隐私链节点搭建

一. 准备工作

1. 准备一台通过sps的sgx cpu功能的服务器
2. 安装ubuntu v20.04 操作系统
3. 能够科学上网的环境
4. 确认go、rust语言环境正常， go version，rustc -V
5. 确认有必备工具的使用权限 apt,bash,wget,curl

二、环境搭建

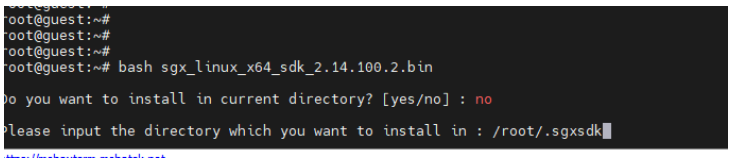
1. 安装sgx环境

```
# 下载官方提供好的脚本
wget https://raw.githubusercontent.com/SecretFoundation/docs/main/docs/node-guides/sgx
# 赋予执行权限
chmod +x sgx
# 执行安装脚本
bash sgx
# 检查是否安装成功,能看到/dev/isgx，说明安装成功
ls /dev/isgx
```

2. 安装Sgx sdk

```
# 下载 sgx sdk
wget https://download.01.org/intel-sgx/sgx-linux/2.14/distro/ubuntu20.04-
server/sgx_linux_x64_sdk_2.14.100.2.bin
# 赋予执行权限
chmod +x sgx_linux_x64_sdk_2.14.100.2.bin
# 创建安装目录
mkdir .sgxsdk
# 运行安装
bash sgx_linux_x64_sdk_2.14.100.2.bin
# 提示安装目录 一般我们安装再 /root/.sgxsdk 下面
# 输入 /root/.sgxsdk
```

安装过程，如下所示：



4. 安装scache

```
wget https://github.com/mozilla/sccache/releases/download/0.2.13/sccache-0.2.13-x86_64-unknown-linux-musl.tar.gz
tar xf ./sccache-*.tar.gz
mv ./sccache*/sccache "$HOME/sccache"
```

5. 安装ghmd

```
# 获取已经打包好的deb
# 目前是 0.0.3-dev 版本，后期更新请更换最新的
wget
https://github.com/HermitMatrixNetwork/HermitMatrixNetwork/releases/download/v0.0.3/hermitmatrixnetwork_0.0.3-dev_amd64.deb
# 安装
dpkg -i hermitmatrixnetwork_0.0.3-dev_amd64.deb
# 检查，看到输出安装成功
ghmd -h
```

三、节点基础配置

1. 初始化enclave

```
# 创建存放sgx远程证明证书的目录,这个证书有因特尔签名的报告
mkdir -p /opt/ghm/.sgx_ghms
# 创建环境变量
# /usr/lib 下存放了三个重要的.so 动态库，与sgx相关
export GHM_ENCLAVE_DIR=/usr/lib
export GHM_SGX_STORAGE=/opt/ghm/.sgx_ghms
# 初始化enclava环境,生成一个英特尔签名的远程证明证书
ghmd init-enclave
# 检查是否有生成证书
ls -h /opt/ghm/.sgx_ghms/attestation_cert.der
# 检查aesmd服务是否正常
service aesmd status
```

