How to join BNB Smart Chain Mainnet as Validator?

*Before you start, make sure you meet the hardware requirements for the validators nodes.

1.Choose Your Validator hardware:

- VPS running recent versions of Mac OS X or Linux.
- IMPORTANT 2T GB of free disk space, solid-state drive(SSD), gp3, 8k IOPS, 250MB/S throughput, read latency <1ms
- 16 cores of CPU and 48 gigabytes of memory (RAM)
- Suggest m5zn.3xlarge instance type on AWS, or c2-standard-8 on Google cloud.
- A broadband Internet connection with upload/download speeds of 10 megabyte per second.

2.Setting up Validator Node on Mainnet:

- Install BSC Fullnode

You can download the pre-build binaries from release page or follow the instructions here to set up a full node.
Download the config files:
Download genesis.json and config.toml by:

```
## mainet
wget --no-check-certificate  $(curl -s https://api.github.com/repos/bnb-chain/bsc/releases/latest |grep browser_ |grep mainnet |cut -d\" -f4)
unzip mainnet.zip
```

3. Write genesis state locally:

```
geth --datadir node init genesis.json
```

You could see the following output:

```
INFO [05-19|14:53:17.468] Allocated cache and file handles         database=/Users/huangsuyu/Downloads/bsc/node/geth/chaindata cache=16.00MiB handles=16
INFO [05-19|14:53:17.498] Writing custom genesis block
INFO [05-19|14:53:17.501] Persisted trie from memory database
nodes=21 size=56.84KiB time=357.915µs gcnodes=0 gcsize=0.00B
gctime=0s livenodes=1 livesize=-574.00B
INFO [05-19|14:53:17.502] Successfully wrote genesis state
database=chaindata hash=7d79cc…fb0d1e
INFO [05-19|14:53:17.503] Allocated cache and file handles         database=/Users/huangsuyu/Downloads/bsc/node/geth/lightchaindata cache=16.00MiB handles=16
INFO [05-19l14:53:17.524] Writing custom genesis block
INFO [05-19l14:53:17.525] Persisted trie from memory database
nodes=21 size=56.84KiB time=638.396µs gcnodes=0 gcsize=0.00B
gctime=0s livenodes=1 livesize=-574.00B
INFO [05-19l14:53:17.528] Successfully wrote genesis state
database=lightchaindata hash=7d79cc…fb0d1e
```

4. Create Consensus Key:

- You need to create an account that represents a validator's consensus key. Use the following command to create a new account and set a password for that account:
-

```
geth account new --datadir ./node
```

This command will return the public address and the path to your private key. If you already have an account, use the seed phrase to recover it:

```
geth account import --datadir ./node
```

5. Start Validator Node:

```
## generate the consensus key and input the password
geth account new --datadir ./node
echo {your-password} > password.txt
geth --config ./config.toml --datadir ./node --syncmode snap -unlock {your-
validator-address} --password password.txt  --mine  --allow-insecure-unlock --
cache 18000
```

6. Wait for node to sync:

Your node should now be catching up with the network by replaying all the transactions from genesis and recreating the blockchain state locally. This will take a long time, so make sure you've set it up on a stable connection so you can leave while it syncs.
View the status of the network with https://bscscan.com/.
You can monitor the status from log: $HOME/node/bsc.log by default.
You've now successfully joined a network as a full node operator.

7. Declare Candidacy:

You can use bnbcli binary to sen create-validator transaction, thus you can declare your candidacy.

- Download bnbcli from GitHub

- Use bnbcli to create an account or recover an account, make sure the account get more than 10000 BNB.
- Make sure your bsc validator have already catched up.
- Command for create validator on mainnet:
-

```
bnbcli staking bsc-create-validator \
-side-cons-addr {your validator address} \
--side-fee-addr {your wallet address on BSC} \
--address-delegator {your wallet address on BC} \
--side-chain-id bsc \
--amount 10000000000:BNB \
--commission-rate {10000000 represent 10%} \
--commission-max-rate {20000000 represent 20%} \
--commission-max-change-rate {500000000 represent 5%} \
--moniker {validator name} \
--details {validator detailed description} \
--identity {keybase identity} \
--website {website for validator} \
--from {key name} \
--chain-id Binance-Chain-Tigris \
--node https://dataseed5.defibit.io:443
```

After Declaring Your Candidacy:

1. Monitor node status:

- Run GethExporter in a Docker container.

```
docker run -it -d -p 9090:9090 \
  -e "GETH=http://mygethserverhere.com:8545" \
  hunterlong/gethexporter
```

2. Update validator profile:

- You can submit a PullRequest to this repository to update your information: https://github.com/bnb-chain/validator-directory
- Reference: https://grafana.com/grafana/dashboards/6976
-

3. Publish Validator Information:

- Please submit a Pull Request to this repo https://github.com/bnb-chain/validator-directory
-

4. Stop Validating:

- You can stop mining new blocks by sending commands in geth console

Connect to your validator node with geth attach ipc:path/to/geth.ipc

```
miner.stop()
```

To resume validating,

```
miner.start()
```