



- 服务管理：实现服务CRUD，域名CRUD，服务健康状态检查，服务权重管理等功能
- 配置管理：实现配置管CRUD，版本管理，灰度管理，监听管理，推送轨迹，聚合数据等功能
- 元数据管理：提供元数据CRUD 和打标能力
- 插件机制：实现三个模块可分可合能力，实现扩展点SPI机制
- 事件机制：实现异步化事件通知，sdk数据变化异步通知等逻辑
- 日志模块：管理日志分类，日志级别，日志可移植性（尤其避免冲突），日志格式，异常码+帮助文档
- 回调机制：sdk通知数据，通过统一的模式回调用户处理。接口和数据结构需要具备可扩展性
- 寻址模式：解决ip，域名，nameserver、广播等多种寻址模式，需要可扩展
- 推送通道：解决server与存储、server间、server与sdk间推送性能问题
- 容量管理：管理每个租户，分组下的容量，防止存储被写爆，影响服务可用性
- 流量管理：按照租户，分组等多个维度对请求频率，长链接个数，报文大小，请求流控进行控制
- 缓存机制：容灾目录，本地缓存，server缓存机制。容灾目录使用需要工具
- 启动模式：按照单机模式，配置模式，服务模式，dns模式，或者all模式，启动不同的程序+UI
- 一致性协议：解决不同数据，不同一致性要求情况下，不同一致性机制
- 存储模块：解决数据持久化、非持久化存储，解决数据分片问题
- Nameserver：解决namespace到clusterid的路由问题，解决用户环境与nacos物理环境映射问题
- CMDB：解决元数据存储，与三方cmdb系统对接问题，解决应用，人，资源关系
- Metrics：暴露标准metrics数据，方便与三方监控系统打通
- Trace：暴露标准trace，方便与SLA系统打通，日志白平化，推送轨迹等能力，并且可以和计量计费系统打通
- 接入管理：相当于阿里云开通服务，分配身份、容量、权限过程
- 用户管理：解决用户管理，登录，sso等问题
- 权限管理：解决身份识别，访问控制，角色管理等问题
- 审计系统：扩展接口方便与不同公司审计系统打通
- 通知系统：核心数据变更，或者操作，方便通过SMS系统打通，通知到对应人数据变更
- OpenAPI：暴露标准Rest风格HTTP接口，简单易用，方便多语言集成
- Console：易用控制台，做服务管理、配置管理等操作
- SDK：多语言sdk
- Agent：dns-f类似模式，或者与mesh等方案集成
- CLI：命令行对产品进行轻量化管理，像git一样好用

1.2 Nacos 部署

1.2.1 Nacos 部署说明

Nacos 快速开始

<https://nacos.io/zh-cn/docs/v2/quickstart/quick-start.html>

版本选择

当前推荐的稳定版本为2.X

<https://github.com/alibaba/nacos/releases>

Nacos支持三种部署模式

- 单机模式 - 用于测试和单机试用。
- 集群模式 - 用于生产环境，确保高可用。

- 多集群模式 - 用于多数据中心场景。

环境准备

- 安装好 JDK，需要 1.8 及其以上版本
- 建议: 2核 CPU / 4G 内存 及其以上
- 建议: 生产环境 3 个节点 及其以上

1.2.2 单机部署

1.2.2.1 预备环境准备

Nacos 依赖 [Java](#) 环境来运行。如果您是从代码开始构建并运行Nacos，还需要为此配置 [Maven](#)环境，请确保是在以下版本环境中安装使用:

1. 64 bit OS, 支持 Linux/Unix/Mac/Windows, 推荐选用 Linux/Unix/Mac。
2. 64 bit JDK 1.8+; [下载 & 配置](#)。
3. Maven 3.2.x+; [下载 & 配置](#)。

1.2.2.2 下载源码或者安装包

你可以通过源码和发行包两种方式来获取 Nacos。

从 Github 上下载源码方式

```
git clone https://github.com/alibaba/nacos.git
cd nacos/
mvn -Prelease-nacos -Dmaven.test.skip=true clean install -U
ls -al distribution/target/

// change the $version to your actual path
cd distribution/target/nacos-server-$version/nacos/bin
```

下载编译后压缩包方式

您可以从 [最新稳定版本](#) 下载 `nacos-server-$version.zip` 包。

```
unzip nacos-server-$version.zip 或者 tar -xvf nacos-server-$version.tar.gz
cd nacos/bin
```

范例:

```
[root@ubuntu2204 ~]#apt update && apt -y install openjdk-11-jdk
[root@ubuntu2204 ~]#apt update && apt -y install openjdk-8-jdk

[root@ubuntu2204 ~]#wget https://github.com/alibaba/nacos/releases/download/2.2.3/nacos-server-2.2.3.tar.gz
[root@ubuntu2204 ~]#tar xf nacos-server-2.2.3.tar.gz -C /usr/local/
[root@ubuntu2204 ~]#ls /usr/local/nacos/
bin  conf  data  LICENSE  logs  NOTICE  target

[root@ubuntu2204 ~]#ls /usr/local/nacos/bin/
shutdown.cmd  shutdown.sh  startup.cmd  startup.sh

[root@ubuntu2204 ~]#ls /usr/local/nacos/target/
nacos-server.jar

[root@ubuntu2204 ~]#ls /usr/local/nacos/conf/
1.4.0-ipv6_support-update.sql  application.properties  cluster.conf.example  mysql-schema.sql
announcement.conf            application.properties.example  derby-schema.sql      nacos-logback.xml

[root@ubuntu2204 ~]#ls /usr/local/nacos/data
derby-data  naming  protocol

#修改配置, 可选
[root@nacos01 ~]#vi /usr/local/nacos/conf/application.properties
#修改默认的访问URL路径
#server.servlet.contextPath=/nacos
server.servlet.contextPath=/
```

1.2.2.3 启动和关闭服务器

注: Nacos的运行建议至少在2C4G 60G的机器配置下运行。

1.2.2.3.1 启动服务

Linux/Unix/Mac

启动命令(standalone代表着单机模式运行, 非集群模式):

```
sh startup.sh -m standalone
```

如果您使用的是ubuntu系统, 或者运行脚本报错提示[[符号找不到, 可尝试如下运行:

```
bash startup.sh -m standalone
```

启动命令(standalone代表着单机模式运行, 非集群模式):

范例:

1.2.2.3.2 关闭服务

Windows

或者双击shutdown.cmd运行文件。

1.2.2.4 服务注册&发现和配置管理

服务注册

```
curl -X POST 'http://127.0.0.1:8848/nacos/v1/ns/instance?
serviceName=nacos.wang.serviceName&ip=1.2.3.4&port=8080'
```

服务发现

```
curl -X GET 'http://127.0.0.1:8848/nacos/v1/ns/instance/list?serviceName=nacos.wang.serviceName'
```