2. 防火墙端口开放

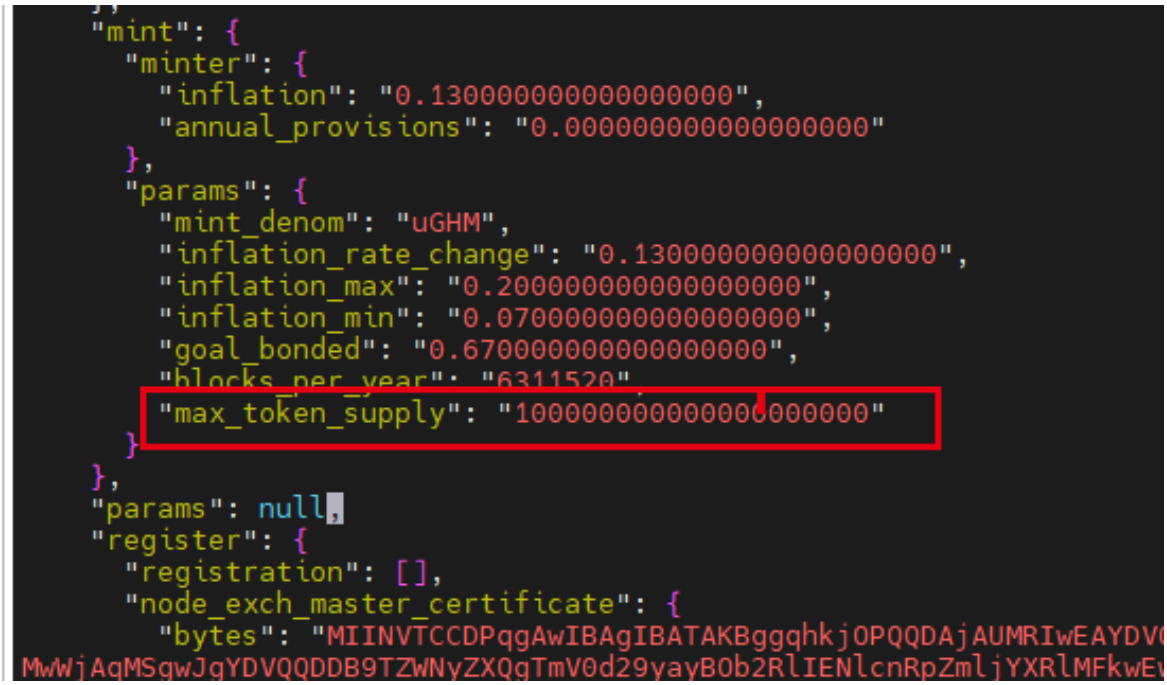
```
# 开启相关端口
ufw allow 26656  # 开放p2p端口
ufw allow 26657  # 开放tendermint端口（可选）
ufw allow 9091   # grpcweb（可选）
ufw allow 9090   # grpc（可选）
ufw allow 1317   # api（可选）
```

3. 设置链参数

```
# 设置链的id
ghmd config chain-id ghmdev
# 设置key模式为test环境
ghmd config keyring-backend test
# 用节点名字和链ID初始化链
ghmd init <node-name: node1> --chain-id <chain-id: ghmd-test>

# 修改 app 中 gas 代币的名字，创世文件中的代币名字
# stake -> uGHM
perl -i -pe 's/"stake"/ "ughm"/g' ~/.ghmd/config/genesis.json

# 修改mint 模块下的最大代币供应量数值， 如下面图：
```



4. 设置初始账号金额

```
# 添加一个账号,助记词会打印在屏幕
ghmd keys add a
ghmd keys add b
ghmd keys add c

# 将账号a,b,c的初始金额信息，写入创世块配置
ghmd  add-genesis-account "$(ghmd keys show -a a)"  100000000000000000ughm
ghmd  add-genesis-account "$(ghmd keys show -a b)"  100000000000000000ughm
ghmd  add-genesis-account "$(ghmd keys show -a c)"  100000000000000000ughm
```

四. 引导节点配置

1. 创建链的第一个验证器

```
# 生成一笔交易： 账号 a 委托的第一个验证器  1 GHM  = 1000000 uGHM  最少自我委托是 1GHM
# 注意要加上gas-prices
ghmd gentx a 1000000ughm --chain-id ghmdev --gas-prices 0.25ughm

# 将这比交易收集进入genesis.json 中
ghmd collect-gentxs

# 验证是genesis.json 是否有效
ghmd validate-genesis
```

