**Akashichain Node Setup Command Manual**

**Observation Node Setup**

1. Create a User  
   If the current logged-in user is the root account, it is recommended to create an akashichain account and switch to it. If the current logged-in user is not a root account, skip this step.

/> useradd -m -s /bin/bash akashichain

/> su akashichain

(This document uses the akashichain account as an example.)

1. Create a Working Directory

/> cd /opt

/> sudo mkdir akashichain

/> sudo chown -R akashichain:akashichain akashichain

/> cd akashichain

/> mkdir /opt/akashichain/bin /opt/akashichain/data

1. Download Executable Files

/> cd /opt/akashichain/bin/

/> wget https://github.com/akashichain-projects/akclayer/releases/download/v0.0.2/akashichain.tar.gz -O /opt/akashichain/bin/akashichain.tar.gz

/> tar zvfx akashichain.tar.gz

/> chmod a+x /opt/akashichain/bin/akashicored

1. Initialize the Chain

/> cd /opt/akashichain

/> bin/akashicored init --chain-id akashichain\_9071-1 --home=./data

Mainnet: akashichain\_9070-1; Testnet: akashichain\_9071-1; Mocknet: akashichain\_19070-1.  
Replace MONIKER with the name of the node, which can only consist of letters and numbers.

1. Download Configuration Files

/> wget https://raw.githubusercontent.com/akashichain-projects/network-config/main/serenity/genesis.json -O ./data/config/genesis.json

/> wget https://raw.githubusercontent.com/akashichain-projects/network-config/main/serenity/client.toml -O ./data/config/client.toml

/> wget https://raw.githubusercontent.com/akashichain-projects/network-config/main/serenity/config.toml -O ./data/config/config.toml

/> wget https://raw.githubusercontent.com/akashichain-projects/network-config/main/serenity/app.toml -O ./data/config/app.toml

1. Set Limits for Open Files and Processes  
   Edit the /etc/security/limits.conf file:

/> sudo vi /etc/security/limits.conf

Add the following lines to the end of the file:

\* soft nproc 262144

\* hard nproc 262144

\* soft nofile 262144

\* hard nofile 262144

Edit the /etc/sysctl.conf file:

/> sudo vi /etc/sysctl.conf

Add the following line to the end of the file:

fs.file-max=262144

1. Configure Auto-Start Service  
   Edit the service configuration file:

/> sudo vi /etc/systemd/system/akashicored.service

Add the following content:

[Unit]

Description=akashicored

After=multi-user.target

StartLimitIntervalSec=0

[Install]

WantedBy=multi-user.target

[Service]

WorkingDirectory=/opt/akashichain

Type=simple

Restart=always

RestartSec=600

User=akashichain

LimitNOFILE=262144

ExecStart=/opt/akashichain/bin/akashicored start --home=/opt/akashichain/data/

Reload the service and enable it:

/> sudo systemctl daemon-reload

/> sudo systemctl enable --now akashicored

1. Synchronize Data  
   Check the status:

/> curl localhost:26657/status

If "catching\_up":false, synchronization is complete.  
If "catching\_up":true, synchronization is still in progress.

**Becoming a Validator**

1. Validator with EIC Assets

9.1 Create a Validator Account  
Example of creating a validator account named operator:

/> bin/akashicored keys add operator --algo eth\_secp256k1 --home=./data

9.2 Transfer Funds to the Validator

/> bin/akashicored tx bank send <from address> <to address> <amount> --fees=3000000000000000aakc --home=./data

Example of transferring 500 AKC:

/> bin/akashicored tx bank send aksha1qpvcrrlg6nucku5zxm88lxfa0cfnms99elcu5 akasha197s5jdgq0tpra3gzypcqp4de3xjfz5sltujkps 500000000000000000000aakc --fees=3000000000000000aakc --home=./data

Check the balance:

/> bin/akashicored query bank balances akasha197s5jdgq0tpra3gzypcqp4de3xjfz5sltujkps --home=./data

9.3 Become a Validator

/> bin/akashicored tx staking create-validator \

--amount=<stake amount> \

--pubkey=$(bin/akashicored tendermint show-validator --home=./data) \

--moniker="MY\_NODE\_NAME" \

--chain-id=<chain\_id> \

--commission-rate="0.10" \

--commission-max-rate="0.20" \

--commission-max-change-rate="0.01" \

--min-self-delegation="1000000" \

--gas="600000" \

--fees=<fee> \

--from=<validator address> \

--home=./data

Example:

/> bin/akashicored tx staking create-validator \

--amount=10000000000000000000000aakc \

--pubkey=$(bin/akashicored tendermint show-validator --home=./data) \

--moniker="akashichain" \

--chain-id=akashichain\_9070-1 \

--commission-rate="0.10" \

--commission-max-rate="0.20" \

--commission-max-change-rate="0.01" \

--min-self-delegation="1000000" \

--gas="600000" \

--fees=6000000000000000aakc \

--from=akasha197s5jdgq0tpra3gzypcqp4de3xjfz5sltujkps \

--home=./data

9.4 Check Validator Status  
Verify if the validator creation transaction succeeded:

/> bin/akashicored query tx --home=./data

Check if the validator is created:

/> bin/akashicored query staking validator $(bin/akashicored keys show <validator address> --bech val -a --home=./data) --home=./data

Check if the node is in the validator list:

/> bin/akashicored query tendermint-validator-set | grep $(bin/akashicored tendermint show-address --home=./data) --home=./data

Check validator signing status:

/> bin/akashicored query slashing signing-info $(bin/akashicored tendermint show-validator --home=./data) --home=./data

9.5 Unjail the Node  
If the node is jailed, execute the following command to unjail:

/> bin/akashicored tx slashing unjail --from=<validator address> --home=./data

9.6 Delegate EIC to the Node

/> bin/akashicored tx staking delegate $(bin/akashicored keys show operator --bech val -a --home=./data) <amount> --from=<your address> --home=./data