发布配置

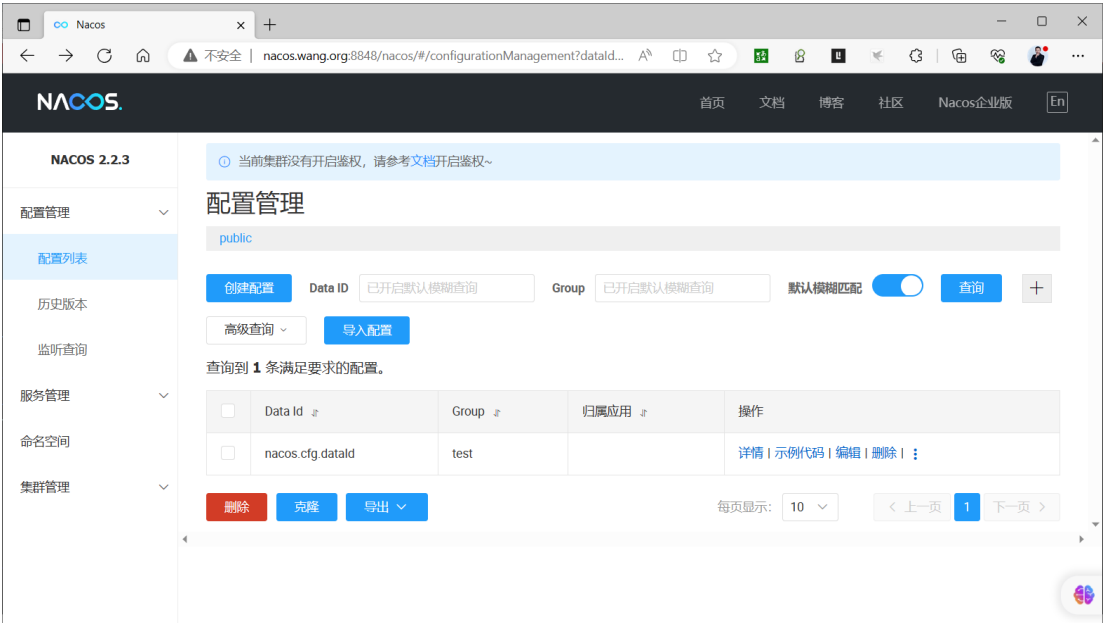
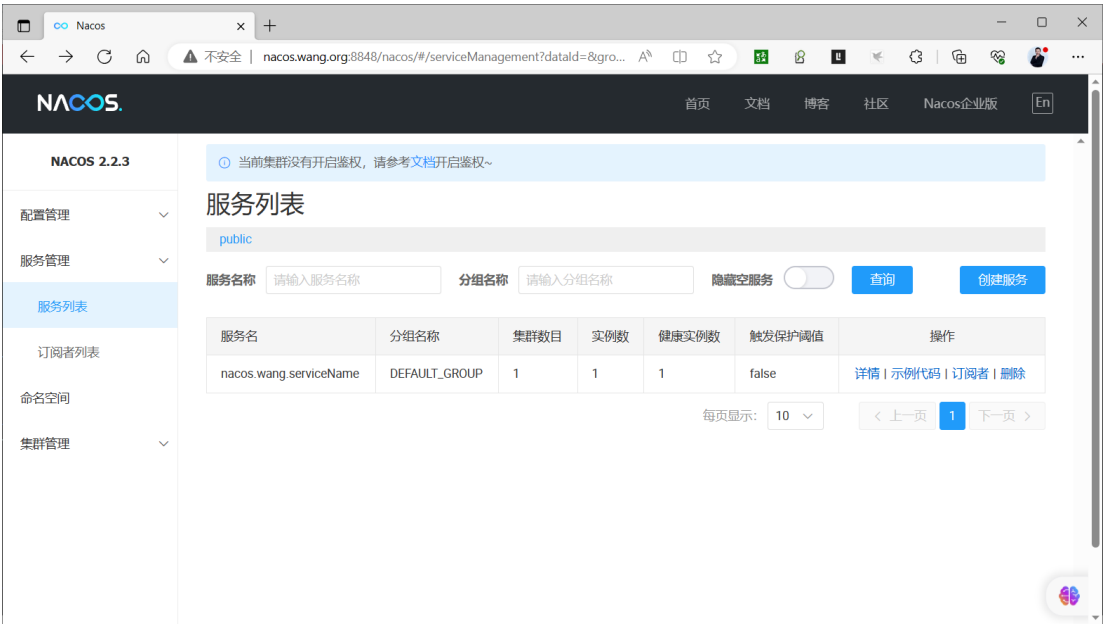
```
curl -X POST "http://127.0.0.1:8848/nacos/v1/cs/configs?dataId=nacos.cfg.dataId&group=test&content=HelloWorld"
```

获取配置

```
curl -X GET "http://127.0.0.1:8848/nacos/v1/cs/configs?dataId=nacos.cfg.dataId&group=test"
```

1.2.2.5 Web 访问

```
http://nacos.wang.org:8848/nacos
```



1.2.2.6 单机模式支持 MySQL

在0.7版本之前，在单机模式时nacos使用嵌入式数据库实现数据的存储，不方便观察数据存储的基本情况。0.7版本增加了支持mysql数据源能力，具体的操作步骤：

- 安装数据库，版本要求：5.6.5+

- 创建数据库nacos和用户nacos，并授权
- 执行数据库初始化脚本 mysql-schema.sql
- 修改conf/application.properties文件，增加支持mysql数据源配置（目前只支持mysql），添加mysql数据源的url、用户名和密码。

```
spring.datasource.platform=mysql

db.num=1
db.url.0=jdbc:mysql://127.0.0.1:3306/nacos?
characterEncoding=utf8&connectTimeout=1000&socketTimeout=3000&autoReconnect=true
db.user=nacos
db.password=123456
```

再以单机模式启动nacos，nacos所有写嵌入式数据库的数据都写到了mysql

范例:

```
[root@ubuntu2204 ~]#apt update && apt -y install mysql-server

[root@ubuntu2204 ~]#mysql
Server version: 8.0.33-0ubuntu0.22.04.4 (Ubuntu)

mysql> create user nacos@'127.0.0.1' identified with mysql_native_password by 'nacos';
mysql> create database nacos;
mysql> grant all on nacos.* to nacos@'10.0.0.%' ;

[root@ubuntu2204 ~]#mysql -unacos -pnacos -h127.0.0.1 nacos < /usr/local/nacos/conf/mysql-schema.sql

[root@ubuntu2204 ~]#vim /usr/local/nacos/conf/application.properties

***** Config Module Related Configurations *****#
### If use MySQL as datasource:
### Deprecated configuration property, it is recommended to use `spring.sql.init.platform` replaced.
# spring.datasource.platform=mysql
# spring.sql.init.platform=mysql
spring.sql.init.platform=mysql #加此行

### Count of DB:
# db.num=1
db.num=1 #加此行

### 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=nacos
# db.password.0=nacos

##加下面三行
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=nacos
db.password.0=nacos

#如果启动,先关闭服务
[root@ubuntu2204 ~]# /usr/local/nacos/bin/shutdown.sh
The nacosServer(4336) is running...
Send shutdown request to nacosServer(4336) OK

[root@ubuntu2204 ~]# /usr/local/nacos/bin/startup.sh -m standalone
/usr/lib/jvm/java-8-openjdk-amd64/bin/java -Djava.ext.dirs=/usr/lib/jvm/java-8-openjdk-amd64/jre/lib/ext:/usr/lib/jvm/java-8-openjdk-amd64/lib/ext -Xms512m -Xmx512m -Xmn256m -Dnacos.standalone=true -Dnacos.member.list= -Xloggc:/usr/local/nacos/logs/nacos_gc.log -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M -Dloader.path=/usr/local/nacos/plugins,/usr/local/nacos/plugins/health,/usr/local/nacos/plugins/cmdb,/usr/local/nacos/plugins/selector -Dnacos.home=/usr/local/nacos -jar /usr/local/nacos/target/nacos-server.jar --spring.config.additional-location=file:/usr/local/nacos/conf/ --logging.config=/usr/local/nacos/conf/nacos-logback.xml --server.max-http-header-size=524288
nacos is starting with standalone
nacos is starting, you can check the /usr/local/nacos/logs/start.out

#查看日志
[root@ubuntu2204 ~]#tail -f /usr/local/nacos/logs/start.out

      ,--.
     ,--.'|
    ,--,: :|
   ,--.'| | :
  | : : | |
  : | \ | : ,--.--.
  | : ' | ; | / \ / \ ; ,. : | : / \

      ,--.
     ,--.'|
    ,--,: :|
   ,--.'| | :
  | : : | |
  : | \ | : ,--.--.
  | : ' | ; | / \ / \ ; ,. : | : / \

Nacos 2.2.3
Running in stand alone mode, All function modules
Port: 8848
Pid: 6587
Console: http://10.0.0.200:8848/nacos/index.html
```

```
https://nacos.io

2023-06-04 17:30:59,801 INFO Tomcat initialized with port(s): 8848 (http)

2023-06-04 17:30:59,884 INFO Root WebApplicationContext: initialization completed in 3317 ms

2023-06-04 17:31:04,055 INFO Adding welcome page: class path resource [static/index.html]

2023-06-04 17:31:04,804 WARN You are asking Spring Security to ignore Ant [pattern='/**']. This is not recommended -- please use permitAll via HttpSecurity#authorizeHttpRequests instead.

2023-06-04 17:31:04,805 INFO Will not secure Ant [pattern='/**']

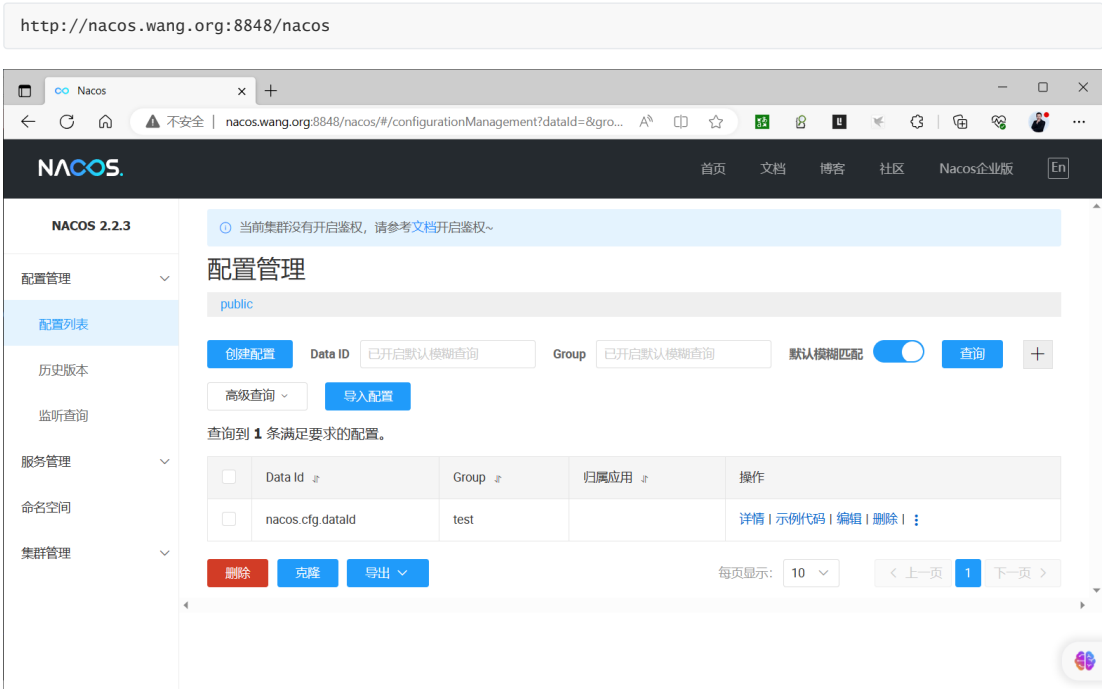
2023-06-04 17:31:04,838 INFO Will secure any request with
[org.springframework.security.web.context.request.async.WebAsyncManagerIntegrationFilter@ec0c838,
org.springframework.security.web.context.SecurityContextPersistenceFilter@f9b7332,
org.springframework.security.web.header.HeaderWriterFilter@2c177f9e,
org.springframework.security.web.csrf.CsrfFilter@290b1b2e,
org.springframework.security.web.authentication.logout.LogoutFilter@b672aa8,
org.springframework.security.web.savedrequest.RequestCacheAwareFilter@6fefce9e,
org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter@1bdf8190,
org.springframework.security.web.authentication.AnonymousAuthenticationFilter@6e46d9f4,
org.springframework.security.web.session.SessionManagementFilter@209775a9,
org.springframework.security.web.access.ExceptionTranslationFilter@33617539]

2023-06-04 17:31:04,875 INFO Exposing 1 endpoint(s) beneath base path '/actuator'

2023-06-04 17:31:04,940 INFO Tomcat started on port(s): 8848 (http) with context path '/nacos'

2023-06-04 17:31:04,981 INFO Nacos started successfully in stand alone mode. use external storage
```

