

How to run Pchain with Ubuntu

1. Ubuntu configuration

1) install go

wget <https://dl.google.com/go/go1.11.4.linux-amd64.tar.gz>

```
ubuntu@ip-172-31-12-155:~$ wget https://dl.google.com/go/go1.11.4.linux-amd64.tar.gz
--2019-01-10 10:01:23-- https://dl.google.com/go/go1.11.4.linux-amd64.tar.gz
Resolving dl.google.com (dl.google.com)... 172.217.0.46, 2607:fb0:4005:809::200e
Connecting to dl.google.com (dl.google.com)|172.217.0.46|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 126643341 (121M) [application/octet-stream]
Saving to: 'go1.11.4.linux-amd64.tar.gz'

go1.11.4.linux-amd64.tar.gz          100%[=====] 120.78M  53.7MB/s   in 2.2s

2019-01-10 10:01:26 (53.7 MB/s) - `go1.11.4.linux-amd64.tar.gz' saved [126643341/126643341]
```

sudo tar -xzf go1.10.4.linux-amd64.tar.gz -C /usr/local

sudo ln -s /usr/local/go/bin/ /usr/bin/

```
ubuntu@ip-172-31-12-155:~$ sudo tar -xzf go1.11.4.linux-amd64.tar.gz -C /usr/local
ubuntu@ip-172-31-12-155:~$ sudo ln -s /usr/local/go/bin/ /usr/bin/
```

2) mkdir -p \$HOME/go-workspace/src

mkdir -p \$HOME/go-workspace/pkg

mkdir -p \$HOME/go-workspace/bin

3) Config go

sudo vim \$HOME/.bashrc

insert in the bottom

```
export GOROOT=/usr/local/go  
export GOPATH=$HOME/go-workspace  
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

Press Esc, :wq save and exit. Make sure the file takes effect.

```
source $HOME/.bashrc
```

4) check go

```
go version
```

```
ubuntu@ip-172-31-12-155:~$ go version  
go version go1.6.2 linux/amd64
```

```
go env
```

```
ubuntu@ip-172-31-12-155:~$ go version  
go version go1.6.2 linux/amd64  
ubuntu@ip-172-31-12-155:~$  
ubuntu@ip-172-31-12-155:~$  
ubuntu@ip-172-31-12-155:~$  
ubuntu@ip-172-31-12-155:~$  
ubuntu@ip-172-31-12-155:~$  
ubuntu@ip-172-31-12-155:~$  
ubuntu@ip-172-31-12-155:~$ go env  
GOARCH="amd64"  
GOBIN=""  
GOEXE=""  
GOHOSTARCH="amd64"  
GOHOSTOS="linux"  
GOOS="linux"  
GOPATH="/home/ubuntu/go"  
GORACE=""  
GOROOT="/usr/lib/go-1.6"  
GOTOOLDIR="/usr/lib/go-1.6/pkg/tool/linux_amd64"  
GO15VENDOREXPERIMENT="1"  
CC="gcc"  
GOGCCFLAGS="-fPIC -m64 -pthread -fmessage-length=0"  
CXX="g++"  
CGO_ENABLED="1"  
ubuntu@ip-172-31-12-155:~$
```

2、 Install git

```
ubuntu@ip-172-31-12-155:~$ sudo apt-get install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-aws-headers-4.4.0-1066 linux-aws-headers-4.4.0-1069 linux-aws-headers-4.4.0-1070 linux-he
  linux-image-4.4.0-1066-aws linux-image-4.4.0-1069-aws linux-image-4.4.0-1070-aws
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-arch git-
  git
0 upgraded, 1 newly installed, 0 to remove and 37 not upgraded.
Need to get 3,176 kB of archives.
After this operation, 24.1 MB of additional disk space will be used.
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 git amd64 1:2.7.4-
Fetched 3,176 kB in 0s (60.4 MB/s)
Selecting previously unselected package git.
(Reading database ... 187261 files and directories currently installed.)
Preparing to unpack .../git_1%3a2.7.4-0ubuntu1.6_amd64.deb ...
Unpacking git (1:2.7.4-0ubuntu1.6) ...
Setting up git (1:2.7.4-0ubuntu1.6) ...
ubuntu@ip-172-31-12-155:~$
```

Check git version

```
ubuntu@ip-172-31-12-155:~$ git version
git version 2.7.4
```

3、 Pchain download and edit

Create folders:

```
mkdir -p pchain/src/github.com/
```

```
mkdir pchain/bin/
```

Source Code

Notice: source has to be under current directory.

