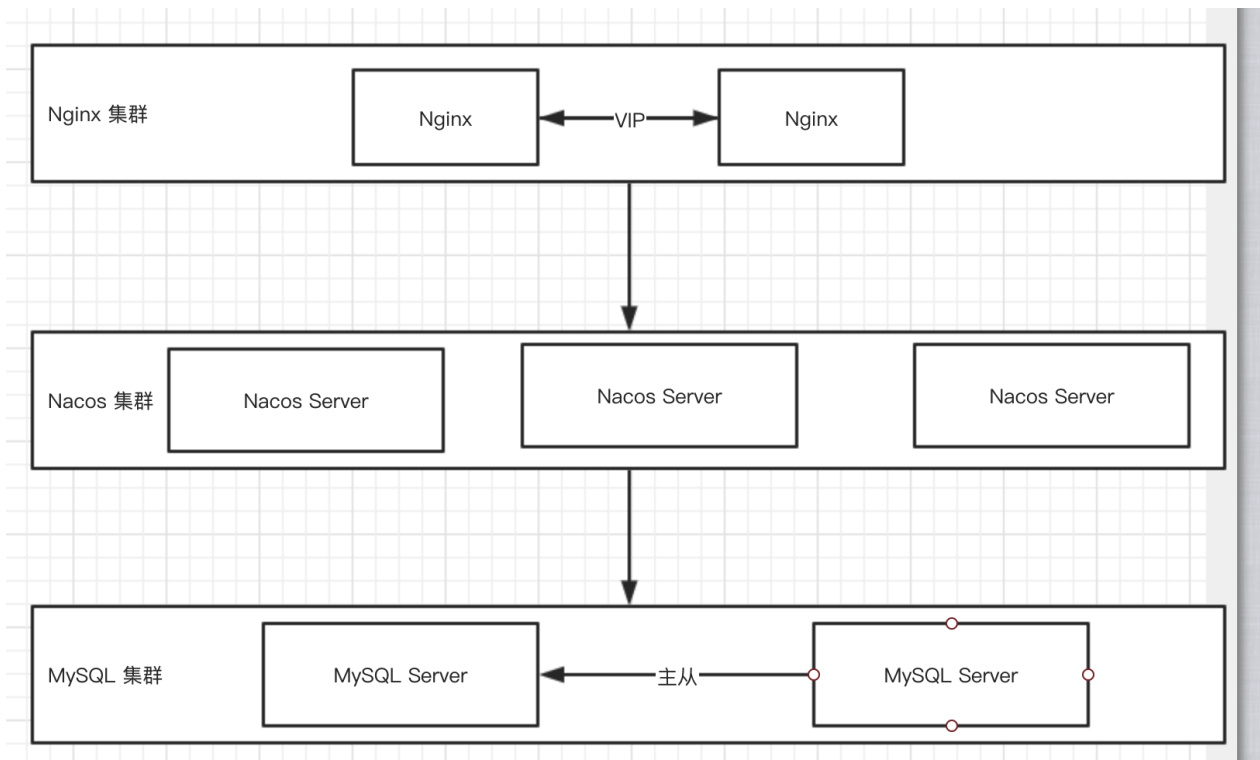


界面还是蛮好看的，采用的是 @alifd/next 前端 UI 框架。

4. 集群部署

友情提示：本小节的内容，在「3. 单机部署（基于 MySQL 数据库）」小节之上，不要跳过来看噢。

在生产环境下，我们需要搭建 Nacos 集群，确保高可用。所以本小节，我们来搭建单机部署一个 Nacos 服务。如下图所示：



- 一共是 Nginx、Nacos、MySQL 三层，每一层都需要做高可用。

4.1 MySQL 安装

搭建 MySQL 集群，保证 MySQL 高可用。具体的方案，可见《[MySQL 高可用方案选型参考](#)》。如果使用阿里云的话，可以考虑直接使用 RDS 服务，减少运维成本。

考虑到让示例更简单一些，茈茈这里暂时只使用 MySQL 单节点。

4.2 Nacos 安装

我们在本地搭建 3 个 Nacos 服务，实现 Nacos 高可用。因为 Nacos 服务之间通过 [Raft 算法](#) 保证一致性，所以我们建议 Nacos 部署的节点数为大于 3 的奇数。