测试访问,可看到下面页面



1.2.2.7 启用登录验证

```
https://nacos.io/zh-cn/docs/auth.html
```

默认无需登录就可管理nacos,基于安全,可以启用登录验证功能

范例:

```
#生成token的值,至少32位,否则无法启动
[root@ubuntu2204 ~]#openssl rand -base64 33
Yfmvri6Kx8tuI+k0lRBr5nm2Wni1i5h0SFBREpPFZ36R

#修改配置文件,共四行内容
[root@ubuntu2204 ~]#vim /usr/local/nacos/conf/application.properties
***** Access Control Related Configurations *****#
```

```
.....
nacos.core.auth.system.type=nacos

#nacos.core.auth.enabled=false
nacos.core.auth.enabled=true #修改此行为true

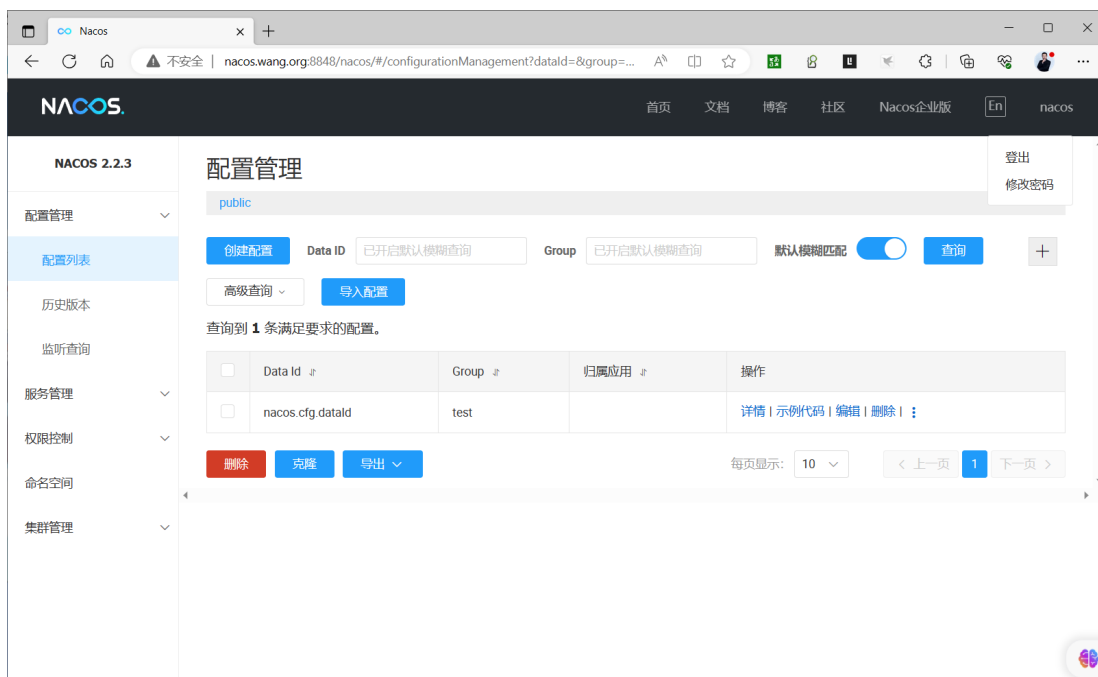
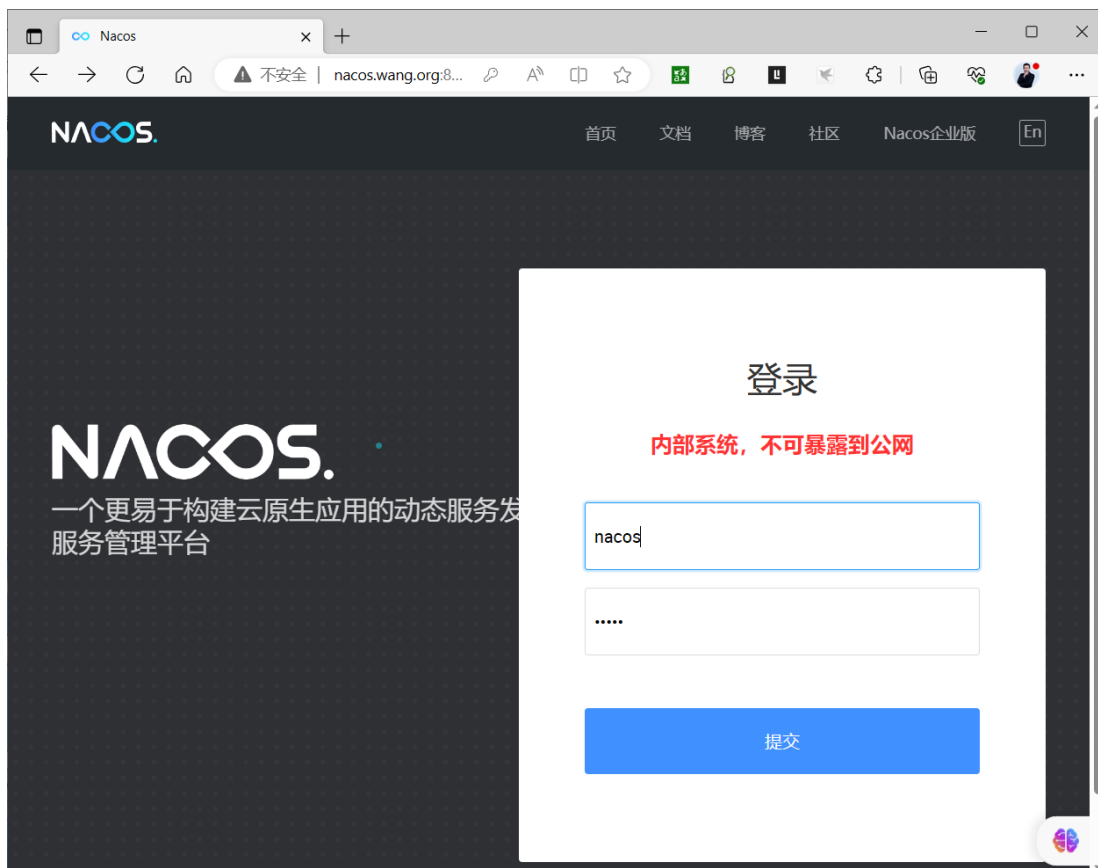
nacos.core.auth.server.identity.key=wang #指定值,否则无法启动
nacos.core.auth.server.identity.value=wang #指定值,否则无法启动

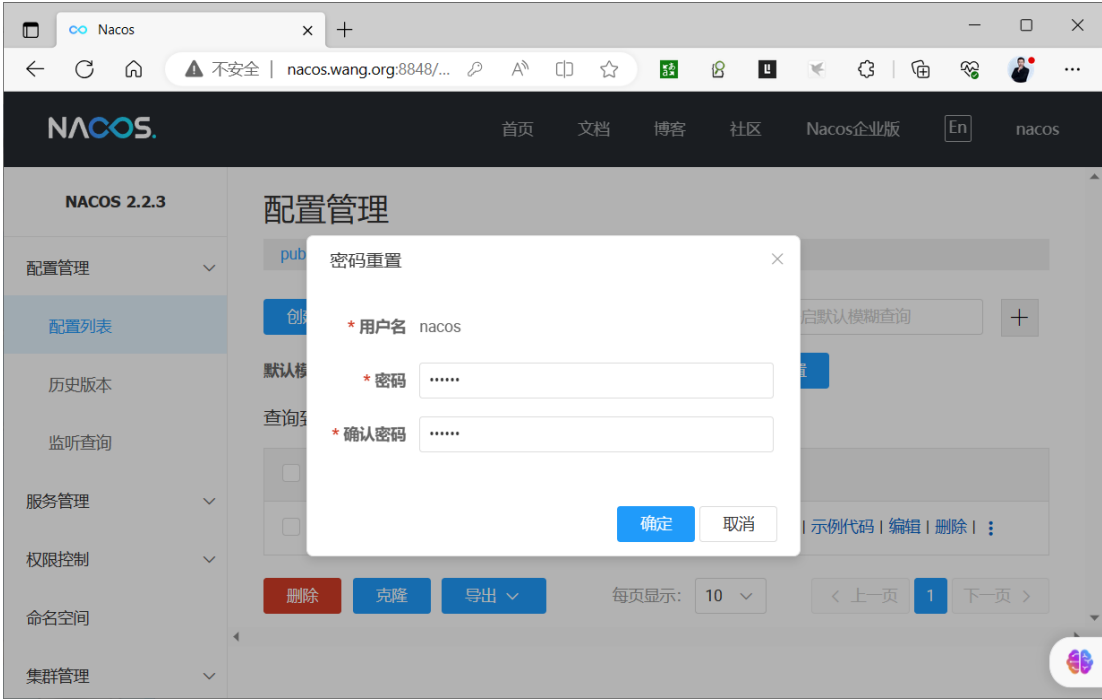
nacos.core.auth.plugin.nacos.token.secret.key=YfmvRi6Kx8tuI+k0lRBr5nm2wNi1i5h0SFBRepPFZ36R #添加前面生成的token的值

#重启服务生效
[root@ubuntu2204 ~]# /usr/local/nacos/bin/shutdown.sh
[root@ubuntu2204 ~]# /usr/local/nacos/bin/startup.sh -m standalone
```

