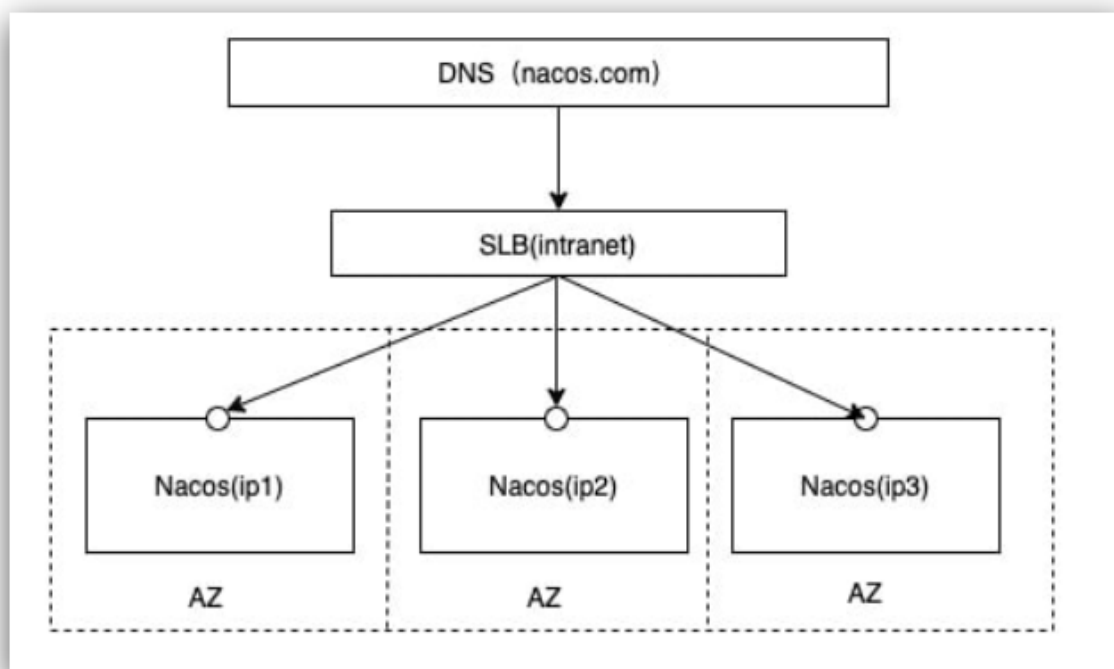


# Nacos集群搭建

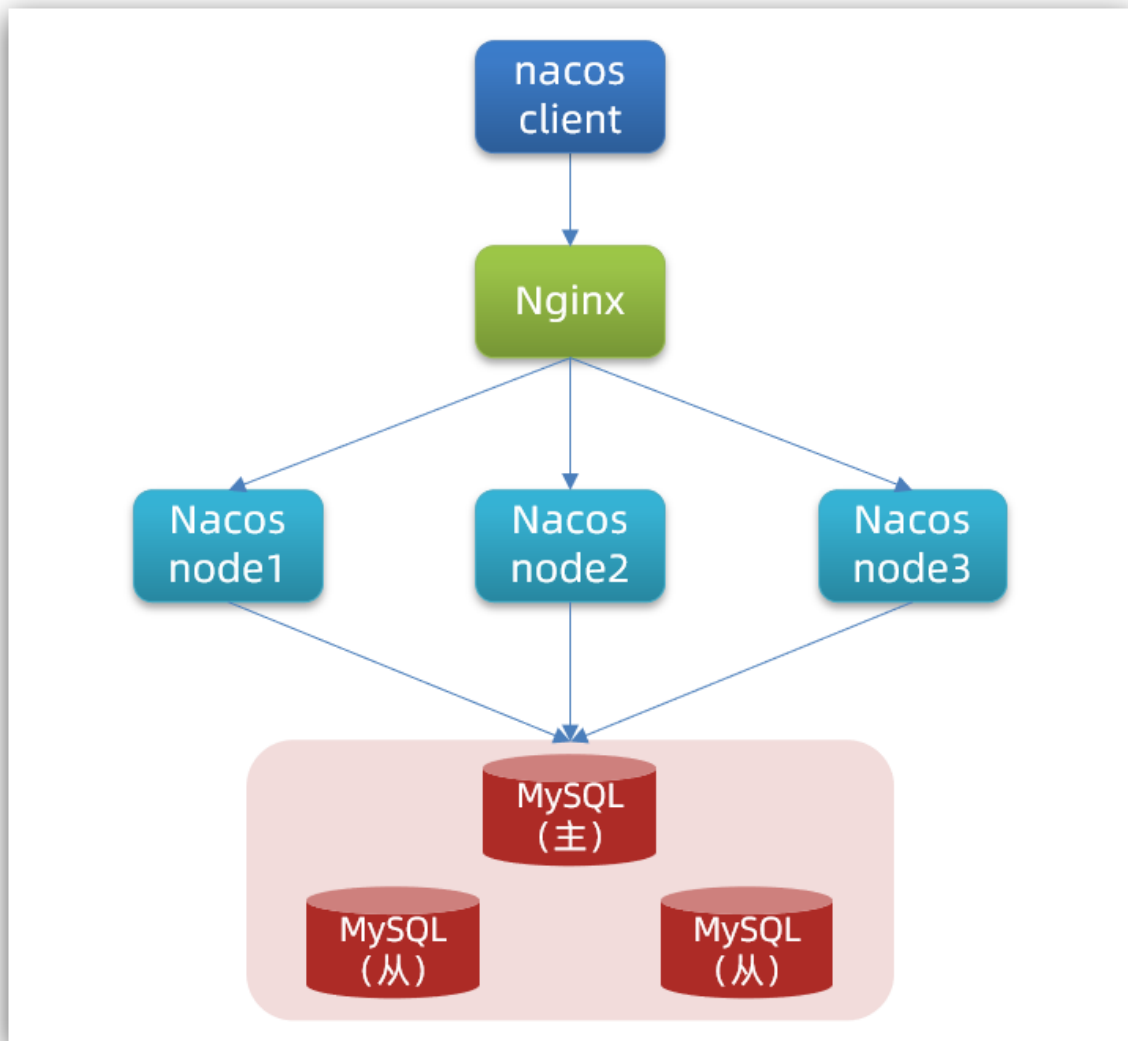
## 1.集群结构图

官方给出的Nacos集群图：



其中包含3个nacos节点，然后一个负载均衡器代理3个Nacos。这里负载均衡器可以使用nginx。

我们计划的集群结构：



三个nacos节点的地址：

节点	ip	port
nacos1	192.168.150.1	8845
nacos2	192.168.150.1	8846
nacos3	192.168.150.1	8847

