IPFS测试文档

IPFS服务器

校内

```
# IP
222.20.126.154
# Port
node0-node3:5000-5002
```

校外

```
# IP
101.200.162.128
# Port
node0-node3:5000-5002
```

测试过程

- 使用Docker, 多虚拟机或多主机搭建有三个节点的IPFS私有网络
- 使用IPFS命令行工具, 在一个节点上传文件，并在另一个节点下载文件
- 改进IPFS Java客户端代码, 实现上传和下载文件的HTTP接口, 每次调用接口不再重新连接IPFS服务器
- 为每个节点分别部署IPFS Java客户端, 在一个节点上传文件，并在另一个节点下载文件

IPFS私有网络搭建

详细见：

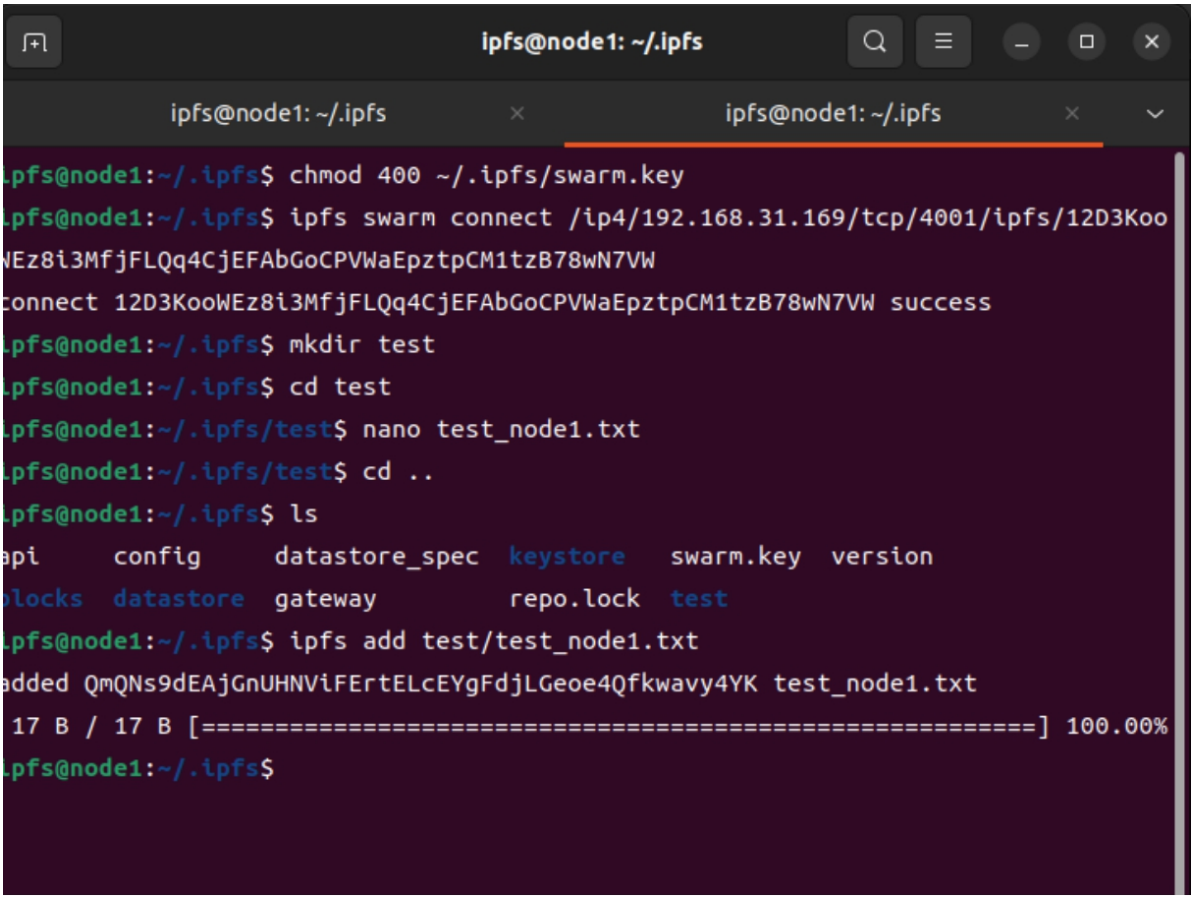
https://github.com/mapkkkk/ipfs/blob/master/doc/private_ipfs_cluster_build.md

不过有一个问题：docker-compose部署的cluster没办法配置策略（只能从localhost发出http请求），所以需要打洞出去（先代理进本机，再端口转发）

docker-compose的config见git repo的config：

https://github.com/mapkkkk/ipfs/blob/master/config/docker-compose.yml

使用IPFS命令行工具上传文件截图及描述



创建了一个test_node1.txt，使用ipfs add上传成功，CID为QmQNs9dEAjGnUHNViFErtELcEYgFdjLGeoe4Qfkwavy4YK