测试访问,默认用户名和密码都是nacos

http://nacos.wang.org:8848/nacos





密码保存在nacos库中的users表中

```
[root@ubuntu2204 ~]#mysql -unacos -pnacos -h127.0.0.1 nacos -e 'select * from users'
mysql: [Warning] Using a password on the command line interface can be insecure.
+-----+-----+-----+
| username | password | enabled |
+-----+-----+-----+
| nacos    | $2a$10$EuWPZHzz32dJN7jexM34MOeYirDdFAZm2kuwj7VE0JhhZkDrxFvUu | 1 |
+-----+-----+-----+
```

1.2.3 集群部署

1.2.3.1 集群部署说明

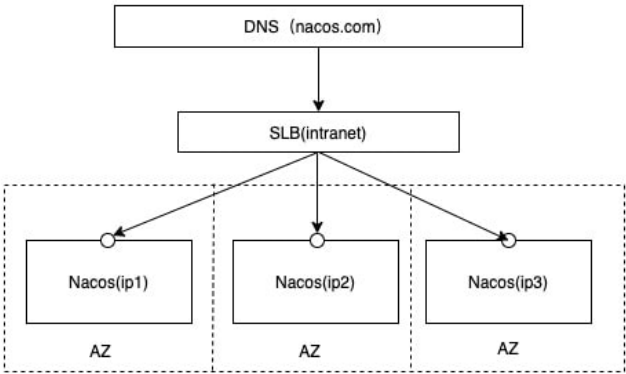
集群部署架构图

因此开源的时候推荐用户把所有服务列表放到一个vip下面，然后挂到一个域名下面

<http://ip1:port/openAPI> 直连ip模式，机器挂则需要修改ip才可以使用。

<http://SLB:port/openAPI> 挂载SLB模式(内网SLB，不可暴露到公网，以免带来安全风险)，直连SLB即可，下面挂server真实ip，可读性不好。

<http://nacos.com:port/openAPI> 域名 + SLB模式(内网SLB，不可暴露到公网，以免带来安全风险)，可读性好，而且换ip方便，推荐模式



端口	与主端口的偏移量	描述
8848	0	主端口，客户端、控制台及OpenAPI所使用的HTTP端口
9848	1000	客户端gRPC请求服务端端口，用于客户端向服务端发起连接和请求
9849	1001	服务端gRPC请求服务端端口，用于服务间同步等
7848	-1000	Jraft请求服务端端口，用于处理服务端间的Raft相关请求

使用VIP/nginx请求时，需要配置成TCP转发，不能配置http2转发，否则连接会被nginx断开。9849和7848端口为服务端之间的通信端口，请勿暴露到外部网络环境和客户端测。

1.2.3.2 部署案例

1.2.3.2.1 预备环境准备

请确保是在环境中安装使用：

- 64 bit OS Linux/Unix/Mac，推荐使用Linux系统。
- 64 bit JDK 1.8+；[下载](#) [配置](#)。
- Maven 3.2.x+；[下载](#) & [配置](#)。
- 3个或3个以上Nacos节点才能构成集群。

1.2.3.2.2 下载源码或者安装包

你可以通过两种方式来获取 Nacos。

从 Github 上下载源码方式

```
apt update && apt -y install git openjdk-8-jdk maven
git clone https://github.com/alibaba/nacos.git
unzip nacos-develop.zip
cd nacos-develop
mvn -Prelease-nacos clean install -U
cd distribution/target/nacos-server-2.3.0-SNAPSHOT/nacos/
```

下载编译后压缩包方式

```
apt update && apt -y install openjdk-8-jdk
apt update && apt -y install openjdk-11-jdk
wget https://github.com/alibaba/nacos/releases/download/2.2.3/nacos-server-2.2.3.tar.gz
tar xf nacos-server-2.2.3.tar.gz -C /usr/local/
```

范例：二进制安装

```
[root@ubuntu2204 ~]#apt update && apt -y install openjdk-11-jdk
[root@ubuntu2204 ~]#wget https://github.com/alibaba/nacos/releases/download/2.2.3/nacos-server-2.2.3.tar.gz
[root@ubuntu2204 ~]#tar xf nacos-server-2.2.3.tar.gz -C /usr/local/
```