## 2.搭建集群

搭建集群的基本步骤：

- 搭建数据库，初始化数据库表结构
- 下载nacos安装包
- 配置nacos
- 启动nacos集群
- nginx反向代理

## 2.1.初始化数据库

Nacos默认数据存储在内置数据库Derby中，不属于生产可用的数据库。

官方推荐的最佳实践是使用带有主从的高可用数据库集群，主从模式的高可用数据库可以参考[传智教育](#)的后续高手课程。

这里我们以单点的数据库为例来讲解。

首先新建一个数据库，命名为nacos，而后导入下面的SQL：

```
CREATE TABLE `config_info` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',  
  `data_id` varchar(255) NOT NULL COMMENT 'data_id',  
  `group_id` varchar(255) DEFAULT NULL,  
  `content` longtext NOT NULL COMMENT 'content',  
  `md5` varchar(32) DEFAULT NULL COMMENT 'md5',  
  `gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',  
  `src_user` text COMMENT 'source user',  
  `src_ip` varchar(50) DEFAULT NULL COMMENT 'source ip',  
  `app_name` varchar(128) DEFAULT NULL,  
  `tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',  
  `c_desc` varchar(256) DEFAULT NULL,  
  `c_use` varchar(64) DEFAULT NULL,  
  `effect` varchar(64) DEFAULT NULL,  
  `type` varchar(64) DEFAULT NULL,  
  `c_schema` text,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_configinfo_datagrouptenant` (`data_id`,`group_id`,`tenant_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info';  
  
/*****  
/* 数据库全名 = nacos_config */  
/* 表名称 = config_info_aggr */  
*****/  
CREATE TABLE `config_info_aggr` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',  
  `data_id` varchar(255) NOT NULL COMMENT 'data_id',  
  `group_id` varchar(255) NOT NULL COMMENT 'group_id',  
  `datum_id` varchar(255) NOT NULL COMMENT 'datum_id',  
  `content` longtext NOT NULL COMMENT '内容',  
  `gmt_modified` datetime NOT NULL COMMENT '修改时间',  
  `app_name` varchar(128) DEFAULT NULL,  
  `tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_configinfoaggr_datagrouptenantdatum`  
  (`data_id`,`group_id`,`tenant_id`,`datum_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='增加租户字段';  
  
/*****  
/* 数据库全名 = nacos_config */  
/* 表名称 = config_info_beta */  
*****/  
CREATE TABLE `config_info_beta` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',  
  `data_id` varchar(255) NOT NULL COMMENT 'data_id',  
  `group_id` varchar(255) NOT NULL COMMENT 'group_id',  
  `datum_id` varchar(255) NOT NULL COMMENT 'datum_id',  
  `content` longtext NOT NULL COMMENT '内容',  
  `gmt_modified` datetime NOT NULL COMMENT '修改时间',  
  `app_name` varchar(128) DEFAULT NULL,  
  `tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_configinfoaggr_datagrouptenantdatum`  
  (`data_id`,`group_id`,`tenant_id`,`datum_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='增加租户字段';
```

```

`id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
`data_id` varchar(255) NOT NULL COMMENT 'data_id',
`group_id` varchar(128) NOT NULL COMMENT 'group_id',
`app_name` varchar(128) DEFAULT NULL COMMENT 'app_name',
`content` longtext NOT NULL COMMENT 'content',
`beta_ips` varchar(1024) DEFAULT NULL COMMENT 'betaIps',
`md5` varchar(32) DEFAULT NULL COMMENT 'md5',
`gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
`gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
`src_user` text COMMENT 'source user',
`src_ip` varchar(50) DEFAULT NULL COMMENT 'source ip',
`tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',
PRIMARY KEY (`id`),
UNIQUE KEY `uk_configinfo_beta_data_group_tenant`
(`data_id`,`group_id`,`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin
COMMENT='config_info_beta';

