**MAP Validator Node Setup**

Map Website: <https://www.maplabs.io/>

Bridge: <https://bridge.maplabs.io/>

Explorer: <https://makalu.mapscan.io/>  
Staking Portal : <https://staking.maplabs.io/>

Mainnet Medium: <https://medium.com/marcopolo-protocol/map-mainnet-goes-live-on-august-31-4d3b044fcd8c>

**About MAP Protocol**

MAP Protocol is the omnichain Network for An Interoperable Web3 that enables developers to build omnichain dApps in simple steps. MAP Protocol lets cryptocurrency, NFT, and data flow around all chains securely and seamlessly at minimum cost. MAP Protocol is the only infrastructure made for omnichain dApps that connects all chains, provides security finality, charges the lowest cross-chain gas fees, and offers complete dApp development service toolkits.

**PREREQUISITES:**

* **Server with: 16 GB of Ram**, **a Quad core 2.5 GHz (64-bit) CPU, 256GB SSD and a 100Mb/s Ethernet connection w/ fiber Internet (ideally redundant connection and HA switches)**
* [**Ubuntu 20.04 LTS**](https://releases.ubuntu.com/20.04/) +
* **You need to have at least 1,000,000 Locked MAP to register a validator and a little bit for gas fees to create an account on the blockchain.**

**Update Linux**

To start we need to ensure that the server is up to date and has all the dependencies required to operate the node.

This guide uses ***Ubuntu 20.04.5 LTS***

First thing is to add update and upgrade the Ubuntu system.

sudo apt update && apt upgrade -y

**Setup User and Firewall**

See Hank the Cranks guide here:

<https://medium.com/@hankthecrank/9a90dd850edd>

Manual Setup from Hank (no tool)

<https://medium.com/@htctimbo/1cfce50d9d2e>

**Setup Server with prerequisites**

Next we need to ensure the server has all the dependencies required to operate the node.

\*\*Building atlas requires \*\*

* Git
* Go (version 1.14 or later)
* C compiler
* Python (optional to use the tool)

apt install git

apt install python3

apt install python3-pip

apt install gcc

apt install make

**Download Go language and do checksum (find latest version at**[**https://go.dev/dl/**](https://go.dev/dl/)**):**

curl -OL <https://go.dev/dl/go1.19.linux-amd64.tar.gz>

**Untar and then remove the tar:**

tar -C /usr/local -xvf go1.19.linux-amd64.tar.gz  
rm go1.19.linux-amd64.tar.gz

**Set permanent path in profile:**

nano ~/.profile

**Add the following lines at the end of the file:**

export PATH=$PATH:/usr/local/go/bin  
export GOROOT=/usr/local/go  
export GOPATH=$HOME/atlas  
export PATH=$GOPATH/bin:$GOROOT/bin:$PATH

CTRL-X to quit, press Y and ENTER to confirm

**Refresh profile:**

source ~/.profile

**Clone atlas:**

git clone <https://github.com/mapprotocol/atlas.git> && cd atlas

**Compile atlas:**

make atlas

**Compile marker:**

make marker

**Download Tool**

git clone <https://github.com/johnashu/pymap.git>

**Install Requirements**

pip3 install -r requirements.txt

**Create .Env file**

Edit example.env file and save it in the pymap dir as .env

nano /pymap/example.env

Ctrl+X to save but rename as .env

Graphical user interface, application, website

Description automatically generated

Type ‘y’ then edit the name to .env

A screenshot of a computer

Description automatically generated with medium confidence >>> A screenshot of a computer

Description automatically generated with medium confidence

**Start Toolbox**

Start the toolbox 0x48169c38072cB8Ff20FeBfA7Cc6bBAb1Fb900ecD

cd pymap  
python3 start\_tool.py

**Create Keystores for Account.**

**NOTE:** All Private Keys, addresses and passwords are throwaway accounts and are only for the purpose of this guide. Please NEVER EVER give anyone your private key, keystore file or password. This information should be treated like any other password, pin or seed phrase. Only YOU need to know this information.