```
cd pchain/src/github.com/
```

```
git clone https://github.com/pchain-org/pchain.git
```

```
ubuntu@ip-172-31-12-155:~/pchain/src/github.com$ git clone https://github.com/pchain-org/pchain.git
Cloning into 'pchain'...
remote: Enumerating objects: 600, done.
remote: Counting objects: 100% (600/600), done.
remote: Compressing objects: 100% (418/418), done.
remote: Total 22812 (delta 327), reused 338 (delta 172), pack-reused 22212
Receiving objects: 100% (22812/22812), 56.29 MiB | 17.92 MiB/s, done.
Resolving deltas: 100% (13593/13593), done.
Checking connectivity... done.
```

Enter pchain directory:

```
ubuntu@ip-172-31-12-155:~$ cd pchain
ubuntu@ip-172-31-12-155:~/pchain$ ls
bin  src
```

```
cd pchain/src/github.com/pchain/restore/
```

vi build.sh

```
#!/bin/sh
set -e

PWD=`pwd`
echo $PWD
workspace=$PWD
pchaindir="$workspace/src/github.com/pchain"

GOPATH="$workspace"
export GOPATH
echo $GOPATH
cd "$pchaindir"
echo `pwd`
make build
#go build -i -v -o $workspace/bin/pchain ./cmd
```

Press Esc, :wq save and exit. Make sure the file takes effect.

Copy the build.sh script file to the pchain directory:

Command:

```
cp build.sh $HOME/pchain/
```

```
ubuntu@ip-172-31-12-155:~/pchain$ ls  
bin  build.sh  src
```

```
find . -name '*.sh' | xargs chmod +x
```

```
ubuntu@ip-172-31-12-155:~/pchain$ find . -name '*.sh' | xargs chmod +x  
ubuntu@ip-172-31-12-155:~/pchain$ ls  
bin  build.sh  src
```

Edit:

```
cd pchain
```

```
vim build.sh
```

```
Press Shift: config unix;
```

```
Input set ff=unix, Enter;
```

```
Press Esc, :wq save and exit.
```

```
#!/bin/sh
set -e
PWD=`pwd`
echo $PWD
workspace=$PWD
pchaindir="$workspace/src/github.com/pchain"
GOPATH="$workspace"
export GOPATH
echo $GOPATH
cd "$pchaindir"
echo `pwd`
make build
#go build -i -v -o $workspace/bin/pchain ./cmd
~  
~  
~  
~  
~  
~  
~  
~  
:set ff=unix
```

Run build script

```
./build.sh
```

```
ubuntu@ip-172-31-12-155:~/pchain$ ./build.sh
/home/ubuntu/pchain
/home/ubuntu/pchain
/home/ubuntu/pchain/src/github.com/pchain
go build -o /home/ubuntu/pchain/bin/pchain ./cmd/
```

After compile, you will find the binary executable file pchain under pchain/bin/

4.Run Pchain

```
cd $HOME/pchain/bin/
```

```
sudo chmod +x pchain
```

1) Init the local account:

```
./pchain --datadir .pchain init_eth_genesis
"{}1000000000000000000000000,100}"
```

```
ubuntu@ip-172-31-12-155:~/pchain/bin$ ./pchain --datadir .pchain init_eth_genesis "{}1000000000000000000000000,100"
INFO [01-10|10:28:45.879] this is init eth genesis
INFO [01-10|10:28:46.663] createPriValidators
account:=0x0388Eaa50925AfE3b43828ef853774e2b7E16f79 pvd:=pchain
ubuntu@ip-172-31-12-155:~/pchain/bin$
```

Check

```
ubuntu@ip-172-31-12-155:~/pchain/bin$ ll
total 36320
drwxrwxr-x 4 ubuntu ubuntu 4096 Jan 10 10:28 .
drwxrwxr-x 4 ubuntu ubuntu 4096 Jan 10 10:21 ../
drwx----- 3 ubuntu ubuntu 4096 Jan 10 10:28 log/
-rwxrwxr-x 1 ubuntu ubuntu 37172776 Jan 10 10:24 pchain*
drwx----- 3 ubuntu ubuntu 4096 Jan 10 10:28 .pchain/
ubuntu@ip-172-31-12-155:~/pchain/bin$
```

2) Init the local pchain