1.2.3.2.3 确定数据源

- 如果使用内置数据源
无需进行任何配置
- 使用外置数据源
生产使用建议至少主备模式，或者采用高可用数据库。
初始化 MySQL 数据库
[sql语句源文件](#)

```
[root@ubuntu2204 ~]#apt update && apt -y install mysql-server
[root@ubuntu2204 ~]#sed -i '/127.0.0.1/s/^/#/' /etc/mysql/mysql.conf.d/mysqld.cnf
[root@ubuntu2204 ~]#systemctl restart mysql
[root@ubuntu2204 ~]#mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 8.0.33-0ubuntu0.22.04.4 (Ubuntu)

mysql> create user nacos@'10.0.0.%' identified with mysql_native_password by 'nacos';
mysql> create database nacos;
mysql> grant all on nacos.* to nacos@'10.0.0.%' ;

[root@node01 ~]#apt update && apt -y install mysql-client
[root@node01 ~]#mysql -unacos -pnacos -h10.0.0.200 nacos < /usr/local/nacos/conf/mysql-schema.sql
```

1.2.3.2.4 配置集群配置文件

在nacos的解压目录nacos的conf目录下，有配置文件cluster.conf，请每行配置成ip:port。（

在所有集群节点修改下面文件

```
[root@node1 ~]#vim /usr/local/nacos/conf/cluster.conf
# ip:port
10.0.0.201:8848
10.0.0.202:8848
10.0.0.203:8848
```

1.2.3.2.5 application.properties 配置

如果使用内置数据源无需修改配置

如果使用MySQL数据源，在所有集群节点上需要执行下面操作

```
[root@node1 ~]#vi /usr/local/nacos/conf/application.properties
##### Config Module Related Configurations #####
### If use MySQL as datasource:
### Deprecated configuration property, it is recommended to use `spring.sql.init.platform` replaced.
# spring.datasource.platform=mysql
spring.sql.init.platform=mysql #取消注释

### Count of DB:
db.num=1 #取消注释

### Connect URL of DB:
#修改下面行
db.url.0=jdbc:mysql://10.0.0.200:3306/nacos?
characterEncoding=utf8&connectTimeout=1000&socketTimeout=3000&autoReconnect=true&useUnicode=true&useSSL=false&serverTimezone=UTC
db.user.0=nacos
db.password.0=nacos

[root@node1 ~]#cd /usr/local/nacos/conf/
[root@node1 conf]#ls
1.4.0-ipv6_support-update.sql application.properties cluster.conf derby-schema.sql nacos-
logback.xml
announcement.conf application.properties.example cluster.conf.example mysql-schema.sql

#将配置同步到其它的节点
[root@node1 conf]#scp application.properties cluster.conf 10.0.0.202:/usr/local/nacos/conf
[root@node1 conf]#scp application.properties cluster.conf 10.0.0.203:/usr/local/nacos/conf
```

1.2.3.2.6 启动和关闭服务

Linux/Unix/Mac

集群模式启动

使用内置数据源

```
/usr/local/nacos/bin/startup.sh -p embedded
```

使用外置数据源

```
#在所有集群节点执行启动
[root@node1 ~]# /usr/local/nacos/bin/startup.sh

[root@node1 ~]#cat /usr/local/nacos/logs/start.out
/usr/lib/jvm/java-8-openjdk-amd64/bin/java -Djava.ext.dirs=/usr/lib/jvm/java-8-openjdk-
amd64/jre/lib/ext:/usr/lib/jvm/java-8-openjdk-amd64/lib/ext -server -Xms2g -Xmx2g -Xmn1g -
XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m -XX:-OmitStackTraceInFastThrow -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/usr/local/nacos/logs/java_heapdump.hprof -XX:-UseLargePages -Dnacos.member.list= -
Xloggc:/usr/local/nacos/logs/nacos_gc.log -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -
XX:+PrintGCTimeStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M -
Dloader.path=/usr/local/nacos/plugins,/usr/local/nacos/plugins/health,/usr/local/nacos/plugins/cmdb,/usr/local
/nacos/plugins/selector -Dnacos.home=/usr/local/nacos -jar /usr/local/nacos/target/nacos-server.jar --
spring.config.additional-location=file:/usr/local/nacos/conf/ --logging.config=/usr/local/nacos/conf/nacos-
logback.xml --server.max-http-header-size=524288

Nacos 2.2.3
Running in cluster mode, All function modules
Port: 8848
Pid: 82453
Console: http://10.0.0.201:8848/nacos/index.html
https://nacos.io

2023-06-03 22:32:23,223 INFO The server IP list of Nacos is [10.0.0.201:8848, 10.0.0.202:8848,
10.0.0.203:8848]

2023-06-03 22:32:24,228 INFO Nacos is starting...

2023-06-03 22:32:31,150 INFO Nacos started successfully in cluster mode. use external storage
```

```
[root@node2 ~]#./usr/local/nacos/bin/startup.sh
/usr/lib/jvm/java-8-openjdk-amd64/bin/java -Djava.ext.dirs=/usr/lib/jvm/java-8-openjdk-
amd64/jre/lib/ext:/usr/lib/jvm/java-8-openjdk-amd64/lib/ext -server -xms2g -Xmx2g -Xmn1g -
XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m -XX:-OmitStackTraceInFastThrow -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/usr/local/nacos/logs/java_heapdump.hprof -XX:-UseLargePages -Dnacos.member.list= -
Xloggc:/usr/local/nacos/logs/nacos_gc.log -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -
XX:+PrintGCTimeStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M -
Dloader.path=/usr/local/nacos/plugins,/usr/local/nacos/plugins/health,/usr/local/nacos/plugins/cmdb,/usr/local
/nacos/plugins/selector -Dnacos.home=/usr/local/nacos -jar /usr/local/nacos/target/nacos-server.jar --
spring.config.additional-location=file:/usr/local/nacos/conf/ --logging.config=/usr/local/nacos/conf/nacos-
logback.xml --server.max-http-header-size=524288
nacos is starting with cluster
nacos is starting, you can check the /usr/local/nacos/logs/start.out

[root@node2 ~]#tail -f /usr/local/nacos/logs/start.out
2023-06-03 21:57:55,128 INFO Nacos is starting...

2023-06-03 21:58:01,105 INFO Nacos started successfully in cluster mode. use external storage

[root@node3 ~]#./usr/local/nacos/bin/startup.sh
/usr/lib/jvm/java-8-openjdk-amd64/bin/java -Djava.ext.dirs=/usr/lib/jvm/java-8-openjdk-
amd64/jre/lib/ext:/usr/lib/jvm/java-8-openjdk-amd64/lib/ext -server -xms2g -Xmx2g -Xmn1g -
XX:MetaspaceSize=128m -XX:MaxMetaspaceSize=320m -XX:-OmitStackTraceInFastThrow -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/usr/local/nacos/logs/java_heapdump.hprof -XX:-UseLargePages -Dnacos.member.list= -
Xloggc:/usr/local/nacos/logs/nacos_gc.log -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -
XX:+PrintGCTimeStamps -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=100M -
Dloader.path=/usr/local/nacos/plugins,/usr/local/nacos/plugins/health,/usr/local/nacos/plugins/cmdb,/usr/local
/nacos/plugins/selector -Dnacos.home=/usr/local/nacos -jar /usr/local/nacos/target/nacos-server.jar --
spring.config.additional-location=file:/usr/local/nacos/conf/ --logging.config=/usr/local/nacos/conf/nacos-
logback.xml --server.max-http-header-size=524288
nacos is starting with cluster
nacos is starting, you can check the /usr/local/nacos/logs/start.out

[root@node3 ~]#tail -f /usr/local/nacos/logs/start.out
2023-06-03 21:57:58,660 INFO Nacos is starting...

2023-06-03 21:58:03,513 INFO Nacos started successfully in cluster mode. use external storage
```

关闭服务

Linux/Unix/Mac

```
[root@node1 ~]#./usr/local/nacos/bin/shutdown.sh
```

1.2.3.3.7 服务注册&发现和配置管理

服务注册

```
curl -X POST 'http://127.0.0.1:8848/nacos/v1/ns/instance?
serviceName=nacos.wang.serviceName&ip=1.2.3.4&port=8080'
```

注意：如果开启默认鉴权插件，需要在Header中带上用户名密码。

服务发现

```
curl -X GET 'http://127.0.0.1:8848/nacos/v1/ns/instance/list?serviceName=nacos.wang.serviceName'
```