2. 初始化节点

```
# 初始化引导节点
ghmd init-bootstrap

# 将节点启动在后台
mkdir logs
nohup ghmd start --rpc.laddr tcp://0.0.0.0:26657 --bootstrap >./logs/nohup.out 2>&1 &

# 查看链的状态信息
ghmd status
```

四. 节点加入

1. 配置链信息

```
# 指向引导节点或全节点
ghmd config node tcp://45.32.116.172:26657
# 给自己节点配置一个名字
ghmd init <node_name:test_node1> --chain-id <chain_id: ghmdev>
# 设置钱包模式为测试环境
ghmd config keyring-backend test
# 创建一个钱包，用来管理这个节点
ghmd keys add fcihpy
# 给上面这个钱包转账
# ghmd tx bank send <from_addr> <to_address> amount --gas-prices 0.0125uGHM
ghmd tx bank send ghm1jvtdv8674llygwn8s0d7z47uctyjf9uxu9p8k4
ghm17tuk77p0eqs8nqevwhqs9lrsqm54e56uef6rey 1050000uGHM --gas-prices 0.0125uGHM
```

2. 获取引导节点信息

```
# 在引导节点或全节点，查询节点id
ghmd tendermint show-node-id
# 在自己节点上做如下操作
export LANG="en_US";export LANGUAGE="en_US";export LC_ALL="en_US";top

PERSISTENT_PEERS="<上一步获取到的id，如：
bdc731280afb3c1fa8e6396eda4be59b333b3fc5>@45.32.116.172:26656"
# 将引导节点信息写入配置文件
sed -i 's/persistent_peers = ""/persistent_peers = "'$PERSISTENT_PEERS'"/g'
~/.ghmd/config/config.toml
# 显示种子
echo "Set persistent_peers: $PERSISTENT_PEERS"
```

3. 注册节点

```
# --from fcihpy 刚才钱包的名字
ghmd tx register auth /opt/ghm/.sgx_ghms/attestation_cert.der -y --from dafni --gas-
prices 0.25uGHM
# 根据上一步操作的交易，查看执行情况
ghmd q tx <hash_str>
```

4. 提取链的共享种子

```
# 提取公钥
PUBLIC_KEY=$(ghmd parse /opt/ghm/.sgx_ghms/attestation_cert.der 2> /dev/null | cut -c
3- )

# 注册成功则可以拿到共享种子
SEED=$(ghmd q register seed "$PUBLIC_KEY" 2> /dev/null | cut -c 3-)

# 打印种子信息，确保不能为空
echo "SEED: $SEED"
```

5. 设置网络证书

```
# 这个命令会生产两个证书
ghmd q register secret-network-params 2> /dev/null

ghmd configure-secret node-master-cert.der "$SEED"
```

6. 替换创世文件

```
# 从引导节点下载创世文件，替换 (/root/.ghmd/config/genesis.json)

# 校验创世文建是否有效
ghmd  validate-genesis
```

7. 运行节点

```
# 指回自己
ghmd config node tcp://localhost:26657

# 运行节点
mkdir logs
nohup ghmd start >./logs/nohup.out 2>&1 &
```

8. 检查运行情况

```
# 查看链的运行状态
ghmd status

# 查看网络监听
netstat -ntlp

# 查看日志，检查是否在同步
tail -f logs/nohup.out
```

五. 成为验证节点

```
# 注意 1 GHM = 1000000ughm
  ghmd tx staking create-validator \
  --amount=1000000ughm \
  --pubkey=$(ghmd tendermint show-validator) \
  --identity=caca-boxi \
  --details="To infinity and beyond!" \
  --commission-rate="0.10" \
  --commission-max-rate="0.20" \
  --commission-max-change-rate="0.01" \
  --min-self-delegation="1" \
  --moniker=caca \
  --from=caca \
  --chain-id=ghmdev \
  --gas-prices 0.25ughm
  # --moniker 指向自己节点的名字,在此命令中设置的的信息：ghmd init testnode1 --chain-id ghmdev
  # --from 治理的id：  为 ghmd keys add key1
```