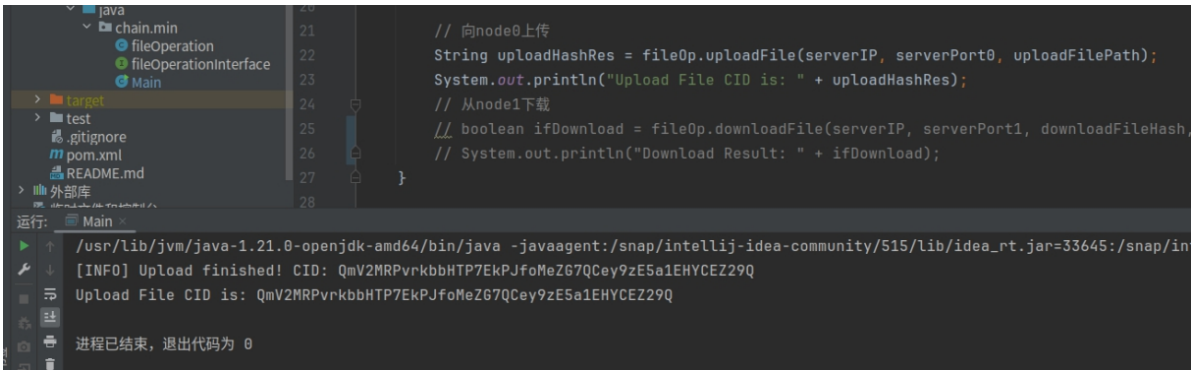
使用IPFS命令行工具下载文件截图及描述

```
ipfs@node1:~/.ipfs/download$ ipfs get QmQNs9dEAjGnUHNViFErtELcEYgFdjLGeoe4Qfkwav
y4YK
Saving file(s) to QmQNs9dEAjGnUHNViFErtELcEYgFdjLGeoe4Qfkwavy4YK
 17 B / 17 B [=============================================================] 100.00% 0s
ipfs@node1:~/.ipfs/download$ ls
QmQNs9dEAjGnUHNViFErtELcEYgFdjLGeoe4Qfkwavy4YK
ipfs@node1:~/.ipfs/download$ cat QmQNs9dEAjGnUHNViFErtELcEYgFdjLGeoe4Qfkwavy4YK
test_node1
12345
ipfs@node1:~/.ipfs/download$
```

下载CID为QmQNs9dEAjGnUHNViFErtELcEYgFdjLGeoe4Qfkwavy4YK的文件，cat之后确实是刚才上传的文件

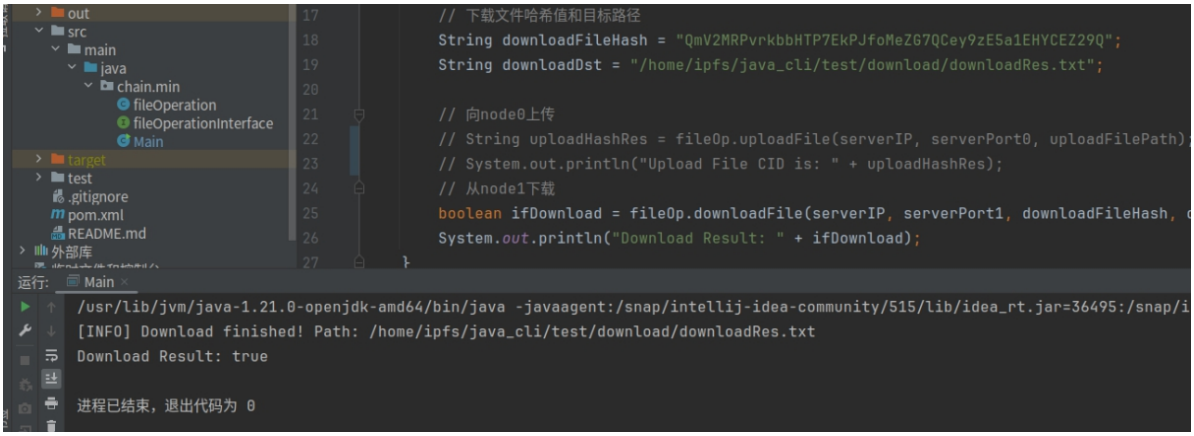
```
ipfs@node0:~/ipfs-cluster-ctl$ ./ipfs-cluster-ctl status QmQiXsdDuqrikmahhYxmigi
2a5uFtBWGArbCKWXGgDJvSB
QmQiXsdDuqrikmahhYxmigi2a5uFtBWGArbCKWXGgDJvSB:
    > cluster0            : UNPINNED | 2024-06-30T00:17:53.316177611Z | Attempt
s: 0 | Priority: false
    > cluster1            : UNPINNED | 2024-06-30T00:17:53.316177611Z | Attempt
s: 0 | Priority: false
    > cluster2            : UNPINNED | 2024-06-30T00:17:53.316177611Z | Attempt
s: 0 | Priority: false
ipfs@node0:~/ipfs-cluster-ctl$
```

使用IPFS Java客户端上传文件截图及描述



node0映射到5000端口，上传测试文件后CID为：
QmV2MRPvrkbbHTP7EkPJfoMeZG7QCey9zE5a1EHYCEZ29Q

使用IPFS Java客户端下载文件截图及描述



从node1下载刚才向node0上传的文件，成功查找到CID，下载成功

可见内容确实是之前上传的文件内容