① 复制 nacos 文件

将「[2. 单机部署](#)」的 `nacos` 文件，复制 `nacos-01`、`nacos-02`、`nacos-03` 三个文件夹，用于搭建三个 Nacos 节点。操作命令如下：

```
# 复制
$ cp -r nacos nacos-01
$ cp -r nacos nacos-02
$ cp -r nacos nacos-03
```

```
# 查看目录
```



```
$ ls -ls
    0 drwxr-xr-x  11 yunai  staff      352 Jan 20 23:33 nacos
    0 drwxr-xr-x  11 yunai  staff      352 Jan 21 09:22 nacos-01
    0 drwxr-xr-x  11 yunai  staff      352 Jan 21 09:22 nacos-02
    0 drwxr-xr-x  11 yunai  staff      352 Jan 21 09:22 nacos-03
102408 -rw-r--r--@  1 yunai  staff 52115827 Jan 20 21:58 nacos-server
```

② 初始化数据库

继续使用「2.2 单机部署」初始化好的数据库 `nacos-example`。

③ 配置数据库连接

对 `nacos-01`、`nacos-02`、`nacos-03` 三个文件夹，都修改

`conf/application.properties` 配置文件，在尾部额外增加 MySQL 数据库配置如下：

```
# 数据源为 MySQL
spring.datasource.platform=mysql

# 数据源的数量。因为这里我们只配置一个数据源，所以设置为 1。
db.num=1
# 第 0 个数据源的配置
db.url.0=jdbc:mysql://127.0.0.1:3306/nacos-example?characterEncoding=u
# 数据库的账号密码
db.user=root
db.password=
```



- 如果使用 MySQL 集群，可以修改 `db.num` 为 MySQL 节点数量，然后在 `db.url.1`、`db.url.2` 不断往下配置。

④ 配置服务器地址

因为我们是本机启动三个 Nacos 服务，所以需要修改其端口，保证能够启动。在 `conf/application.properties` 配置文件，修改 `server.port` 配置项，可以修改 Nacos 服务器端口。这里我们分别修改如下：

- `nacos-01` 对应 18848 端口。
- `nacos-02` 对应 28848 端口。
- `nacos-03` 对应 38848 端口。

⑤ 配置 Nacos 集群

在 `nacos-01`、`nacos-02`、`nacos-03` 三个文件夹中，创建 `conf/cluster.conf` 配置文件，配置一个 Nacos 集群的所有节点。具体内容如下：

```
# ip:port
192.168.3.44:18848
192.168.3.44:28848
192.168.3.44:38848
```

- 每一行为 Nacos 节点的服务器地址，格式为 `ip:port`。
- 注意，不要使用 `127.0.0.1` 这个 IP 地址，因为 Nacos 获取的是外部 IP。因此，芋芋这里使用的是 `192.168.3.44`。