Command to Process:

<Path To Atlas Binary> --datadir <Path to DataDir> account new

**Text

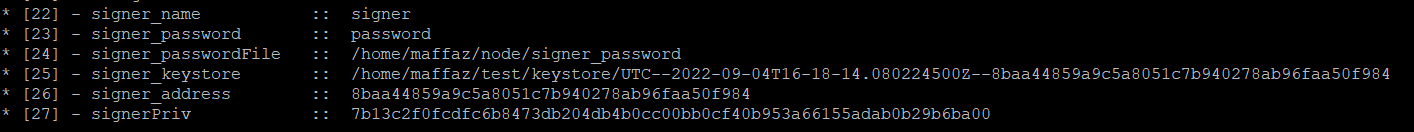
Description automatically generated**

**Create Keystores for Signer**

Creating an account for the Signer and selecting ‘yes’ for ‘isSigner’ the tool will save the pk, keystore location, password, passwordFile and address under the `signer\_` section of the .env.

Text

Description automatically generated

****Be sure to backup the keystore files and passwords given somewhere safe and not on the server.

**Buy Some MAP!!**

**# Buy MAP**

Map is available on several chains and can be purchased with the contract addresses provided.

\* Ethereum :

Uniswap : <https://app.uniswap.org/#/swap?outputCurrency=0x9E976F211daea0D652912AB99b0Dc21a7fD728e4>

Kucoin : <https://www.kucoin.com/trade/MAP-USDT>

> 0x9E976F211daea0D652912AB99b0Dc21a7fD728e4

​

\* BNB Chain -

Pancake Swap: <https://pancakeswap.finance/swap?outputCurrency=0x8105ECe4ce08B6B6449539A5db23e23b973DfA8f>

> 0x8105ECe4ce08B6B6449539A5db23e23b973DfA8f

\* Polygon:  
Uniswap : <https://app.uniswap.org/#/swap?outputCurrency=0xBAbceE78586d3e9E80E0d69601A17f983663Ba6a&chain=polygon>

Quickswap : <https://quickswap.exchange/>

> 0xBAbceE78586d3e9E80E0d69601A17f983663Ba6a

**# Bridge MAP to MAP protocol**

Goto the MAP bridge and send the MAP from Eth / Matic / BNB -> MAP Address created above.

> https://bridge.maplabs.io/

1. Select the chain from and the amount of map to bridge

2. Approve contract

3. bridge assets

4. Add MAP protocol to Metamask by changing the 'From' Protocol to 'MAP'.  This will trigger auto adding the network to Metamask

You should now be able to see you Map in Metamask and also by selecting 'Get Balance' in the Tool.

****

**Create Validator Account**

Now we have some MAP in our validator keystore, we can create a validator account that will be able to participate in consensus once registered.

**NOTE:** --namePrefix is –-name on mainnet but the tool will change if required.

Command to Process:

        /home/maffaz/atlas/build/bin/marker createAccount --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/test/keystore/UTC--2022-09-04T16-14-39.335346423Z--72e8cad69a440a4082428f57c038ad393289d6fd --namePrefix validator

Text

Description automatically generated

**Authorise Signer**

The Signer is responsible for signing blocks in the node. We need to Authorise an address to sign consensus messages on behalf of the account. This authorised address is called the signer. This address is only responsible for signing, your reward will not be issued to the signer, but to the Validator account created in the previous step.

Command to Process:

        /home/maffaz/atlas/build/bin/marker authorizeValidatorSigner --signerPriv 7b13c2f0fcdfc6b8473db204db4b0cc00bb0cf40b953a66155adab0b29b6ba00 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/test/keystore/UTC--2022-09-04T16-14-39.335346423Z--72e8cad69a440a4082428f57c038ad393289d6fd

**A computer screen capture

Description automatically generated with medium confidence**

**Start Node Syncing and Create Service File**

To start the node syncing with the chain we need to start atlas using the signer to start the process.

**NOTE:** If using testnet, be sure to add the --testnet flag to the end of the command. The tool will take this into consideration.

You can get the command and start running the service by choosing ‘Join Network (Sync) from the menu.

Command to Process:

        /home/maffaz/atlas/build/bin/atlas --datadir /home/maffaz/test --syncmode full --port 30321 --miner.validator 8baa44859a9c5a8051c7b940278ab96faa50f984 --unlock 8baa44859a9c5a8051c7b940278ab96faa50f984 --testnet --v5disc --mine

Text

Description automatically generated

**Create Systemd Service**

The best way to ensure that the node is consistently running is to create a system service to run in the background as a daemon and this will restart if there is an issue and automatically start on boot.

Select ‘Setup atlasNode Service ‘ and the tool will create and start this for you. If you make any changes, you can re-run the service creation and it will reload and restart the service.

A computer screen capture

Description automatically generated with low confidence

Then check the logs with ‘Show Node Logs’:

You should see the node starting to sync.

