

运维经验

运维问题

```
1 ##### Near方面
2 ##### 系统选择
3 ubuntu18.04
4 ubuntu20.04
5 一般情况下 Ubuntu 最新长期支持版
6 1. 区块链项目大多数文档中支持Ubuntu 或者文档只用Ubuntu系统做例子
7 2. 出个项目文档明确指定Ubuntu 版本，通常用最新长期支持版
8 ##### 系统配置
9 CPU 8c (CPU Features CMPXCHG16B, POPCNT, SSE4.2, AVX) (neard程序使用都是单核
   存档节点都是使用单核性能 很奇怪 然后同步慢)
10 内存 16GB
11 存储 验证节点 rpc节点 1T ssd
12     存档节点 10T NVMe (tar存档4.4T) (空间不够可以边下边解压，或者从开始同步)
13 边下载边解压 -> 4.8T 6T NVMe 存储
14 wget -c https://near-protocol-public.s3.ca-central-1.amazonaws.com/backups/mainnet/archive/data.tar -O - | tar -x
15 curl https://near-protocol-public.s3.ca-central-1.amazonaws.com/backups/mainnet/archive/data.tar | tar -x
16 ##### Node 类型选择
17 Validator Node: 验证器节点，存储大小和RPC节点一样，如果不需要运行验证器，可以选择rpc节点
18 RPC Node: 提供RPC服务，在给人提供服务的情况下，CPU和内存需要加配置，根据自己提供人数情况下，自己裁定
19 Archival Node: 默认不运行这个模式，这个节点需要的存储空间很大，需要查询老区块数据时和运行Indexer节点时需要，Indexer节点需要从genesis开始区块9820214解析数据，否则很容易卡块
20
21
22 ##### 运行2个节点
23 a节点 存档模式
24 b节点 非存档模式
25 同步运行2个节点 只需区别运行目录和修改config.json端口配置 运行时修改运行主目录
26
27 # 其中一个修改配置端口
28 # rpc端口 config.json
29 "rpc": { "addr": "0.0.0.0:3030", }
```

```
30 "network": { "addr": "0.0.0.0:24567", }
31
32 # 运行
33 a节点 存档模式
34 neard --home ~/.near/mainnet run --archive
35 b节点 非存档模式
36 neard --home ~/.near/mainnet2 run
37
38
39 ##### Indexer方面
40 ##### 卡块
41 先用非严格模式运行 跳过有问题区块 运行正常没有卡块后再切换到严格模式运行
42 --stop-after-number-of-blocks 100 运行100区块后停止
43 # 非严格模式运行
44 cargo run --release -- --home-dir ~/.near/mainnet run --store-genesis --stream-while-syncing --non-strict-mode --stop-after-number-of-blocks 100 sync-from-block --height 55095298
45 # 严格模式运行
46 cargo run --release -- --home-dir ~/.near/mainnet run --store-genesis --stream-while-syncing --concurrency 1 sync-from-block --height 55095398
47
48
49 ##### 什么时候启动多线程
50 从头开始时 最后选择1个线程开始同步 多个线程数量不能提高运行速度 反而会降低速度或者卡死
51 如果已经有数据的情况 但是随着版本的升级 数据库更新多了些字段, 检测内容多了东西 可以开100或更多线程从头开始索引数据 将老区块中新内容检测上数据库
52
53
54 ##### 大版本升级方式
55 最好1台服务器上日常运行2个indexer节点 或者日常运行1个存档节点数据 等到升级时临时添加非存档节点
56 a节点 运行存档模式
57 b节点 运行非存档模式
58 升级方法和顺序
59 1. 先下载最新GitHub文件 near-indexer-for-explorer
60 2. 执行数据库升级 (这里可以热升级 单一般)
61 cargo install diesel_cli --no-default-features --features "postgres"
62 diesel migration run
63 3. 使用非存档节点目录运行 near-indexer-for-explorer (有时候升级版本时 neard会升级数据库版本 在大升级时非常耗时 存档节点数据多耗时更多)
64 4. 等待非存档节点同步到最新区块并且文档后 升级存档节点数据
65
```

66

67 ##### 监控Indexer是否卡块

68 1. 监控数据库最新快是否卡块

69 # 查看数据库最高快高是否更新 10检测一次 如果10秒内最高快高度没有更新 说明很久没有新的解析数据到来 可以发送异常警告

70 #!/bin/bash

71 cmd1=`PGPASSWORD='{{password}}' psql -h {{IP}} -U {{username}} {{dbname}} -c 'select block_height from blocks order by block_height DESC limit 1;' |tail -n 3 | head -n 1 |awk '{print \$1}'`

72 echo \$cmd1

73

74

75 ##### 检测 tag

76 1. 使用github网站api获取最新tag github api文档 <https://docs.github.com/en/rest/reference/releases>

77 <https://api.github.com/repos/near/near-indexer-for-explorer/releases/latest>

78 2. 获取最新tag版本后结合监控发送信息