```
./pchain --datadir .pchain init .pchain/pchain/eth_genesis.json
```

```
ubuntu@ip-172-31-12-155:~/pchain/bin$ ./pchain --datadir .pchain init .pchain/pchain/eth_genesis.json
INFO [01-10|10:29:20.962] init_eth_blockchain 0 with dbPath: .pchain/pchain/geth/chaindata
INFO [01-10|10:29:20.962] Allocated cache and file handles database=.pchain/pchain/geth/chaindata cache=16 handles=16
INFO [01-10|10:29:20.962] init_eth_blockchain
INFO [01-10|10:29:20.962] init_eth_blockchain 2
INFO [01-10|10:29:20.968] Writing current genesis block
INFO [01-10|10:29:20.968] Persisted tries from memory database nodes=1 size=270.00B time=31.931ms gcnodes=0 gcsizes=0.00B gctime=0s livenodes=1 livesize=0.00B
INFO [01-10|10:29:20.968] init_eth_blockchain end
INFO [01-10|10:29:20.968] successfully wrote genesis block and/or chain rule set: b5155420a9b2ccb2fc9197e3ca9ef5ad81206c5400d8a965481a3b06e82b8c8
INFO [01-10|10:29:20.968] Database closed database=.pchain/pchain/geth/chaindata
INFO [01-10|10:29:20.969] checkAccount() coinbase is 0388eaa50925afe3b43828ef853774e2b7e16f79
INFO [01-10|10:29:20.969] checkAccount(), address is 0388eaa50925afe3b43828ef853774e2b7e16f79, balance is 10000000000000000000000000000000, amount is 100
ubuntu@ip-172-31-12-155:~/pchain/bin$
```

3) Run pchain

Copy run.sh under pchain directory then run ./run.sh

touch run.sh

```
sudo chmod +x run.sh
```

Mining mode: (Default add --mine)

```
#!/bin/bash
```

```
./pchain --datadir .pchain      --networkid=10000 --port=30201  
--syncmode "full"   --mine       --rpc           --rpcport    6969  
--rpcaddr=0.0.0.0          --rpccorsdomain=*  --mine       --rpcapi  
"eth,web3,personal,net,admin,miner,tdm"      --verbosity=4    >  
pchain.log 2>&1 &
```

Non-mining mode:

```
#!/bin/bash
```

```
./pchain --datadir .pchain      --networkid=10000 --port=30201  
--syncmode "full"   --rpc           --rpcport  6969      --rpcaddr=0.0.0.0  
--rpccorsdomain=*                           --rpcapi  
"eth,web3,personal,net,admin,miner,tdm"      --verbosity=4    >  
pchain.log 2>&1 &
```

and then run this file