We will see the message ` err="not an elected validator 0x…”`

Text

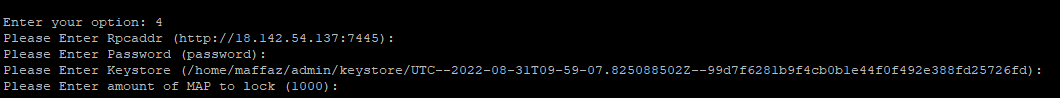
Description automatically generated

**LOCK MAP**

Now we have setup our accounts, keystore and are syncing the node, we need to lock some MAP ready tom register the node on the network.

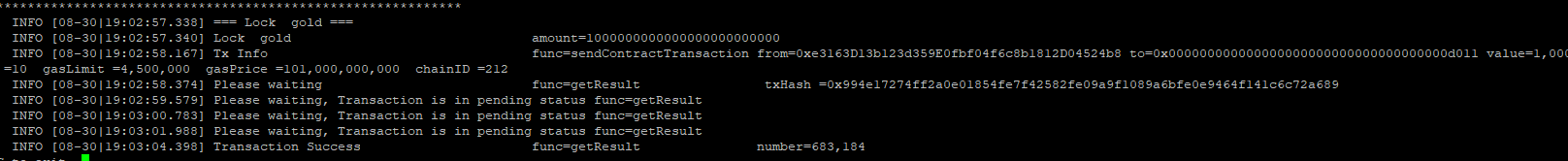
Locking Map makes the tokens available to vote (stake)

You need to Lock 1,000,000 MAP to be able to register as a validator.



Command to Process:

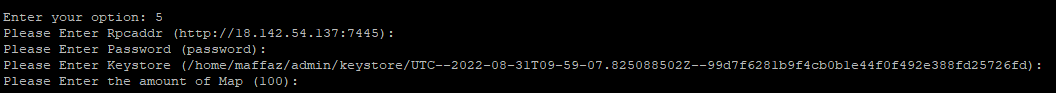
/home/maffaz/atlas/build/bin/marker lockedMAP --lockedNum 1000000 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/admin/keystore/UTC--2022-08-31T09-59-07.825088502Z--99d7f6281b9f4cb0b1e44f0f492e388fd25726fd



**Unlock Map**

If you wish to unlock MAP and take it out of consensus you can do the following.

The Unlock process will take 15 days.

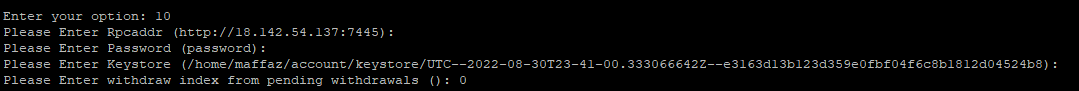


Command to Process:

        /home/maffaz/atlas/build/bin/marker unlockMap --mapValue 100 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/admin/keystore/UTC--2022-08-31T09-59-07.825088502Z--99d7f6281b9f4cb0b1e44f0f492e388fd25726fd

Withdraw MAP

This step will redeem the status of the reward from the unlocked state to the balance, but this step needs to be unlocked for 15 days before it can be executed.

****

Command to Process:

        /home/maffaz/atlas/build/bin/marker withdrawMap --withdrawIndex 0 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/node/keystore/UTC--2022-08-30T23-41-00.333066642Z--e3163d13b123d359e0fbf04f6c8b1812d04524b8

**Register Validator on the Network**

Register the node on the network as a validator having the required 1,000,000 MAP locked in the account.

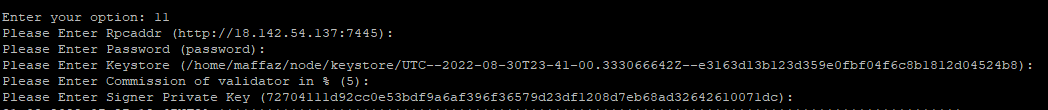
Once registered, the node is eligible to receive votes and participate in consensus if elected.

Note that commission is 10000 for the cli

So 5% is 5 \* 10000 = 50000

Reward will be distributed every epoch to all the online validators according to the uptime. Once a validator receives rewards, it will take a portion of it as commission and then it will split the rest of it to all of its voters.