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = config_info_tag */
/*****/
CREATE TABLE `config_info_tag` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `data_id` varchar(255) NOT NULL COMMENT 'data_id',
  `group_id` varchar(128) NOT NULL COMMENT 'group_id',
  `tenant_id` varchar(128) DEFAULT '' COMMENT 'tenant_id',
  `tag_id` varchar(128) NOT NULL COMMENT 'tag_id',
  `app_name` varchar(128) DEFAULT NULL COMMENT 'app_name',
  `content` longtext NOT NULL COMMENT 'content',
  `md5` varchar(32) DEFAULT NULL COMMENT 'md5',
  `gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
  `src_user` text COMMENT 'source user',
  `src_ip` varchar(50) DEFAULT NULL COMMENT 'source ip',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_configinfo_tag_data_group_tenant_tag`
(`data_id`,`group_id`,`tenant_id`,`tag_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='config_info_tag';

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = config_tags_relation */
/*****/
CREATE TABLE `config_tags_relation` (
  `id` bigint(20) NOT NULL COMMENT 'id',
  `tag_name` varchar(128) NOT NULL COMMENT 'tag_name',
  `tag_type` varchar(64) DEFAULT NULL COMMENT 'tag_type',
  `data_id` varchar(255) NOT NULL COMMENT 'data_id',
  `group_id` varchar(128) NOT NULL COMMENT 'group_id',
  `tenant_id` varchar(128) DEFAULT '' COMMENT 'tenant_id',
  `nid` bigint(20) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`nid`),
  UNIQUE KEY `uk_configtagrelation_configidtag` (`id`,`tag_name`,`tag_type`),
  KEY `idx_tenant_id` (`tenant_id`)

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin
COMMENT='config_tag_relation';

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = group_capacity */
/*****/
CREATE TABLE `group_capacity` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键ID',
  `group_id` varchar(128) NOT NULL DEFAULT '' COMMENT 'Group ID, 空字符表示整个集群',
  `quota` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '配额, 0表示使用默认值',
  `usage` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '使用量',
  `max_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个配置大小上限, 单位为字节, 0表示使用默认值',
  `max_aggr_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '聚合子配置最大个数, , 0表示使用默认值',
  `max_aggr_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个聚合数据的子配置大小上限, 单位为字节, 0表示使用默认值',
  `max_history_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '最大变更历史数量',
  `gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_group_id` (`group_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='集群、各Group容量信息表';

/*****/
/* 数据库全名 = nacos_config */
/* 表名称 = his_config_info */
/*****/
CREATE TABLE `his_config_info` (
  `id` bigint(64) unsigned NOT NULL,
  `nid` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `data_id` varchar(255) NOT NULL,
  `group_id` varchar(128) NOT NULL,
  `app_name` varchar(128) DEFAULT NULL COMMENT 'app_name',
  `content` longtext NOT NULL,
  `md5` varchar(32) DEFAULT NULL,
  `gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `src_user` text,
  `src_ip` varchar(50) DEFAULT NULL,
  `op_type` char(10) DEFAULT NULL,
  `tenant_id` varchar(128) DEFAULT '' COMMENT '租户字段',
  PRIMARY KEY (`nid`),
  KEY `idx_gmt_create` (`gmt_create`),
  KEY `idx_gmt_modified` (`gmt_modified`),
  KEY `idx_did` (`data_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='多租户改造';

/*****/
/* 数据库全名 = nacos_config */

