



# Prime Stake Pool Operator (SPO) Technical Guide

## Disclaimer

*There is a certain level of technical proficiency required for a stake pool operator. Cyber security, system administrating and devops skills are crucial. Bad operations may lead to missed funds or even theft of funds, as well as network instability. HAL8 and/or Apex Fusion Foundation is not responsible and can not help in such cases, responsibility is completely on the side of a stake pool operator.*

## Introduction

Find the latest version of Prime with the accompanying docs on [Prime GitHub repo](#).

This guide provides comprehensive instructions for setting up, configuring, and maintaining a stake pool on the Prime blockchain network. While the core steps remain consistent with Cardano, ensure you have the correct `genesis.json` file for the Prime network. Let's get started!

## References

This guide re-uses previous guides created by a large and thriving Cardano SPO community and we wish to thank them for their efforts. There are many dependencies referenced in this documentation and we apologize for not listing all of them. If you find an open-source guide or tool used here and not referenced, let us know and we will add it to the list. Please check out original guides if you need more details:

1. [Coin Cashew](#)
2. [Stakepool 24/7 EU](#)
3. [Cardano Foundation](#)
4. [IOHK](#)

## Setting Up Your Environment

Ensure you have the required tools and libraries installed, such as GHC, Cabal, Libsodium, and the Cardano CLI. Follow the detailed steps in the Prime documentation or the respective installation guides.

## Starting the Nodes

### Block Producer Node and Relay Node

```
`sudo systemctl start cardano-node`
```

Ensure your nodes are running and begin syncing the blockchain.

### Monitoring Node Status with gLiveView

Install and set up gLiveView to monitor your node status.

```
cd $NODE_HOME
sudo apt install bc tcptraceroute -y
curl -s -o gLiveView.sh https://raw.githubusercontent.com/cardano-comm
curl -s -o env https://raw.githubusercontent.com/cardano-community/gu
chmod 755 gLiveView.sh
```

```
sed -i env -e "s/\#CONFIG=\"\${CNODE_HOME}\files/config.json\"/CONF
./gLiveView.sh
```

## Generating Keys for the Block-producing Node

### Create KES Key Pair

```
cd $NODE_HOME cardano-cli node key-gen-KES --verification-key-file ke
```

### Create Cold Keys

On an air-gapped offline machine:

```
mkdir $HOME/cold-keys cd $HOME/cold-keys cardano-cli node key-gen --c
```

## Setting Up Payment and Stake Keys

### Create Payment Key Pair

On an air-gapped offline machine:

```
cd $NODE_HOME cardano-cli address key-gen --verification-key-file pay
```

### Create Stake Address Key Pair

On an air-gapped offline machine:

```
cardano-cli stake-address key-gen --verification-key-file stake.vkey
```

## Build Stake Address

```
cardano-cli stake-address build --stake-verification-key-file stake.v
```

## Build Payment Address

```
cardano-cli address build --payment-verification-key-file payment.vke
```

## Registering Your Stake Address

### Create Stake Address Registration Certificate

On an air-gapped offline machine:

```
cardano-cli stake-address registration-certificate --stake-verificati
```

## Build and Submit Transaction

Follow similar steps as outlined in "Setting Up Payment and Stake Keys" to build and submit the transaction, ensuring to include the `stake.cert`.

## Registering Your Stake Pool

### Step 1: Create Pool Metadata

```
cat > $NODE_HOME/poolMetadata.json << EOF {  "name": "MyPoolName",
```

### Step 2: Upload Metadata

Upload `poolMetadata.json` to a web server.

## Step 3: Create Registration Certificate

On the air-gapped computer:

```
cardano-cli stake-pool registration-certificate --cold-verification-k
```

## Step 4: Submit the Registration Certificate

Follow similar steps as outlined in "Setting Up Payment and Stake Keys" to build and submit the transaction, ensuring to include the `pool.cert`.

# Verifying Stake Pool Operation

## Step 1: Compute Stake Pool ID

On the air-gapped computer:

```
cardano-cli stake-pool id --cold-verification-key-file node.vkey --ou
```

## Step 2: Verify Registration

On the block producer node:

```
cardano-cli query stake-snapshot --stake-pool-id $(cat stakepoolid.tx
```

# Configuring Legacy Network Topology

## Step 1: Create Topology Updater Script

On ``relaynode1``:

```
cat > $NODE_HOME/topologyUpdater.sh << EOF #!/bin/bash USERNAME=$(who
```

## Step 2: Set Up Cron Job

```
crontab -l | { cat; echo "33 * * * * ${NODE_HOME}/topologyUpdater.sh"
```

# Setting Up Dashboards

## Step 1: Install Prometheus and Grafana

```
sudo apt-get install -y prometheus prometheus-node-exporter grafana s
```

## Step 2: Configure Prometheus

Edit ``/etc/prometheus/prometheus.yml``:

```
global:   scrape_interval: 15s   scrape_configs:   - job_name: 'promet
```

Restart the services:

```
sudo systemctl restart grafana-server.service prometheus.service prom
```

# Configuring Slot Leader Calculations

## Cardano-CLI Query

```
cardano-cli query leadership-schedule --mainnet --genesis $NODE_HOME/
```

## Securing Your Stake Pool Using a Hardware Wallet

### Step 1: Install cardano-hw-cli

```
cd $NODE_HOME wget https://github.com/vacuumlabs/cardano-hw-cli/releases
```

### Step 2: Generate Hardware Wallet Keys

```
cardano-hw-cli address key-gen --path 1852H/1815H/0H/2/0 --verification
```

### Step 3: Update Stake Pool Registration

Update the registration certificate to include the new hardware wallet keys.

