## BTD节点搭建并同步数据块

## 系统准备

1. 节点部署准备，采用Linux Centos7.0或Ubuntu16.0以上系统;
2. 全节点，数据硬盘至少500G;（也可设置为非全阶段，只保存部分区块节点）
3. 修改新增/etc/hosts文件，添加如下配置信息：

52.221.197.208 chain1.bitdisk.io

13.229.130.119 chain2.bitdisk.io

54.251.142.15 chain3.bitdisk.io

18.136.100.51 chain4.bitdisk.io

18.136.102.103 chain5.bitdisk.io

## 将chainsqld 和配置文件拷贝至指定目录

1. 将chainsqld 和配置文件chainsqld.cfg 文件拷贝至/home/BTD 目录；
2. 新建数据库存放路径/home/data/db；
3. db保存为已同步区块数据；

## 链/节点启动

1. 每个节点配置文件生成后，将可执行文件(chainsqld)与节点对应配置文件(chainsqld.cfg) 分别放到各个节点
2. 执行启动命令（配置文件名为chainsqld.cfg时可不加--conf）：

nohup ./chainsqld &

* 注意：每个网络节点均要执行上述命令，使chainsql服务在后台运行

1. 检查各节点chainsqld程序运行情况

watch ./chainsqld server\_info

* 注意：节点第一次启动时，约一分钟左右在字段"complete\_ledgers"中输出类似 "1-10" 的显示，表明chainsqld服务启动成功

1. 查看其它节点运行情况

watch ./chainsqld peers

## 退出终端

执行exit命令退出终端

* 注意：不能直接强制关闭，否则会杀死nohup模式启动的chainsqld进程

## 节点停止

./chainsqld stop

* 注意：stop指令执行完毕后，可以执行以下指令查看chainsqld服务状态

ps –ef|grep chainsqld

## 查看本节点运行状态

watch ./chainsqld server\_info

看到complete\_ledgers 字段数字在递增说明当前链状态正常。

## 查看信任节点运行状况

watch ./chainsqld peers

说明：

1. 如果complete\_ledgers 字段的数字在增加，表明当前节点状态正常
2. 如果某个peer的complete\_ledgers数字没有变化，说明节点状态不正常，并且不能进行交易。