注意：如果开启默认鉴权插件，需要在Header中带上用户名密码。

发布配置

```
curl -X POST "http://127.0.0.1:8848/nacos/v1/cs/configs?dataId=nacos.cfg.dataId&group=test&content=helloWorld"
```

注意：如果开启默认鉴权插件，需要在Header中带上用户名密码。

获取配置

```
curl -X GET "http://127.0.0.1:8848/nacos/v1/cs/configs?dataId=nacos.cfg.dataId&group=test"
```

注意：如果开启默认鉴权插件，需要在Header中带上用户名密码。

1.2.3.3.8 配置 haproxy 实现代理

```
#在新的服务器上配置代理
[root@ubuntu2204 ~]#apt update && apt install haproxy
```

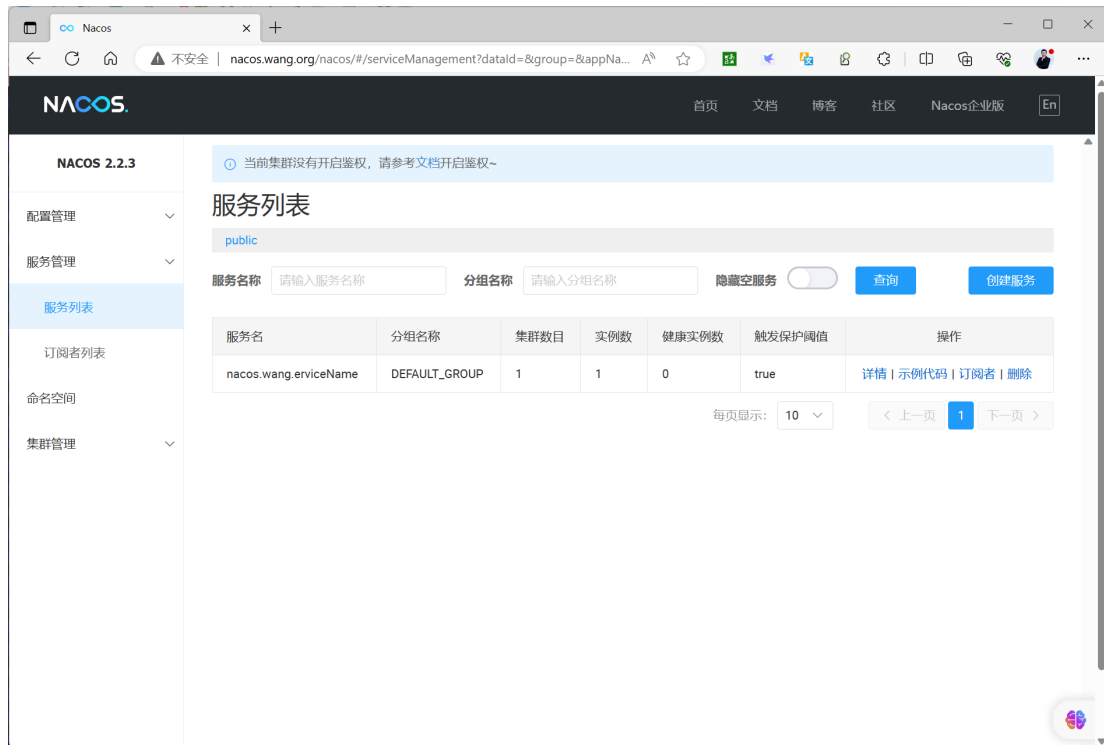
```
[root@ubuntu2204 ~]#vim /etc/haproxy/haproxy.cfg
#添加下面行
listen stats
mode http
```

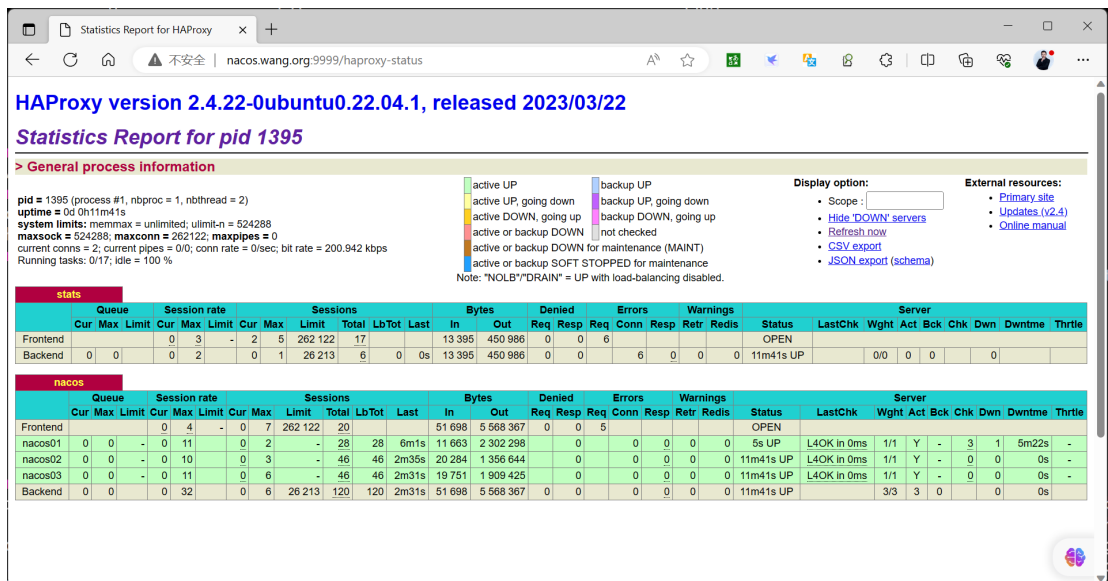
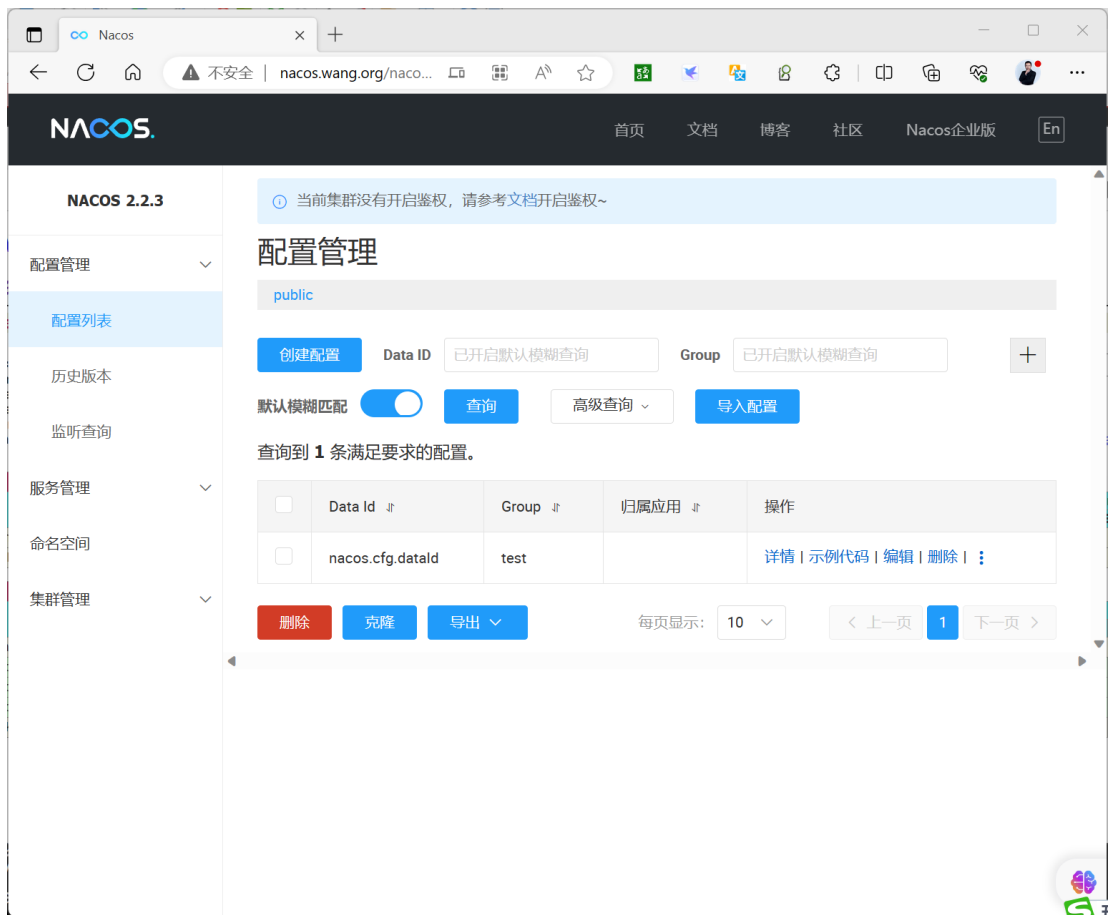
```
bind 0.0.0.0:9999
stats enable
log global
stats uri    /haproxy-status
stats auth   admin:123456

listen nacos
bind 0.0.0.0:80
server nacos01 10.0.0.201:8848 check
server nacos02 10.0.0.202:8848 check
server nacos03 10.0.0.203:8848 check
[root@ubuntu2204 ~]#systemctl reload haproxy
```