## Checking Stake Pool Rewards

```
cardano-cli query stake-address-info --address $(cat stake.addr) --ma
```

## Claiming Stake Pool Rewards

### Step 1: Find Current Slot

```
currentSlot=$(cardano-cli query tip --mainnet | jq -r '.slot') echo C
```

## Step 2: Set Reward Balance

```
rewardBalance=$(cardano-cli query stake-address-info --mainnet --addr
```

## Step 3: Set Destination Address

```
destinationAddress=$(cat payment.addr) echo destinationAddress: $dest
```

## Step 4: Find Payment Address Balance

```
cardano-cli query utxo --address $(cat payment.addr) --mainnet > full
```

## Step 5: Build and Submit Transaction

Follow similar steps as outlined in "Setting Up Payment and Stake Keys" to build and submit the transaction.

# Delegating to a Stake Pool

## Step 1: Create Stake Address Registration Certificate

```
cardano-cli stake-address registration-certificate --stake-verificati
```

## Step 2: Build and Submit Transaction



Follow similar steps as outlined in "Setting Up Payment and Stake Keys" to build and submit the transaction, ensuring to include the ``stake.cert``.

### Step 3: Create Delegation Certificate

```
cardano-cli stake-address delegation-certificate --stake-verification
```

### Step 4: Build and Submit Transaction

Follow similar steps as outlined in "Setting Up Payment and Stake Keys" to build and submit the transaction, ensuring to include the

```
deleg.cert
```

## Issuing a New Operational Certificate

### Step 1: Determine Counter Value

```
cardano-cli query kes-period-info --mainnet --op-cert-file node.cert
```

### Step 2: Set Counter Value

On an air-gapped offline machine:

```
cd $HOME/cold-keys cardano-cli node new-counter --cold-verification-k
```

### Step 3: Generate KES Key Pair

On the block producer node:

```
cardano-cli node key-gen-KES --verification-key-file kes.vkey --signi
```

## Step 4: Calculate Starting KES Period

```
slotNo=$(cardano-cli query tip --mainnet | jq -r '.slot') slotsPerKES
```

## Step 5: Issue New Operational Certificate

On an air-gapped offline machine:

```
cardano-cli node issue-op-cert --kes-verification-key-file kes.vkey -
```

## Step 6: Restart Node

On the block producer node:

```
sudo systemctl restart cardano-node
```

# Updating Stake Pool Information

## Step 1: Generate Protocol Parameters

```
cardano-cli query protocol-parameters --mainnet --out-file $NODE_HOME
```

## Step 2: Create Updated Registration Certificate

On an air-gapped offline machine:

```
cardano-cli stake-pool registration-certificate --cold-verification-k
```

### Step 3: Submit the Updated Certificate

Follow similar steps as outlined in "Setting Up Payment and Stake Keys" to build and submit the transaction, ensuring to include the updated `pool.cert`.

## Upgrading a Node

### Step 1: Upgrade CNCLI and Guild LiveView

Follow the detailed instructions in the respective sections to upgrade CNCLI and Guild LiveView.

### Step 2: Set GHC and Cabal Versions

Ensure you have the required versions of GHC and Cabal installed for the latest Cardano node release.

### Step 3: Update Libsodium and secp256k1

Follow the detailed instructions in the respective sections to update Libsodium and secp256k1.

### Step 4: Install New Binaries and Configuration Files

Follow the detailed instructions in the respective sections to install the latest binaries and configuration files.

### Step 5: Restart Node

```
sudo systemctl restart cardano-node
```

# Retiring Your Stake Pool

## Step 1: Generate Protocol Parameters

```
cardano-cli query protocol-parameters --mainnet --out-file $NODE_HOME
```

## Step 2: Calculate Current Epoch

```
startTimeGenesis=$(cat $NODE_HOME/shelley-genesis.json | jq -r .system
```

## Step 3: Find Retirement Epoch

```
poolRetireMaxEpoch=$(cat $NODE_HOME/params.json | jq -r '.poolRetireM
```

## Step 4: Create Deregistration Certificate

On an air-gapped offline machine:

```
cardano-cli stake-pool deregistration-certificate --cold-verification
```

## Step 5: Submit Deregistration Certificate

Follow similar steps as outlined in "Setting Up Payment and Stake Keys" to build and submit the transaction, ensuring to include the `pool.dereg`.

# Auditing Your Nodes Configuration

## Step 1: Download Audit Script

```
cd $HOME/git git clone https://github.com/Kirael12/Cardano-Audit-Coin
```

## Step 2: Make Script Executable

```
cd $HOME/git/Cardano-Audit-Coincashew chmod +x audit-coincashew.sh
```

## Step 3: Run the Script

```
sudo -E ./audit-coincashew.sh
```

# KES Key Rotation / Operational Certificate Companion Script

## Step 1: Download Companion Script

```
cd $HOME/git git clone https://github.com/Kirael12/Cardano-KES-Rotate
```

## Step 2: Make Script Executable

```
cd $HOME/git/Cardano-KES-Rotate-Companion/ chmod +x rotateKES.sh
```

## Step 3: Run the Script

```
sudo -E ./rotateKES.sh
```

This guide provides detailed instructions for aspiring stake pool operators on the Prime network, leveraging the same codebase and tools as Cardano. Ensure you have the correct configuration files and always follow best practices for security and maintenance. Happy staking!

---

[Privacy policy](#) [Terms of service](#)

2025 Apex Fusion. All rights reserved.