```
./run.sh
```

Check running log

```
tail -f pchain.log
```

```
ubuntu@ip-172-31-12-155:~/pchain/bin$ ./run.sh
ubuntu@ip-172-31-12-155:~/pchain/bin$ tail -f pchain.log
Validators:
    Validator(ADD:24494A76BCF508033F771BCF408973E43F38383C PK:0071A435043F78CC14AD09AD2D28ABE15E56664E3D925434E49549AE1CD45E1434A792CE7F5F2A695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6
075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100)
DEBUG[01-10|10:56:56.236] addRound(round) round=0
DEBUG[01-10|10:56:56.236] addRound(round) round=0
INFO[01-10|10:56:56.236] Tendermint (backend) GetBroadcaster: 0x0004a1e0
INFO[01-10|10:56:56.236] Tendermint (backend) GetBroadcaster: 0x0004a1e0
INFO[01-10|10:56:56.236] Received StepPreCommitWait from miner. RoundStepNewHeight has been set with height: 164
INFO[01-10|10:56:56.236] Received tick. old_til: (2s 163 0 RoundStepPreCommitWait, new_til: (999.05632ms 164 0 RoundStepNewHeight)
INFO[01-10|10:56:56.236] Scheduler timeout. dur: 999.05632ms, height: 164, round: 0, step: RoundStepNewHeight
INFO[01-10|10:56:56.236] Timed out. dur: 999.05632ms, height: 164, round: 0, step: RoundStepNewHeight
INFO[01-10|10:56:56.236] step is RoundStepNewHeight
INFO[01-10|10:56:56.236] enterNewRound(164/0). Current: 164/0/RoundStepNewHeight
INFO[01-10|10:56:56.236] Validators: ValidatorSet{
    Validators:
        Validator(ADD:24494A76BCF508033F771BCF408973E43F38383C PK:0071A435043F78CC14AD09AD2D28ABE15E56664E3D925434E49549AE1CD45E1434A792CE7F5F2A695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6
075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100)
}
DEBUG[01-10|10:56:57.236] addRound(round) round=1
DEBUG[01-10|10:56:57.236] addRound(round) round=1
INFO[01-10|10:56:57.236] cs_proposer.cs_Height.cs_Round are ({163 0 Validator(ADD:24494A76BCF508033F771BCF408973E43F38383C PK:0071A435043F78CC14AD09AD2D28ABE15E56664E3D925434E49549AE1CD45E1434A792CE7F5F2A695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100), C7E7F5F2AA695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100)
DEBUG[01-10|10:56:57.236] update proposer height:164 round=0 dvr=0
DEBUG[01-10|10:56:57.236] proposer_priv_validator are (Validator(ADD:24494A76BCF508033F771BCF408973E43F38383C PK:0071A435043F78CC14AD09AD2D28ABE15E56664E3D925434E49549AE1CD45E1434A792CE7F5F2A695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100), PrivValidator(24494A76BCF508033F771BCF408973E43F38383C))
DEBUG[01-10|10:56:57.236] IsProposer() return true
INFO[01-10|10:56:57.236] enterPropose(164/0). Current: 164/0/RoundStepNewRound
INFO[01-10|10:56:57.236] cs_proposer.cs_Height.cs_Round are ({164 0 Validator(ADD:24494A76BCF508033F771BCF408973E43F38383C PK:0071A435043F78CC14AD09AD2D28ABE15E56664E3D925434E49549AE1CD45E1434A792CE7F5F2A695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100), C7E7F5F2AA695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100)
DEBUG[01-10|10:56:57.236] proposer_priv_validator are (Validator(ADD:24494A76BCF508033F771BCF408973E43F38383C PK:0071A435043F78CC14AD09AD2D28ABE15E56664E3D925434E49549AE1CD45E1434A792CE7F5F2A695093193E0AD4B31C43E317DA0F976AF5CEDA4E8BCD67651721B5B56A6075803715168A727FD5C4AFCEC5CE441AD323B6677281818682B39F7B422D75A97BCAA470031863CF7526EEA17540CB832332CC5ED0617B39694 VP:100), PrivValidator(24494A76BCF508033F771BCF408973E43F38383C))
DEBUG[01-10|10:56:57.236] IsProposer() return true
```

How to become PCHAIN Validator

1. Join Pchain main network

1) reference:

<https://github.com/pchain-org/pchain/wiki/How-to-run-pchain>

2) Download Files:

https://s3.us-east-2.amazonaws.com/download.pchain.org/pchain_init_dirs.zip

3) Use the files in zip to replace 3 files as below in the local folder named pchain

.pchain/static-nodes.json

.pchain/pchain/genesis.json

.pchain/pchain/eth_genesis.json

4) init pchain with eth_genesis.json

cd \$HOME/pchain

`./pchain --datadir .pchain init .pchain/pchain/eth_genesis.json`

5) Run pchain

Run mining mode:

```
#!/bin/bash

./pchain --datadir .pchain      --networkid=10000 --port=30201
--syncmode "full"   --mine     --rpc  --rpcport 6969 --rpcaddr=0.0.0.0
--rpccorsdomain=*                                         --rpcapi
"eth,web3,personal,net,admin,miner,tdm"           --verbosity=4    >
pchain.log 2>&1 &
```

and then run this file

```
./run.sh
```

Check whether the main network is connected successfully.

Get the current block number:

```
curl -X POST -H "Content-Type:application/json" --data
'{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":1}' localhost:6969/pchain
```

```
pchain@pchain-01:~/pchain$ curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":1}' localhost:6969/pchain
{"jsonrpc":"2.0","id":1,"result":"0xdcfc9"}
```

Result blockNumber:{"jsonrpc":"2.0","id":1,"result":"0xdcfc9"}

Get block number 0xdcfc9 details:

```
curl -X POST -H "Content-Type:application/json" --data
'{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["0xdcf9
",true],"id":1}' localhost:6969/pchain
```

Check main network block number 0xdcfc9 details:

(Pchain Testnet Ip:34.221.29.3)

```
curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["0xdcf9c99d0000000000000000000000000000000000000000000000000000000000000000","true],"id":1}' 34.221.29.3:6969/pchain
```

Successful connection to primary network with the same hash value,