1.2.3.3.9 Web 页面访问

http://nacos.wang.org/nacos





1.2.4 基于 Docker 部署

操作步骤

- Clone 项目

```
git clone https://github.com/nacos-group/nacos-docker.git
cd nacos-docker
```

- 单机模式 Derby

```
docker-compose -f example/standalone-derby.yaml up
```

- 单机模式 MySQL

如果希望使用MySQL5.7

```
docker-compose -f example/standalone-mysql-5.7.yaml up
```

如果希望使用MySQL8

```
docker-compose -f example/standalone-mysql-8.yaml up
```

- 集群模式

```
docker-compose -f example/cluster-hostname.yaml up
```

- 服务注册

```
curl -X POST 'http://127.0.0.1:8848/nacos/v1/ns/instance?
serviceName=nacos.naming.serviceName&ip=20.18.7.10&port=8080'
```

- 服务发现

```
curl -X GET 'http://127.0.0.1:8848/nacos/v1/ns/instance/list?serviceName=nacos.naming.serviceName'
```

- 发布配置

```
curl -X POST "http://127.0.0.1:8848/nacos/v1/cs/configs?dataId=nacos.cfg.dataId&group=test&content=helloworld"
```

- 获取配置

```
curl -X GET "http://127.0.0.1:8848/nacos/v1/cs/configs?dataId=nacos.cfg.dataId&group=test"
```

- Nacos 控制台

link: <http://127.0.0.1:8848/nacos/>

Common property configuration

属性名称	描述	选项
MODE	系统启动方式: 集群/单机	cluster/standalone默认 cluster
NACOS_SERVERS	集群地址	p1:port1空槽p2:port2 空槽p3:port3
PREFER_HOST_MODE	支持ip还是域名模式	hostname/ip 默认 ip
NACOS_SERVER_PORT	Nacos运行端口	默认 8848
NACOS_SERVER_IP	多网卡模式下可以指定IP	
SPRING_DATASOURCE_PLATFORM	单机模式下支持MYSQL数据库	mysql / 空 默认空
MYSQL_SERVICE_HOST	数据库 连接地址	
MYSQL_SERVICE_PORT	数据库端口	默认: 3306
MYSQL_SERVICE_DB_NAME	数据库库名	
MYSQL_SERVICE_USER	数据库用户名	
MYSQL_SERVICE_PASSWORD	数据库用户密码	
MYSQL_SERVICE_DB_PARAM	数据库连接参数	default: characterEncoding=utf8&connectTimeout=1000&socketTimeout=3000&autoReconnect=true&useSSL=false
MYSQL_DATABASE_NUM	数据库编号	默认: 1
JVM_XMS	>Xms	默认: 1g
JVM_XMX	>Xmx	默认: 1g
JVM_XMN	>Xmn	默认: 512m
JVM_MS	>XX:MetaspaceSize	默认: 128m
JVM_MMS	>XX:MaxMetaspaceSize	默认: 320m
NACOS_DEBUG	是否开启远程DEBUG	y/n 默认: n
TOMCAT_ACCESSLOG_ENABLED	server.tomcat.accesslog.enabled	默认: false
NACOS_AUTH_SYSTEM_TYPE	权限系统类型选择,目前只支持nacos类型	默认: nacos
NACOS_AUTH_ENABLE	是否开启权限系统	默认: false
NACOS_AUTH_TOKEN_EXPIRE_SECONDS	token 失效时间	默认: 18000
NACOS_AUTH_TOKEN	token	默认: SecretKey0123456789012345678901234567890123456789012345678901234567890123456789
NACOS_AUTH_CACHE_ENABLE	权限缓存开关,开启后权限缓存的更新默认有15秒的延迟	默认: false
MEMBER_LIST	通过环境变量变量的方式设置集群地址	例子: 192.168.16.101:8847;raft_port=8807,192.168.16.101:raft_port=8808,192.168.16.101:8849;raft_port=8809
EMBEDDED_STORAGE	是否开启集群嵌入式存储模式	embedded 默认: none
NACOS_AUTH_CACHE_ENABLE	nacos.core.auth.caching.enabled	default: false
NACOS_AUTH_USER_AGENT_AUTH_WHITE_ENABLE	nacos.core.auth.enable.userAgentAuthWhite	default: false
NACOS_AUTH_IDENTITY_KEY	nacos.core.auth.server.identity.key	default: serverIdentity
NACOS_AUTH_IDENTITY_VALUE	nacos.core.auth.server.identity.value	default: security
NACOS_SECURITY_IGNORE_URLS	nacos.security.ignore.urls	default: /,/error,//**/.*.css,//**/.*.js,//**/.*.html,/**/*.map,//**/*.*.svg,//**/*.*.png,//**/*.*ico,/console-fe/public/**,/v1/auth/**,/v1/console/health/**,/actuator/**,/v1/console/server/**

1.2.5 基于 Kubernetes 部署

本项目包含一个可构建的Nacos Docker Image, 旨在利用StatefulSets在Kubernetes上部署Nacos

- 推荐使用[Nacos Operator](#)在Kubernetes部署Nacos Server

例子部署环境

- 机器配置

内网IP	主机名	配置
10.0.0.200	k8s-master	Ubuntu22.04 Single-core processor Mem 4G disk 40G
10.0.0.201	node01	Ubuntu22.04 Single-core processor Mem 4G disk 40G
10.0.0.202	node02	Ubuntu22.04 Single-core processor Mem 4G disk 40G

- Kubernetes
- NFS 版本: **4.1** 在k8s-master进行安装Server端, 并且指定共享目录, 本项目指定的/data/nfs-share
- Git

1.2.5.1 快速开始

- Clone 项目

```
git clone https://github.com/nacos-group/nacos-k8s.git
```

- 简单例子

如果你使用简单方式快速启动,请注意这是没有使用持久化卷的,可能存在数据丢失风险:

```
cd nacos-k8s
chmod +x quick-startup.sh
./quick-startup.sh
```

- 测试

- 服务注册

```
curl -X POST 'http://cluster-ip:8848/nacos/v1/ns/instance?
serviceName=nacos.naming.serviceName&ip=20.18.7.10&port=8080'
```

- 服务发现

```
curl -X GET 'http://cluster-ip:8848/nacos/v1/ns/instance/list?serviceName=nacos.naming.serviceName'
```

- 发布配置

```
curl -X POST "http://cluster-ip:8848/nacos/v1/cs/configs?
dataId=nacos.cfg.dataId&group=test&content=helloWorld"
```

- 获取配置

```
curl -X GET "http://cluster-ip:8848/nacos/v1/cs/configs?dataId=nacos.cfg.dataId&group=test"
```

1.2.1.2 高级使用

在高级使用中,Nacos在K8S拥有自动扩容缩容和数据持久特性,请注意如果需要这部分功能请使用PVC持久卷,Nacos的自动扩容缩容需要依赖持久卷,以及数据持久化也是一样,本例中使用的是NFS来使用PVC.

