

快速部署JDChain

由 [赵光伟](#) 创建, 最后修改于大约1分钟以前

目前部署JDChain需要修改三个配置文件，要花费一定的时间，为了快速部署JDChain，特此开发了这个小工具。

简要步骤

- 1) 将安装的zip包解压到centos指定路径下，比如：`/export/jdchain/peer0`；`/export/jdchain/peer1`；`/export/jdchain/peer2`；`/export/jdchain/peer3`
- 2) (此步可以省略，使用配置文件中的默认公私钥)每个peer都生成公私钥。进入四个peer节点中的bin路径，分别执行如下命令：

`peer0: sh keygen.sh -n a.com`
`peer1: sh keygen.sh -n b.com`
`peer2: sh keygen.sh -n c.com`
`peer3: sh keygen.sh -n d.com`
- 3) 安装ump管理工具，每个peer节点对应一个配置文件。参考样例下载地址：<http://box.jd.com/sharedInfo/D81333A4DC57FAA5>
其中0application.properties对应peer0的配置，其它类似。主要修改其中的安装路径。比如修改完0application.properties配置文件后，其它的配置文件可以复制。唯一的不同的是每个配置文件的：`peer.peerName=a.com` 值要根据实际填写。

0application.properites

折叠源码

```
server.port=9000

logging.level.root = WARN
logging.level.org.springframework.web = DEBUG
logging.file=./logs/jdchain-deploy-peer0.log
#logging.path=/var/log

#setup the params; must set; it's also used in the dev mode;
peer.peerName=a.com
#generate the public/private keys by the peer.name;
peer.basePath=/Users/zhaoguangwei/Documents/jdchain/peer0
#peer.basePath=/export/11-jdchain/peer0
peer.host=127.0.0.1
peer.initPort=8800
peer.consensusPort=36000
peer.visitPort=17800
peer.serverPort=9000
#when start up, invoke the gateway to push the params related;
#gateway.url=192.168.101.47:9000/gw
peer.mediatorUrl=localhost:9000/deploy
# if true, then this peer's private key will send to the gateway;
peer.gatewayBindPeer=true
schema.retrieval.url=http://192.168.151.39:8082
task.retrieval.url=http://192.168.151.39:10005

ump.dbUri=F:\\tmp\\jdchainDeploy\\peer0\\conf\\ump.db
ump.dbPasswd=

#register all Peer by single machine; dev model;
dev.peer0.peerId=0
dev.peer0.peerName=a.com
dev.peer0.basePath=/Users/zhaoguangwei/Documents/jdchain/peer0
dev.peer0.host=127.0.0.1
dev.peer0.initPort=8800
dev.peer0.consensusPort=36000
dev.peer0.visitPort=17800
dev.peer0.serverPort=9000
dev.peer0.mediatorUrl=localhost:9000/deploy
dev.peer0.gatewayBindPeer=true
dev.peer0.peerPubKey=3snPdw7i7PbUX47tXkHVuRd2mP8mXhEJbaBM441h9wa67TAeMfiqUX
dev.peer0.peerPrivKey=177gjuzAyyf8W2KYST8tVPmvGBsPLhXsf55HpHxSbHF7Va995ekXvxjNmEYNt5wP6GxTpW
dev.peer0.peerPasswd=DYu3G8aGtMBW1WrTw76zxQJQU4DHLw9MLyy7peG4LKkY
#peer1
dev.peer1.peerId=1
dev.peer1.peerName=b.com
dev.peer1.basePath=/Users/zhaoguangwei/Documents/jdchain/peer1
dev.peer1.host=127.0.0.1
dev.peer1.initPort=8810
dev.peer1.consensusPort=36010
dev.peer1.visitPort=17810
dev.peer1.serverPort=9010
dev.peer1.mediatorUrl=localhost:9000/deploy
dev.peer1.gatewayBindPpeer=true
dev.peer1.peerPubKey=3snPdw7i7PdBTU3FEHZa59uWG4xSsDx4MGn2E2vAbwfNFUbpDRc2Am
```

```
dev.peer1.peerPrivKey=177gjstrJDHmZabZTxybLMaH5n24DBPueigsQDx8jc2erw7cAu1Xy2uX3yJut6xgPbPV8mx
dev.peer1.peerPasswd=DYu3G8aGTMBW1WrTw76zxQJQU4DHLw9MLyy7peG4LKkY
#peer2
dev.peer2.peerId=2
dev.peer2.peerName=c.com
dev.peer2.basePath=/Users/zhaoguangwei/Documents/jdchain/peer2
dev.peer2.host=127.0.0.1
dev.peer2.initPort=8820
dev.peer2.consensusPort=36020
dev.peer2.visitPort=17820
dev.peer2.serverPort=9020
dev.peer2.mediatorUrl=localhost:9000/deploy
dev.peer2.gatewayBindPpeer=true
dev.peer2.peerPubKey=3snPdw7i7Pi6PYrxgrcXKCEV78VpJQAZSy6YAbZS8bwquVeURrah3w
dev.peer2.peerPrivKey=177gjxt5VtjjQ6S8Jge7tuGQ6WZbFtwUrrN2Ek3iMDu4xLrt9A71pbHXtWZSzgabmTZbAUG
dev.peer2.peerPasswd=DYu3G8aGTMBW1WrTw76zxQJQU4DHLw9MLyy7peG4LKkY
#peer3
dev.peer3.peerId=3
dev.peer3.peerName=d.com
dev.peer3.basePath=/Users/zhaoguangwei/Documents/jdchain/peer3
dev.peer3.host=127.0.0.1
dev.peer3.initPort=8830
dev.peer3.consensusPort=36030
dev.peer3.visitPort=17830
dev.peer3.serverPort=9030
dev.peer3.mediatorUrl=localhost:9000/deploy
dev.peer3.gatewayBindPpeer=true
dev.peer3.peerPubKey=3snPdw7i7PeGmrc7jyX2b8gr7vyjgV64zmkiVvjZ4FuyJSuPMw7D87
dev.peer3.peerPrivKey=177gjsXgUPgGwJQsEAbvKWRSiBgLv7PqV58XwywyXuVDfhYB6UMJrHCUEfaMGFexf8bZ3px
dev.peer3.peerPasswd=DYu3G8aGTMBW1WrTw76zxQJQU4DHLw9MLyy7peG4LKkY
```