```
"hash":"0x723020e29263f7cda758b2c54a6a6b285eb43aaea7a899dc48  
984a78f1860ff8"
```

6) Checking account balance

Check account:

```
curl -X POST -H "Content-Type:application/json" --data  
'{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":1}'  
localhost:6969/pchain  
  
pchain@pchain-01:~/pchain$ curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"eth_accounts","params":[],"id":1}' localhost:6969/pchain  
{"jsonrpc":"2.0","id":1,"result":["0x5defded18fe902f3f005a81d25955522416c6470"]}  
pchain@pchain-01:~/pchain$
```

```
curl -X POST -H "Content-Type:application/json" --data  
'{"jsonrpc":"2.0","method":"eth_getBalance","params":["0x5defded18fe9  
02f3f005a81d25955522416c6470", "latest"],"id":1}' localhost:6969/pchain  
  
pchain@pchain-01:~/pchain$ curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"eth_getBalance","params":["0x5defded18fe902f3f005a8  
1d25955522416c6470", "latest"],"id":1}' localhost:6969/pchain  
{"jsonrpc":"2.0","id":1,"result":"0x0"}  
pchain@pchain-01:~/pchain$
```

2. Receive account balance

Link: <https://testnet.pchain.org/vfaucet.html>

https://testnet.pchain.org/vfaucet.html

Email address
z...@outlook.com
Fill in the email to get the verification code.
Send Email

Verification code
t8vdyidf

Address
0x5defded18fe902f3f005a81d25955522416c6470

Public ip
59...10

get 100k tPAI

https://testnet.pchain.org/vfaucet.html

PCHAIN

Email address
z...@outlook.com
Fill in the email to get the verification code.
Send Email

Verification code
07f5oyus

Address
0x5defded18fe902f3f005a81d25955522416c6470

Public ip
59...210

get 100k tPAI

Check:

```
curl -X POST -H "Content-Type:application/json" --data
```

```
'{"jsonrpc":"2.0","method":"eth_getBalance","params":["0x5defded18fe9  
02f3f005a81d25955522416c6470", "latest"],"id":1}' localhost:6969/pchain
```

```
pchain@pchain-01:~/pchains$ curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"eth_getBalance","params":["0x5defded18fe902f3f005a8  
02f3f005a81d25955522416c6470", "latest"],"id":1}' localhost:6969/pchain  
{"jsonrpc":"2.0","id":1,"result":"0x3f870857a3e0e3800000"}  
pchain@pchain-01:~/pchains$
```

```
{"jsonrpc":"2.0","id":1,"result":"0x3f870857a3e0e3800000"}
```

3. Vote:

- 1) Before bidding, users should prepare the following parameters:
{address, public key, the amount of tPAI, Hash}
- 2) Check the current epoch number.

```
curl -X POST -H "Content-Type:application/json" --data  
'{"jsonrpc":"2.0","method":"tdm_getCurrentEpochNumber","params":[],  
"id":1}' localhost:6969/pchain
```

```
pchain@pchain-01:~/pchains$ curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"tdm_getCurrentEpochNumber",  
"params":[],"id":1}' localhost:6969/pchain  
{"jsonrpc":"2.0","id":1,"result":11}  
pchain@pchain-01:~/pchains$
```

- 3) Check Validators details in current epoch, and get to know when to vote for the next Epoch.

```
curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"tdm_getEpoch","params":[11],"id":1}' localhost:6969/pchain

localhost:6969/pchain

[pchain@chain-01:~/pchains curl -X POST -H "Content-Type:application/json" --data '{"jsonrpc":"2.0","method":"tdm_getEpoch","params":[11],"id":1}' localhost:6969/pchain