1.2.1.2.1 部署 NFS

- 创建角色

```
kubectl create -f deploy/nfs/rbac.yaml
```

如果的K8S命名空间不是default,请在部署RBAC之前执行以下脚本:

```
# Set the subject of the RBAC objects to the current namespace where the provisioner is being deployed
$ NS=$(kubectl config get-contexts|grep -e "\*" |awk '{print $5}')
$ NAMESPACE=${NS:-default}
$ sed -i 's/namespace:.*/namespace: $NAMESPACE/g' ./deploy/nfs/rbac.yaml
```

- 创建 ServiceAccount 和部署 NFS-Client Provisioner

```
kubectl create -f deploy/nfs/deployment.yaml
```

- 创建 NFS StorageClass

```
kubectl create -f deploy/nfs/class.yaml
```

- 验证NFS部署成功

```
kubectl get pod -l app=nfs-client-provisioner
```

1.2.1.2.2 部署数据库

```
cd nacos-k8s

kubectl create -f deploy/mysql/mysql-nfs.yaml
```

- 验证数据库是否正常工作

```
kubectl get pod
NAME                                READY   STATUS    RESTARTS   AGE
mysql-gf2vd                         1/1     Running   0           111m
```

执行数据库初始化语句

数据库初始化语句位置 <https://github.com/alibaba/nacos/blob/develop/distribution/conf/nacos-mysql.sql>

1.2.1.2.3 部署Nacos

- 修改 `deploy/nacos/nacos-pvc-nfs.yaml`

```
data:
  mysql.host: "数据库地址"
  mysql.db.name: "数据库名称"
  mysql.port: "端口"
  mysql.user: "用户名"
  mysql.password: "密码"
```

- 创建 Nacos

```
kubectl create -f nacos-k8s/deploy/nacos/nacos-pvc-nfs.yaml
```

- 验证Nacos节点启动成功

```
kubectl get pod -l app=nacos
```

NAME	READY	STATUS	RESTARTS	AGE
nacos-0	1/1	Running	0	19h
nacos-1	1/1	Running	0	19h
nacos-2	1/1	Running	0	19h

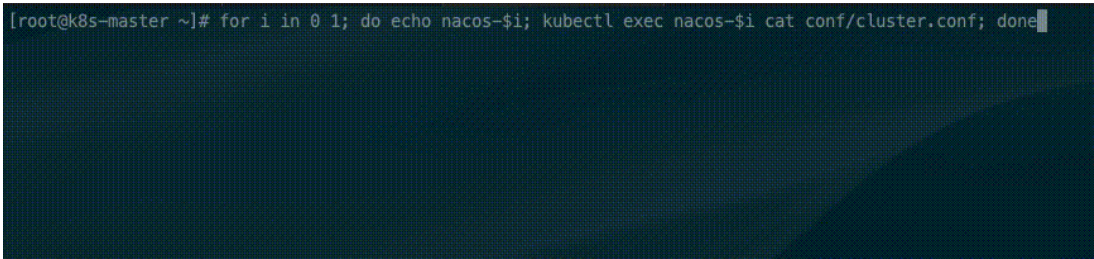
1.2.1.2.4 扩容测试

- 在扩容前，使用 `kubectl exec` 获取在pod中的Nacos集群配置文件信息

```
for i in 0 1; do echo nacos-$i; kubectl exec nacos-$i cat conf/cluster.conf; done
```

StatefulSet控制器根据其序号索引为每个Pod提供唯一的主机名。主机名采用 - 的形式。因为nacos StatefulSet的副本字段设置为2，所以当前集群文件中只有两个Nacos节点地址

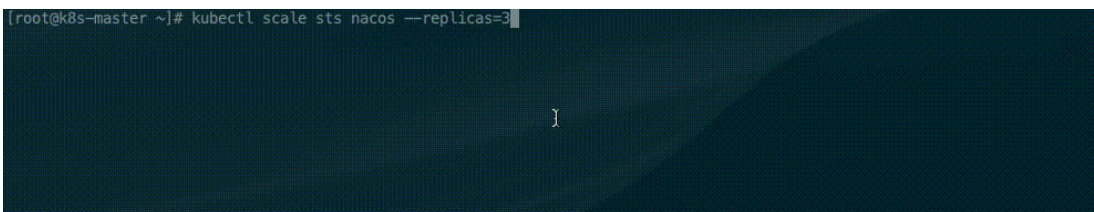
```
[root@k8s-master ~]# for i in 0 1; do echo nacos-$i; kubectl exec nacos-$i cat conf/cluster.conf; done
```



- 使用 `kubectl scale` 对Nacos动态扩容

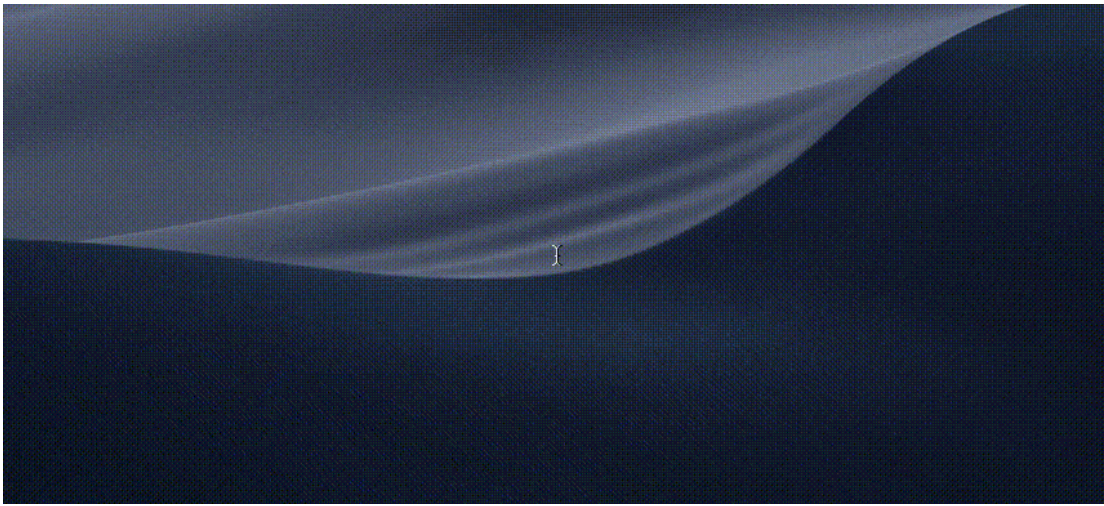
```
kubectl scale sts nacos --replicas=3
```

```
[root@k8s-master ~]# kubectl scale sts nacos --replicas=3
```



- 在扩容后，使用 `kubectl exec` 获取在pod中的Nacos集群配置文件信息

```
for i in 0 1 2; do echo nacos-$i; kubectl exec nacos-$i cat conf/cluster.conf; done
```

- 使用 `kubectl exec` 执行Nacos API 在每台节点上获取当前Leader是否一致

```
for i in 0 1 2; do echo nacos-$i; kubectl exec nacos-$i curl -X GET "http://localhost:8848/nacos/v1/ns/raft/state"; done
```

到这里你可以发现新节点已经正常加入Nacos集群当中

1.2.1.2.5 持久化

必须要使用持久卷，否则会出现数据丢失的情况

项目目录

目录	描述
plugin	帮助Nacos集群进行动态扩容的插件Docker镜像源码
deploy	K8s 部署文件

配置属性

- nacos-pvc-nfs.yaml or nacos-quick-start.yaml

名称	必要	描述
mysql.host	Y	自建数据库地址,使用外部数据库时必须指定
mysql.db.name	Y	数据库名称
mysql.port	N	数据库端口
mysql.user	Y	数据库用户名(请不要含有符号,)
mysql.password	Y	数据库密码(请不要含有符号,)
SPRING_DATASOURCE_PLATFORM	Y	数据库类型,默认为embedded嵌入式数据库,参数只支持mysql或embedded
NACOS_REPLICAS	N	确定执行Nacos启动节点数量,如果不适用动态扩容插件,就必须配置这个属性, 否则使用扩容插件后不会生效
NACOS_SERVER_PORT	N	Nacos 端口 为peer_finder插件提供端口
NACOS_APPLICATION_PORT	N	Nacos 端口
PREFER_HOST_MODE	Y	启动Nacos集群按域名解析

- nfs deployment.yaml

名称	必要	描述
NFS_SERVER	Y	NFS 服务端地址
NFS_PATH	Y	NFS 共享目录
server	Y	NFS 服务端地址
path	Y	NFS 共享目录

- mysql

名称	必要	描述
MYSQL_ROOT_PASSWORD	N	ROOT 密码
MYSQL_DATABASE	Y	数据库名称
MYSQL_USER	Y	数据库用户名
MYSQL_PASSWORD	Y	数据库密码
MYSQL_REPLICATION_USER	Y	数据库复制用户
MYSQL_REPLICATION_PASSWORD	Y	数据库复制用户密码
Nfs:server	N	NFS 服务端地址，如果使用本地部署不需要配置
Nfs:path	N	NFS 共享目录，如果使用本地部署不需要配置