执行命令如下：

```
java -jar ump-1.0.1.RELEASE.jar --spring.config.location=file:./0application.properties
java -jar ump-1.0.1.RELEASE.jar --spring.config.location=file:./1application.properties
java -jar ump-1.0.1.RELEASE.jar --spring.config.location=file:./2application.properties
java -jar ump-1.0.1.RELEASE.jar --spring.config.location=file:./3application.properties
```

4) 4个ump应用启动之后，

通过执行：<http://localhost:9000/deploy/getAllConf4dev> 来加载配置，然后执行：<http://localhost:9000/deploy/md/planLedger>

来完成节点的初始化配置。如果执行成功，提示如下：

```
[“a.com make conf successfully.”,“d.com make conf successfully.”,“b.com make conf successfully.”,“c.com make conf successfully.”]
```

5) 查看每个peer的配置文件是否已经修改成功，位置如下：peerxxx/config/init中的三个文件：bftsmart.config ledger.init local.conf。

6) 启动每个peer的ledger-init.sh脚本，位置：peerxxx/bin/ledger-init.sh

7) 启动每个peer的startup.sh脚本，位置：peerxxx/bin/startup.sh。**注意：**如果在单机上启动4个peer节点，则需要保证脚本中的端口不同，不然会出现冲突。还有在单机启动4个节点，内存占用较高，可以将startup.sh脚本中的内存参数改为512M和1g。

8) 修改网关的配置文件，启动网关(具体参见部署文件，位置：<http://ledger.jd.com/downloaddocs.html>)。

将网关zip安装包解压至gw下，配置文件在：gw/config/gateway.conf。

填写网关和共识节点的信息，以及使用的公私钥信息。

gateway.conf

折叠源码

```
#网关的HTTP服务地址；
http.host=192.168.151.45
#网关的HTTP服务端口；
http.port=18081
#网关的HTTP服务上下文路径，可选；
#http.context-path=

#共识节点的服务地址；
peer.host=192.168.151.39
#共识节点的服务端口；
peer.port=17800
#共识节点的服务是否启用安全证书；
peer.secure=false
#共识节点的服务提供解析器
peer.providers=com.jd.blockchain.consensus.mq.MsgQueueConsensusProvider

data.retrieval.url=http://192.168.151.39:10001

#默认公钥的内容（Base58编码数据）；
keys.default.pubkey=endPsK36koyFr1D245Sa9j83vt6pZudFBJoJRB3xAswM6cwhRbna
#默认私钥的路径；在 pk-path 和 pk 之间必须设置其一；
keys.default.privkey-path=
#默认私钥的内容（加密的Base58编码数据）；在 pk-path 和 pk 之间必须设置其一；
keys.default.privkey=
```

```
#默认私钥的解码密码:  
keys.default.privkey-password=
```

无标签