通过该配置文件，每个 Nacos 服务可以知道集群中的其它 Nacos 节点。

⑥ 启动 Nacos 服务

现在，让我们来启动三个 Nacos 服务。

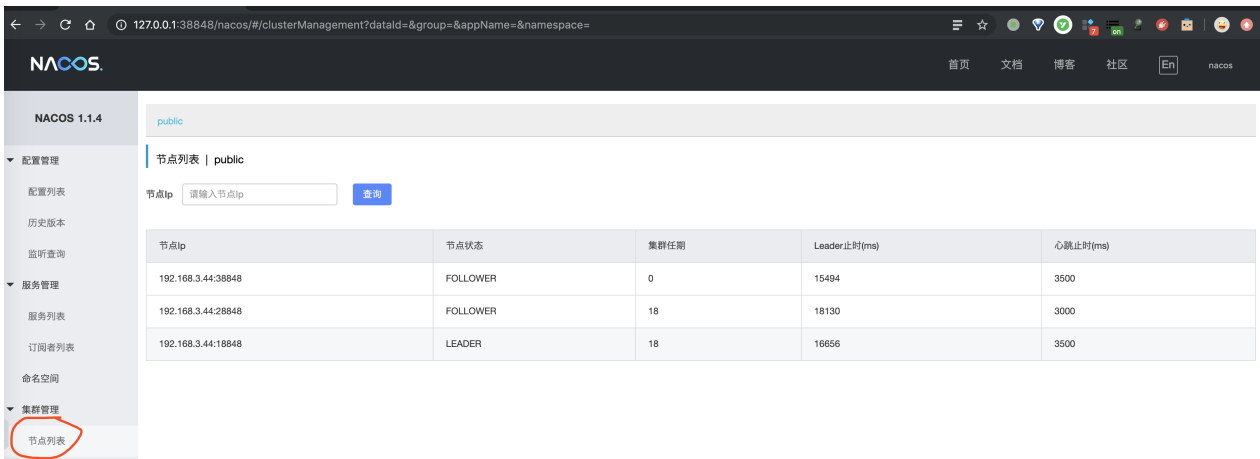
- 执行 `sh nacos-01/bin/startup.sh` 命令，启动 Nacos 节点 01。
- 执行 `sh nacos-02/bin/startup.sh` 命令，启动 Nacos 节点 02。
- 执行 `sh nacos-03/bin/startup.sh` 命令，启动 Nacos 节点 03。

每个 Nacos 节点是否启动成功，胖友自己去看看 `logs/start.out` 日志文件。

⑦ 访问 UI 界面

三个节点启动成功后，我们访问 Nacos 节点 01 的 UI 界面。使用浏览器中，打开 <http://127.0.0.1:18848/nacos> 地址，使用默认的用户「nacos/nacos」进行登录。

登录成功后，点击「节点列表」菜单，我们就可以看到搭建的 Nacos 集群信息。界面如下：



4.3 Nginx 安装

搭建 Nginx 集群，保证 Nginx 高可用。具体的方案，可见《[Configuring Active-Active High Availability and Additional Passive Nodes with keepalived](#)》。如果使用阿里云的话，可以考虑直接使用 SLB 服务，减少运维成本。

考虑到让示例更简单一些，茈茈这里暂时只使用 Nginx 单节点。🐱 因为茈茈主要使用 Tengine 为主，所以接下来，我们使用 Tengine，哈哈哈~

① 安装 Tengine

参考《[芋道 Tengine 极简入门](#)》文章，先搭建一个 Tengine 服务。

② 配置转发

搭建完成后，我们配置 Tengine 配置文件 `conf/nginx.conf` 如下，增加对 Nacos 服务的负载均衡。主要配置如下：

```
# // 省略其它配置内容...

http {
    # // 省略其它配置内容...

    upstream nacosCluster {
        server 192.168.3.44:18848;
        server 192.168.3.44:28848;
        server 192.168.3.44:38848;
    }

    location /nacos/ {
        proxy_pass http://nacosCluster/nacos/;
        #allow SOME.IP.ADD.RESS;
```

```
        #deny all;  
    }  
}
```

配置完成后，记得使用 `sbin/nginx -s reload` 命令，重新让 Tengine 加载下最新配置噢。

③ 简单测试

友情提示：芬芳的 Tengine 是安装在 IP 为 172.16.48.180 服务器上。

使用浏览器，访问 <http://172.16.48.180/nacos> 地址，就可以访问 Nacos 集群服务的 UI 界面。

后续，应用中可以直接使用该 Nginx 负载均衡的地址。当然，推荐给 Nginx 绑定一个域名，这样未来 Nginx 服务及时迁移服务器，对于应用也是可以透明的，无需做任何改动。例如说，将 <http://nacos.iocoder.cn> 转发到 Nginx 服务上。

4.4 补充说明

至此，我们已经完成了 Nacos 集群部署的搭建。生产环境下，请将 Nacos 多个节点部署到多个服务器，保证 Nacos 的高可用。

更多内容，胖友也可以阅读《[Nacos 文档 —— 集群部署说明](#)》文章。

5. 集成到 Spring Boot

Nacos 集成到 Spring Boot 项目中，通过 [nacos-spring-boot-project](#) 项目。

- 作为**配置中心**，可见《[芋道 Spring Cloud Alibaba 配置中心 Nacos 入门](#)》文章。
- 作为**注册中心**，可见《[芋道 Spring Boot 注册中心 Nacos 入门](#)》文章。

