

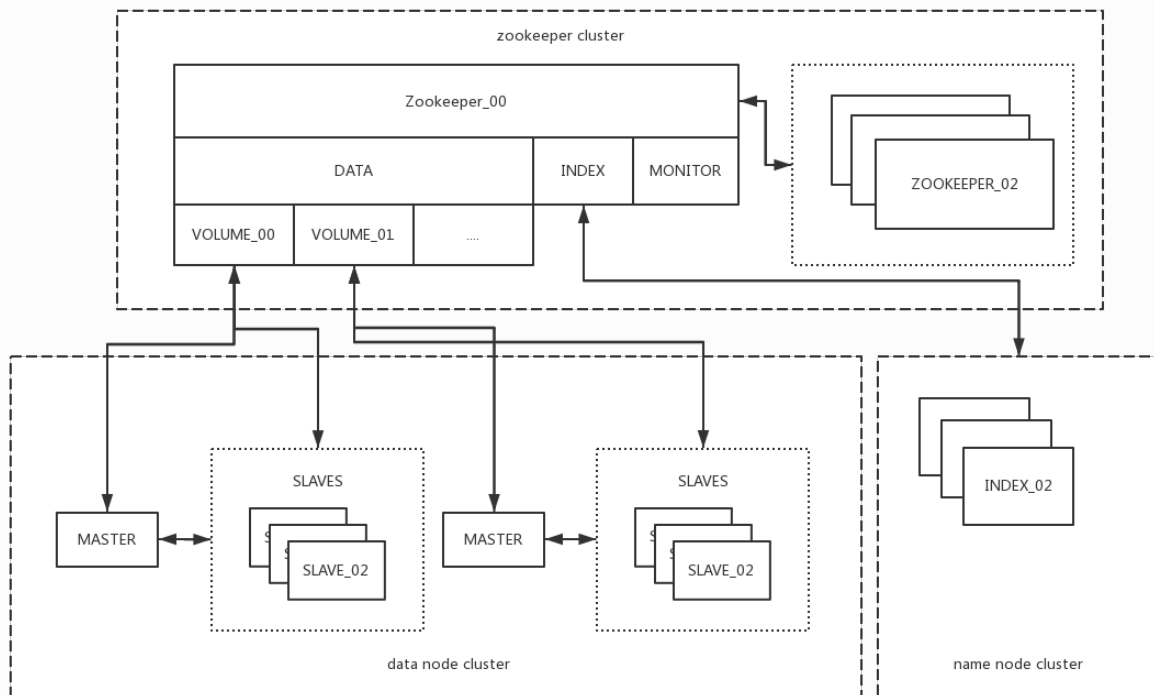
# NDFS (NERO DISTRIBUTED FILE SYSTEM)



用户手册

CENOCLOUD 分布式基础中间件技术组

# OVERVIEW



## 关于配置

NDFS提供了简单明了的配置方式 config.properties

```
1 # 文件块列表文件
2 chunk.ListFilePath=/Users/neroyang/project/iotcloud/middleware/ndfs/test_data/
  header.list
3 # 文件块存放路径
4 chunk.FilePath=/Users/neroyang/project/iotcloud/middleware/ndfs/test_data/
5 # 数据节点启动监听端口
6 server.host=localhost
7 server.port=1080
8 # 注册中心地址，若有多个，使用分号分隔
9 zookeeper.host=www.cenocloud.com:2181
10 # 注册中心超时时间
11 zookeeper.timeout=10000
12 # 索引节点负载均衡策略 默认轮询
13 load-balance.type=polling
14 # 节点名称
15 node.name=data_01
16 # 节点备注
```

```
17 node.note=data node 01
18 # 节点虚拟卷/命名空间
19 node.volume=DATA_01
20 # 节点类型 Master 主, Slave 备
21 node.type=Master
22 # 若节点类型为Master, 该项配置需要注释, 若节点类型为salve, 则需要填写需要备份的
   节点名称
23 # node.master = null
24 # 数据防篡改, 打开意味着所有对数据的修改和删除操作将失效, 并且会产生文件hash
   链, 进行防篡改校验, 默认关闭
25 data.tamper-proof.enable=false
26 # 最大允许上传文件
27 data.max-upload-size=5
28 # 文件块大小, 单位为兆字节, 默认64M, 参考 64/128/256/... 建议不做修改
29 data.chunk-size=64
30 # 节点间操作校验密钥, 若配置就意味着与该节点通信的所有节点均需配置, 包括
   namenode, client需带着密钥访问
31 auth.enable=false
32 auth.key=2186394109e2427b80710fb561485184
33 auth.secret=2186394109e2427b80710fb5614851842186394109e2427b80710fb56148
   5184
```

## 配置解读

### 文件块列表文件

文件块列表文件是

## 关于部署

NDFS提供了简单易用的启动脚本

索引节点 ndfsIndexServer.sh

```
1 #!/usr/bin/env bash
```

数据节点 ndfsDataServer.sh

```
1 #!/usr/bin/env bash
```