## 链交易验证

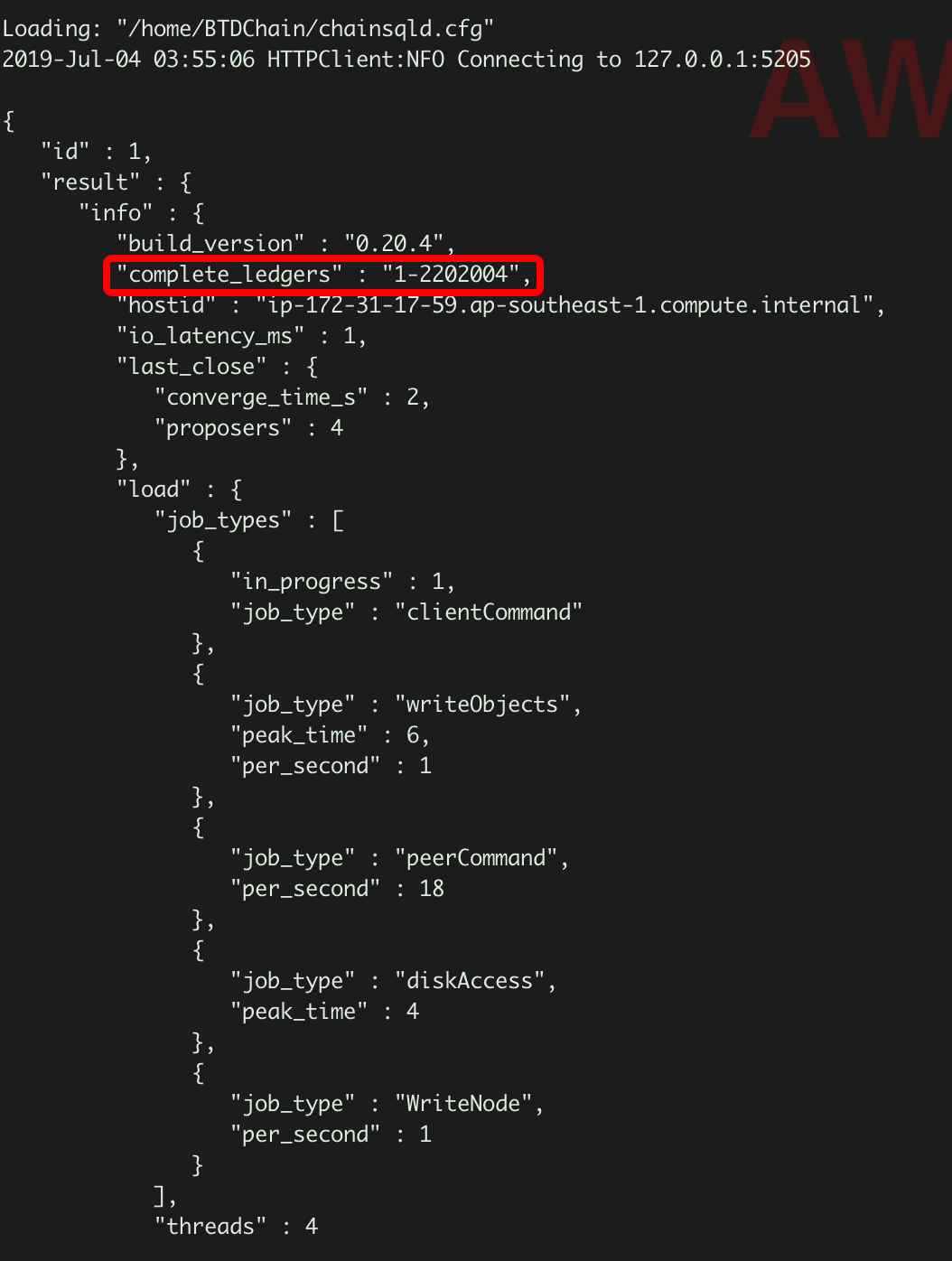
可通过发送一笔账户激活交易来验证链状态是否正常。

可以通过查询历史交易记录来验证链的历史数据是否正确

## 节点数据块同步

watch ./chainsqld server\_info

查看complete\_ledeger字段如果该字段不断向前则说明节点同步数据正常（初始节点同步数据需要一个过程，大概要同步200多万已用的链区块数据，请耐心等待）



## BTD主网钱包生成

## 2.1批量生成BTD钱包账户

以[Nodejs SDK](https://github.com/aeternity/aepp-sdk-python)服务为例，节点服务器上运行BTD\_Server服务：

生成BTD钱包账号地址和私钥

接口： <http://localhost:3000/new-account>

请求方式：POST

参数：无

返回：

{

"code":1,

"data":

{

address: "zNHC6Ce8o4d8AqF8DraKgh1g2uerh4KvPF"

publicKey: "cB4\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*sm"

secret: "xxHP\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*E6w"

},

"info":"操作成功"

}

## BTD 主网转账操作

## 3.1 用户充值及提现

以[Nodejs SDK](https://github.com/aeternity/aepp-sdk-python)服务为例，下载安装完成后，节点服务器上运行SDK服务：

账号充值及提现转账

接口： http://localhost:3000/ transfer

请求方式：POST

请求参数：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
| user | object | 转账钱包 |
| address | String | 接受转账地址 |
| amount | String | 数量 |
| currency | String | Enum(‘BTD’)币种 |
| msg | String | 消息 |
| msgType | String | 消息类型 |

User对象

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
| address | String | 钱包地址 |
| secret | String | 私钥 |

请求参数示意：

{

"user": { //转账地址和私钥

"secret":"xh9W\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*zCRL",

"address":"znnsshTWEeRd5Q7rhqZS1C9rH8attnC8rU"

},

"address": "zN6NM2oz78RqbGXTibwNEfm1srMNsc4Z1o", //到账地址

"amount": "1", //数量

"currency": "BTD", //币种，写死为‘BTD’

"msg": "", //备注信息

"msgType": "pay" //备注类型

}

返回：

{

"code":1,

"data":{

"tx\_hash":"257A0EF847F5FDF2E7053A30A383AF2" //交易哈希

},

"info":"操作成功"

}

转账失败返回

{

"code": -1,

"data":{

},

"info":"操作失败"

}

## 3.2 获取交易历史

以[Nodejs SDK](https://github.com/aeternity/aepp-sdk-python)服务为例，下载安装完成后，节点服务器上运行SDK服务：

查看账号地址历史交易

接口： [http://localhost:3000/ get-transaction-list](http://localhost:3000/%20get-transaction-list%20)

请求方式：POST

参数：

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 备注 |
| address | String | 地址 |
| ledger | String | 接口返回表示如果有返回表示还有下一页 |
| seq | String | 同ledger |
| pageSize | Int | 交易记录分页数 |

请求参数示意：

{

"address": "z96km7xeC13B9jFxkEvc6nq4LoWs6P4Bpg",

"pageSize": 1,

"ledger": "722393", // 上一次请求返回，如果没有返回即为最后一页

"seq": 11212, // 上一次请求返回，如果没有返回即为最后一页

}

返回：

{

"code":1,

"data":

{

"rows":

[{

"fromAddr":"znnsshTWEeRd5Q7rhqZS1C9rH8attnC8rU", //转出地址

"toAddr":"zN6NM2oz78RqbGXTibwNEfm1srMNsc4Z1o", //转入地址

"hash":"257A0EF847F5FD64383FBDF2E705", //交易hash

"currency":"BTD",

"amount":"1",

"msg":{"msg":"","msgTpye":""} //备注

}],

"marker":{"ledger":722393,"seq":0} //如果没下一页不会返回该字段

},

"info":"操作成功"

}`