{"jsonrpc":"2.0","id":1,"result":{"number":11,"reward_per_block":1286827650572402986,"start_block":893569,"start_time":"2019-01-14T13:28:51.819+00:00","end_time":"2001-01-01T00:00:00Z","vote_start_block":954147,"vote_end_block":962223,"reveal_start_block":962224,"reveal_end_block":970300,"status":3,"validators":[{"address":"0x4cacbcfb218679dc957490a2061bca4a8d8b6c","public_key":"085586041F70435700850E19B7D54B3F795C5EC4C6EC502D19030E04F42122823E5A765E56CA7BC457E50561F7B022313C38959AC303F3C957DB72824123F344778B95AC0E4679AFA4D1843357025C8895AC50B80458BF50367236B59AF2D5B0524053E40AD8C1DA0AB8A5D972F5E6D12031603F4D40D496973BD75E60","voting_power":100}, {"address":"0x79cd31b59e3faab68fbbaaf4da40d8fbb6","public_key":"7315DF293B07C52E6F6C1FC05018A1CA4F6B630F6F4D1216804FEDDC2F04CD2932A5B7286910145ED975A9FA0CDCB818F9281FDEA8033BF4259AC3400552065951D2440C25A994367E1DC60EF34B34C8B85CD95304B24F9A07473163F1F24C79AC5CBEC24085PEA80907F683ED044FD8341BF6FE179334105FED6E790","voting_power":100}, {"address":"0x6c179a651918888251380a4e3fee6fa81cf091d1","public_key":"72747AF4569D6BE89415A600F774B05A66439414C8C8688FC369956472F3A056471581713E0A99B5E95C216744CC68CCC0A0390A00824D80BC2679C989CD599631472C4000F64D779148a7427475716D6909F6695F3D2E3D919A1457D57FABDF4ACD7DB88BA35B7D6CDB8859FC4A5","voting_power":100}, {"address":"0xd2fd09246e2ced295411f1f863e6e7b5929bcc59","public_key":"0C75143EB5952A46803215D0A3E8F53E2245501C28CB0A285E624D6E74C6D703157710EBAS023EF1B082F08CEA27670E131745ED01954A48892AD1F8CF2602303B2D37035353410982E31AC15BEF8090283891593A312C9D917F2ED5C941164425FE4B19C1C653C3E2C1B544A3BABF85E0BEDE5D9E1F8C004C28A9C93E47F","voting_power":100}]}}

pchain@chain-01:~/pchains ]
```

Check current blockNumber:

```
curl -X POST -H "Content-Type:application/json" --data
'{"jsonrpc":"2.0","method":"eth_blockNumber","params":[],"id":1}'  
localhost:6969/pchain
```

4) Sealed-bid

Send Hash of vote to the PCHAIN chain through the RPC provided by PCHAIN for the Validator competition.

Get priv_validator.json the detail(address/privateKey/publicKey).

```
vi $HOME/.pchain/pchain/priv_validator.json
```

```
pchain@pchain-01:~/pchain$ cat .pchain/pchain/priv_validator.json
{
    "address": "5DEFDED18FE902F3F005A81D25955522416C6470",
    "consensus_priv_key": [
        "4",
        "3ED5C4F04EC492FE0993CDE99D32CA860FDEF4E[REDACTED]9CBA02785D5"
    ],
    "consensus_pub_key": [
        "4",
        "3042D450DC227A63CA1601E6DBED9C4F2EC6513B9CBA278EAF6F44093ED44802128644DA01C76859356ED223AC3E1DFA0C1DB8D175DFDEF104D96EBE3C103E7706D023D71A0F9
6833002CDC3F9A94F5C5210E4A9275A5FBC9A53EFC77803C3DD736E98F4550E8D9DB4E41C40C4B387A83A360B23'[REDACTED]211A6000A668AE"
    ]
}
pchain@pchain-01:~/pchain$
```

Unlock account:

```
unlock account:
curl -X POST -H "Content-Type:application/json" --data
'{"jsonrpc":"2.0","method":"personal_unlockAccount","params":["0x5DEFDED18FE902F3F005A81D25955522416C6470", "PASSWORD", 0],"id":1}'
http://localhost:6969/pchain
```

web3.getVoteHash(from,pubkey,amount,salt):

<https://github.com/pchain-org/pweb3/wiki/JavaScript-API#web3getVoteHash>

```
Example:
var Web3 = require('pweb3');
var web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));

var from = "0x5DEFDED18FE902F3F005A81D25955522416C6470";
var pubkey =
"0x[REDACTED]227A63CA1601E6DBED9C4F2EC6513B9CBA278EAF6F44093ED4480[REDACTED]DA01C76859356ED223AC3E1DFA0C1DB8D175DFDEF104D96EBE3C103E7706D023D71A0F96833002CD
C3F9A94F5C5210E4A9275A5FBC9A53EFC77803C3DD736E98F4550E8D9DB4E41C40C4B387A83A360B23E322051928211A60[REDACTED]";
var amount = "0x2a5a058fc295ec000000";
var salt = "PCHAIN";

var hash = web3.getVoteHash(from,pubkey,amount,salt);
console.log(hash);