6. 集成到 Spring Cloud

Nacos 集成到 Spring Cloud 项目中，通过 [spring-cloud-alibaba](#) 项目。

- 作为**配置中心**，可见《[芋道 Spring Cloud Alibaba 配置中心 Nacos 入门](#)》文章。

- 作为**注册中心**，可见《芋道 Spring Cloud Alibaba 注册中心 Nacos 入门》文章。

7. 集成到 Dubbo

Nacos 集成到 Dubbo 项目中：

- 作为**注册中心**，通过 `dubbo-registry-nacos` 项目，可见《芋道 Spring Boot Dubbo 入门》的「6. 整合 Nacos」小节。
- 作为**配置中心**，通过 `dubbo-metadata-report-nacos` 项目，TODO

666. 彩蛋

如果对源码感兴趣的胖友，可以看看《Nacos 源码解析》噢。

☐ Nacos

☐ ☐ ☐ ☐ ☐ ☐

PREVIOUS:

☐ 芋道 CAT 极简入门（应用监控平台）

NEXT:

☐ 芋道 Soul 极简入门（国产微服务网关）

博主【芋芳】在看的课程

【老牛逼了】Dubbo 源码解析系列

Netty 源码解析系列

Spring 源码解析系列

Spring MVC 源码解析系列

Spring Boot 源码解析系列

MyBatis 源码解析系列

数据库实体设计合集

【老牛逼了】Java 面试题

Spring Boot 学习路线

Spring Cloud 学习路线



扫一扫二维码关注公众号

关注后，可以看到

「RocketMQ」

「MyCAT」

所有源码解析文章

—— 近期更新「Sharding-JDBC」中 ——

你有233个小伙伴已经关注

微信公众号福利：芋道源码

0. 阅读源码葵花宝典

1. RocketMQ / MyCAT / Sharding-JDBC 详细中文注释源码

2. 您对于源码的疑问每条留言都将得到认真回复

3. 新的源码解析文章实时收到通知，每周六十点更新

4. 认真的源码交流微信群

分类

APISIX ¹

ActiveMQ ²

Apollo ³⁵

CAT ¹

Canal ⁷

Elastic-Job-Cloud ⁶

Elastic-Job-Lite ¹⁶

Elasticsearch ³

Eureka ²⁴

Fescar ⁵

Guava ¹³

HikariCP ²¹

Hmily ¹

Hystrix ¹²

IDEA ⁴

JDK 源码 ⁵

JUC ³⁶

JVM ¹

Java 面试 ¹²

Jenkins ¹

Jetty ⁶

Kafka ²⁷

Kong ¹

Maven ¹

MongoDB ³

MyBatis ⁷

MyCAT ⁹

Nacos ¹³

Netty ¹¹

Nginx ³

NodeJS ²

Onemall ¹

Prometheus ¹

RabbitMQ ²

Redis ³

Resilience4j ¹²

Ribbon ⁹

RocketMQ ²⁹

RxJava ⁷

SOFA Mosn ¹

SOFA RPC ⁴

Seata ⁶

Sentinel ¹⁷

Sentry ¹

Sharding Sphere ⁶

Sharding-JDBC ¹⁹

Shiro ⁸

SkyWalking ⁴²

Solr ¹

Soul ¹

Spring ²⁴

Spring Boot ⁹⁹

Spring Cloud ³³

Spring Security ³⁴

Spring Session ¹

Spring Webflux ⁸

Spring-Cloud-Config ¹

Spring-Cloud-Gateway ²⁶

Spring-MVC ¹³

TCC-Transaction ⁷

Tomcat ¹⁸

XXL-JOB ¹

Yudao ²

Zipkin ¹¹

Zookeeper ²

Zuul ⁸

书单 ³³

工作内推 ⁴

性能测试 ⁹

数据结构与算法 ⁴

电商开源项目 ²

精进 ⁴⁷⁹¹

芋道源码的周八 ²⁰

设计模式 ²⁵

鸡汤 ⁷

芋道快速开发平台 Boot + Cloud

星主：芋道源码



知识星球

© 2023 芋道源码 && 总访客数 6972156 次 && 总访问量 15663691 次 && Hosted by **Coding Pages**

沪ICP备17037075号-1