```

```

/* 表名称 = tenant_capacity */
/*****/
CREATE TABLE `tenant_capacity` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT COMMENT '主键ID',
  `tenant_id` varchar(128) NOT NULL DEFAULT '' COMMENT 'Tenant ID',
  `quota` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '配额, 0表示使用默认值',
  `usage` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '使用量',
  `max_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个配置大小上限, 单位为字节, 0表示使用默认值',
  `max_aggr_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '聚合子配置最大个数',
  `max_aggr_size` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '单个聚合数据的子配置大小上限, 单位为字节, 0表示使用默认值',
  `max_history_count` int(10) unsigned NOT NULL DEFAULT '0' COMMENT '最大变更历史数量',
  `gmt_create` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `gmt_modified` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '修改时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_tenant_id` (`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='租户容量信息表';

CREATE TABLE `tenant_info` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT 'id',
  `kp` varchar(128) NOT NULL COMMENT 'kp',
  `tenant_id` varchar(128) default '' COMMENT 'tenant_id',
  `tenant_name` varchar(128) default '' COMMENT 'tenant_name',
  `tenant_desc` varchar(256) DEFAULT NULL COMMENT 'tenant_desc',
  `create_source` varchar(32) DEFAULT NULL COMMENT 'create_source',
  `gmt_create` bigint(20) NOT NULL COMMENT '创建时间',
  `gmt_modified` bigint(20) NOT NULL COMMENT '修改时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_tenant_info_kptenantid` (`kp`, `tenant_id`),
  KEY `idx_tenant_id` (`tenant_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin COMMENT='tenant_info';

CREATE TABLE `users` (
  `username` varchar(50) NOT NULL PRIMARY KEY,
  `password` varchar(500) NOT NULL,
  `enabled` boolean NOT NULL
);

CREATE TABLE `roles` (
  `username` varchar(50) NOT NULL,
  `role` varchar(50) NOT NULL,
  UNIQUE INDEX `idx_user_role` (`username` ASC, `role` ASC) USING BTREE
);

CREATE TABLE `permissions` (
  `role` varchar(50) NOT NULL,
  `resource` varchar(255) NOT NULL,
  `action` varchar(8) NOT NULL,
  UNIQUE INDEX `uk_role_permission` (`role`, `resource`, `action`) USING BTREE
);

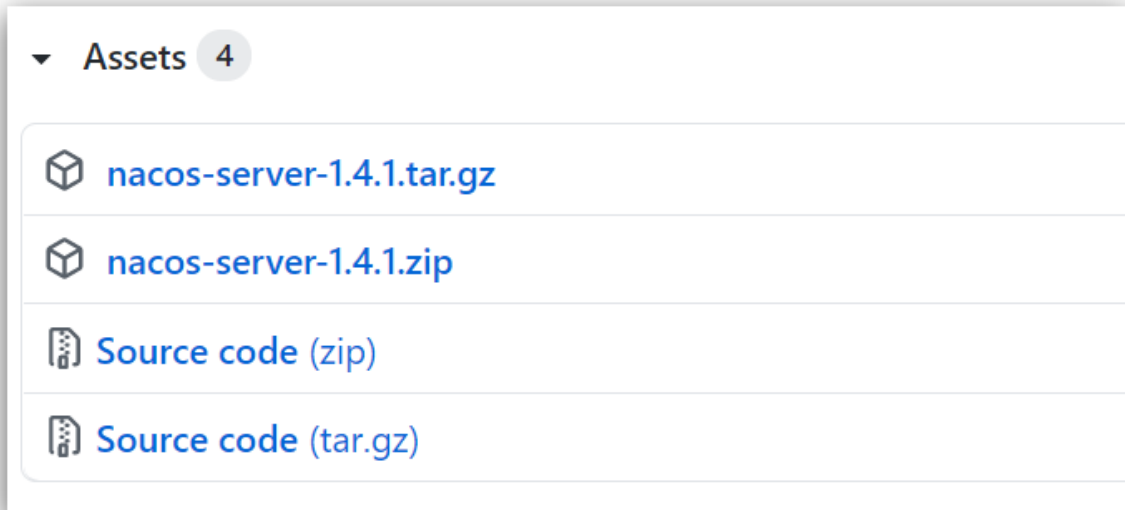
```

```
INSERT INTO users (username, password, enabled) VALUES ('nacos',  
'$2a$10$EuWPZHzz32dJN7jexM34MOeYi rDdFAZm2kuwj7VEOJhhZkDrxfvUu', TRUE);  
  
INSERT INTO roles (username, role) VALUES ('nacos', 'ROLE_ADMIN');
```