Returns:
0x[REDACTED]98de00bf032bb608726e73254b5147e84032b05267b24bb5403[REDACTED]
```

Get return hash and voted:

https://github.com/pchain-org/pchain/wiki/JSON-RPC#tdm_votenextepoch

```
voted:
curl -X POST -H "Content-Type:application/json" --data
'{"jsonrpc":"2.0","method":"tdm_voteNextEpoch","params":["0x5DEFDED18FE902F3F005A81D25955522416C6470",
"0x[REDACTED]38de00bf032bb608726e73254b5147e84032b05267b24bb540[REDACTED]"],"id":1}' http://localhost:6969/pchain
```

5) Get the details of the vote includes epoch number, start block, end block and the number of votes.

https://github.com/pchain-org/pchain/wiki/JSON-RPC#tdm_getnextepochvote

tdm_getNextEpochVote

Returns the votes details of the next epoch

Parameters

none

Returns

`result` , string - votes detail of the next epoch, such as epoch number, start block, end block, votes

6) Reveal the vote

When an Epoch reaches 85%~95% of the total time period, the next round of Validator bidding begins. Upload the address, public key, and number of bets prepared for the previous round of bidding to the chain through the RPC provided by PCHAIN.

Sign Address

https://github.com/pchain-org/pchain/wiki/JSON-RPC#chain_signAddress

Sign the Address with BLS Private Key, return the BLS Signature to proof you are the owner of the BLS Public Key

Parameters

from: address, 20 Bytes - Address which will be signed

privateKey: hex string, 32 Bytes - BLS Private Key

Example

```
sign:  
curl -X POST -H "Content-Type:application/json" --data  
'{"jsonrpc":"2.0","method":"chain_signAddress","params":["0x5DEFDED18FE902F3F005A81D25955522416C6470",  
"0x[REDACTED]EC492FE0993CDE99D32CA860FDEF4E6E925243FBA89CBA[REDACTED"],"id":1}' http://localhost:6969/pchain
```

Returns

DATA, 64 Bytes - the BLS Signature for the Address

```
result:  
{"jsonrpc":"2.0","id":1,"result":"0x[REDACTED]176a210bf9bacd5b0211f4987ee3680d6e12b[REDACTED]22a458df01b3615b52825f32[REDACTED]b851048b4e67aeacadee[REDACTED]da9[REDACTED]"}  
[REDACTED]
```

Revealvote:

https://github.com/pchain-org/pchain/wiki/JSON-RPC#tdm_revealvote

Reveal the vote, the content of vote should matched with the hash which provided in tdm_voteNextEpoch (The valid window for this vote is between 85% - 95% of current epoch time, to know the detail block height, use tdm_getEpoch)

Parameters

- from: address, 20 Bytes - the address who triggers the action

- voteHash: hex string, 32 Bytes - hash of the vote. (How to get the vote hash, use Keccak-256 (not the standardized SHA3-256) example: keccak256(from + pubkey + amount + salt))
- gasPrice: hex string - (optional, default: 1 gwei) gas price from the request

Example

```
revealvote:
curl -X POST -H "Content-Type:application/json" --data
'{"jsonrpc":"2.0","method":"tdm_revealvote","params":["0x5DEFDED18FE902F3F005A91D25955522416C6470",
"3CA1601E6DBED9C4F2EC6513B9CA278EAF6F44093ED44802128644D901C76859356ED223AC3E1DFA0C1DB8D175DFDEF104D96EBE3C103E7706D023D71A0F96833002CD
C3F9A94F5C5210E4A9275A5FB953EFC77803C3D736E98F4550E8D9DB4E41C40CB387A83A360B23E322051928211", "0x2a5a058fc295ec000000", "PCHAIN",
"0x"376a210bf9bacd5b0211f4987e3680d6e12b6a80bc607c476122a458df01b3615b52825f3214430b851048b4e67aeacadeda9a15"], "id":1}''
http://localhost:6969/pchain
```

Returns

hash, string - the transaction hash

- 7) Check if you have competed successfully to be Validator based on the vote for the next Epoch.

https://github.com/pchain-org/pchain/wiki/JSON-RPC#tdm_getnextepochvalidators

tdm_getNextEpochValidators

Returns the validators of the next epoch based on the votes

Parameters

none

Returns

result , DATA - validators of the next epoch, such as address, public key, voting power