Rewards will be in the form of Locked Map and will be auto-compounded as Votes



Command to Process:

        /home/maffaz/atlas/build/bin/marker register --commission 50000 --signerPriv 72704111d92cc0e53bdf9a6af396f36579d23df1208d7eb68ad32642610071dc --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/node/keystore/UTC--2022-08-30T23-41-00.333066642Z--e3163d13b123d359e0fbf04f6c8b1812d04524b8

Graphical user interface

Description automatically generated

We can check our locked maps using the command *getAccountTotalLockedGold*

Graphical user interface, text

Description automatically generated

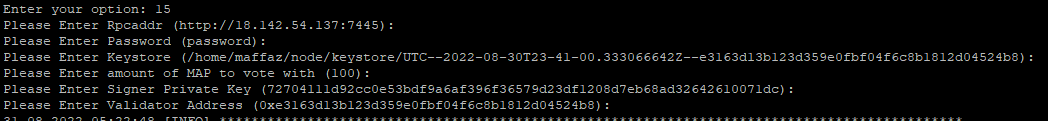
**Voting**

Voting is where we ‘stake’ our map to a validator to enable to earn rewards and participate in consensus.

Once MAP is locked, it is available for voting (staking). Validators must have at least 0.001 proportion of the total votes to be considered for the election. So the validator can't have no votes.

We can use our validator account to vote for ourselves, or we can let other validators or voters vote for ourselves.

Once MAP is allocated as votes to a validator, the votes will be automatically activated at the next epoch.

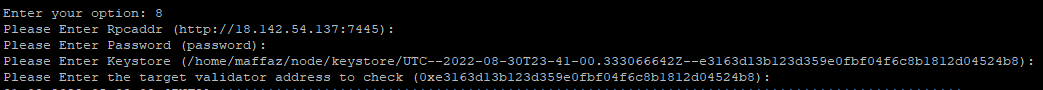


Command to Process:

        /home/maffaz/atlas/build/bin/marker vote --voteNum 100 --signerPriv 72704111d92cc0e53bdf9a6af396f36579d23df1208d7eb68ad32642610071dc --validator 0xe3163d13b123d359e0fbf04f6c8b1812d04524b8 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/node/keystore/UTC--2022-08-30T23-41-00.333066642Z--e3163d13b123d359e0fbf04f6c8b1812d04524b8

**Check Votes by Validator**

To check the number of votes you have given for a validator use *getActiveVotesForValidatorByAccount*

****

Command to Process:

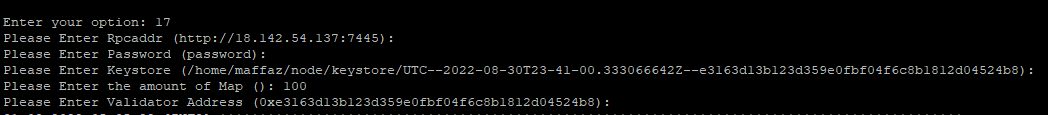
        /home/maffaz/atlas/build/bin/marker getActiveVotesForValidatorByAccount --target 0xe3163d13b123d359e0fbf04f6c8b1812d04524b8 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/node/keystore/UTC--2022-08-30T23-41-00.333066642Z--e3163d13b123d359e0fbf04f6c8b1812d04524b8

**Revoke Pending Votes**

If you wish to take the votes out of the validator before they are activated you can use the `revokePending`command.

Revokes value pending votes for validator.

This command will put the voting MAP turn into nonvoting MAP.



Command to Process:

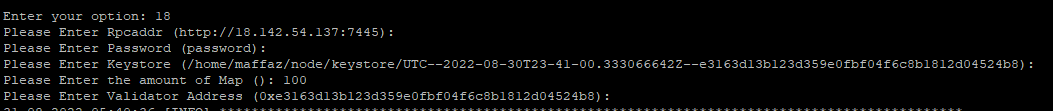
        /home/maffaz/atlas/build/bin/marker revokePending --mapValue 100 --validator 0xe3163d13b123d359e0fbf04f6c8b1812d04524b8 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/node/keystore/UTC--2022-08-30T23-41-00.333066642Z--e3163d13b123d359e0fbf04f6c8b1812d04524b8

**Revoke Active Votes**

If you wish to take the votes out of the validator before they are activated you can use the ` revokeActive` command.

Revokes Active votes for validator.

This command will put the voting MAP turn into nonvoting MAP.



Command to Process:

        /home/maffaz/atlas/build/bin/marker revokeActive --mapValue 100 --validator 0xe3163d13b123d359e0fbf04f6c8b1812d04524b8 --rpcaddr http://18.142.54.137:7445 --password password --keystore /home/maffaz/node/keystore/UTC--2022-08-30T23-41-00.333066642Z--e3163d13b123d359e0fbf04f6c8b1812d04524b8