## 2.2. 下载nacos

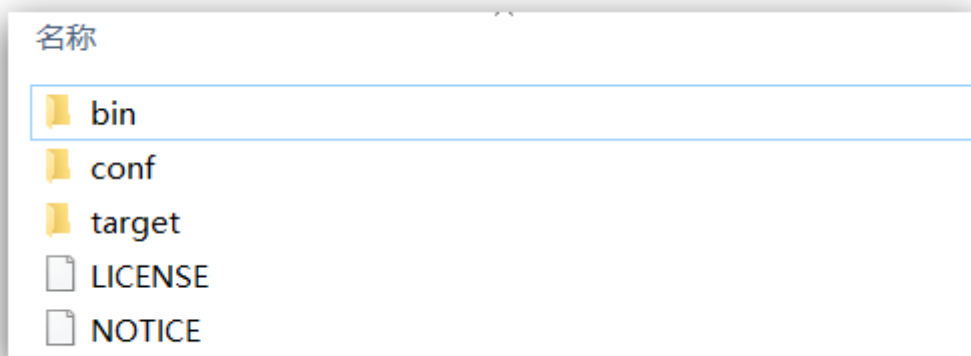
nacos在GitHub上有下载地址: <https://github.com/alibaba/nacos/tags>, 可以选择任意版本下载。

本例中才用1.4.1版本:



## 2.3. 配置Nacos

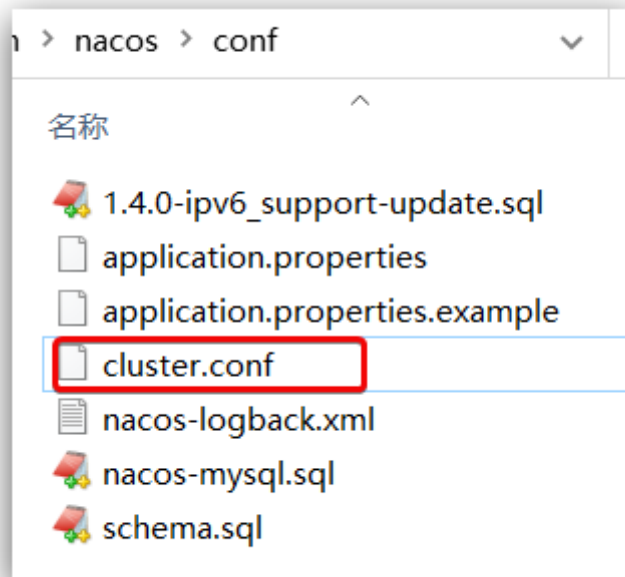
将这个包解压到任意非中文目录下, 如图:



目录说明:

- bin: 启动脚本
- conf: 配置文件

进入nacos的conf目录，修改配置文件cluster.conf.example，重命名为cluster.conf：



然后添加内容：

```
127.0.0.1:8845  
127.0.0.1:8846  
127.0.0.1:8847
```

然后修改application.properties文件，添加数据库配置

```
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  
&useUnicode=true&useSSL=false&serverTimezone=UTC  
db.user.0=root  
db.password.0=123
```

## 2.4.启动

将nacos文件夹复制三份，分别命名为：nacos1、nacos2、nacos3

```
📁 nacos1  
📁 nacos2  
📁 nacos3
```



然后分别修改三个文件夹中的application.properties,

nacos1:

```
server.port=8845
```

nacos2:

```
server.port=8846
```

nacos3:

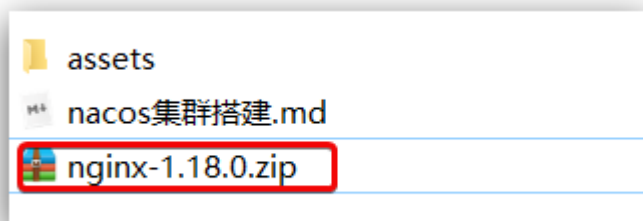
```
server.port=8847
```

然后分别启动三个nacos节点:

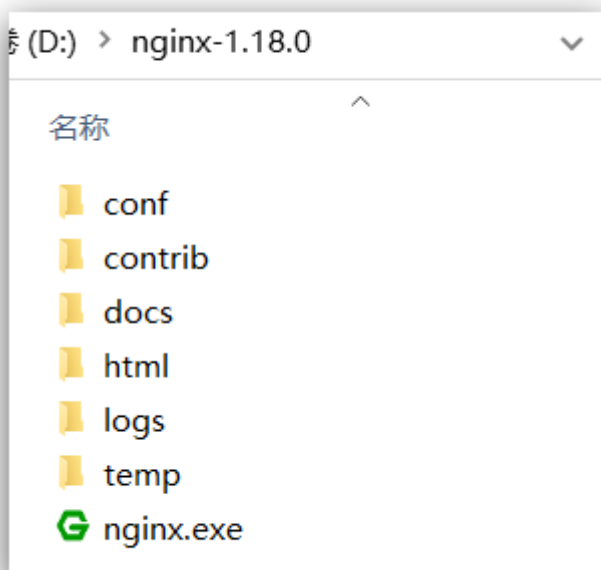
```
startup.cmd
```

## 2.5.nginx反向代理

找到课前资料提供的nginx安装包:



解压到任意非中文目录下:



修改conf/nginx.conf文件，配置如下：

```
upstream nacos-cluster {  
    server 127.0.0.1:8845;  
    server 127.0.0.1:8846;  
    server 127.0.0.1:8847;  
}  
  
server {  
    listen      80;  
    server_name localhost;  
  
    location /nacos {  
        proxy_pass http://nacos-cluster;  
    }  
}
```

而后在浏览器访问：<http://localhost/nacos>即可。

代码中application.yml文件配置如下：

```
spring:  
  cloud:  
    nacos:  
      server-addr: localhost:80 # Nacos地址
```

## 2.6.优化

---

- 实际部署时，需要给做反向代理的nginx服务器设置一个域名，这样后续如果有服务器迁移nacos的客户端也无需更改配置。
- Nacos的各个节点应该部署到多个不同服务器，做好容灾和隔离