CESS Network

The Decentralized Data Infrastructure

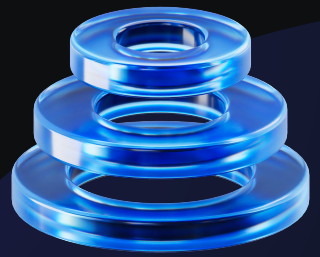
Episode 5
Demo: Running a Consensus Node

https://www.cess.network

Course Logistics

Course Website: https://course.cess.network/

Episode 1	CESS Network Introduction
Episode 2	CESS Architecture & Key Technologies
Episode 3	CESS Ecosystem and Applications
Episode 4	CESS Nodes & CESS Account Setup
Episode 5	Demo: Running a Consensus Node
Episode 6	Demo: Running a Storage Node
Episode 7	CESS DeOSS and DeOSS REST API
Episode 8	dApp Development using ink! Smart Contract
Episode 9	dApp Development using Solidity Smart Contract
Episode 10	Building Custom Pallet



System Requirements

CESS

Resource	Specification
Recommended OS	Ubuntu_x64 20.04 or higher
# CPU Processor	≥ 4
Intel SGX Enabled	Required with FLC
Memory (SGX encrypted memory)	≥ 16 GB
Bandwidth	≥ 5 Mbps
Public Network IP	Required
Linux Kernel Version	5.11 or higher

Prerequisites

CESS

Intel SGX with FLC

- Enable from BIOS
- Recommended CPU: Intel E, E3, Celeron
- Prefered CPU: Intel Core i5 10500
- Recommended Motherboard: Supermicro

Static Public IP

- curl -4 ifconfig.co

CESS Wallet Accounts

- Stash Account: 3Mil TCESS
- Controller Account: 100 TCESS

CESS Consensus Node Operational Capacity

CESS

1. Full Node

- Fully-capable consensus node with all necessary functions.
- Generates random challenges, verifies data, computes tags.
- Generates and replaces space holder data.
- Participates in network consensus.
- Requires binding to consensus nodes for registration.

2. Verifier Node

- Participates in network consensus.
- Handles random challenges for idle and service data.
- Requires binding to consensus nodes for registration.

Requirements

Full and Verifier Operational Capacities

Stash Account

- Keeps all the funds you want to stake.
- Requires at least 3,000,000 TCESS for staking.
- Can be funded by the node owner or delegated by other users.
- Bonds/unbond funds and designates the Controller Account.

Controller Account

- Pays gas fees for staking-related transactions and registration.
- Takes actions on behalf of the bonded funds in the Stash Account.
- Cannot move bonded funds out of the Stash Account.

CESS Consensus Node Operational Capacity

CESS

3. Marker Node

- Computes tags for user's service data.
- Creates, verifies, and replaces idle data segments.
- Serves a designated storage node cluster.
- Can be registered independently.
- Operating as a Marker does not increase reputation points.

Requirements

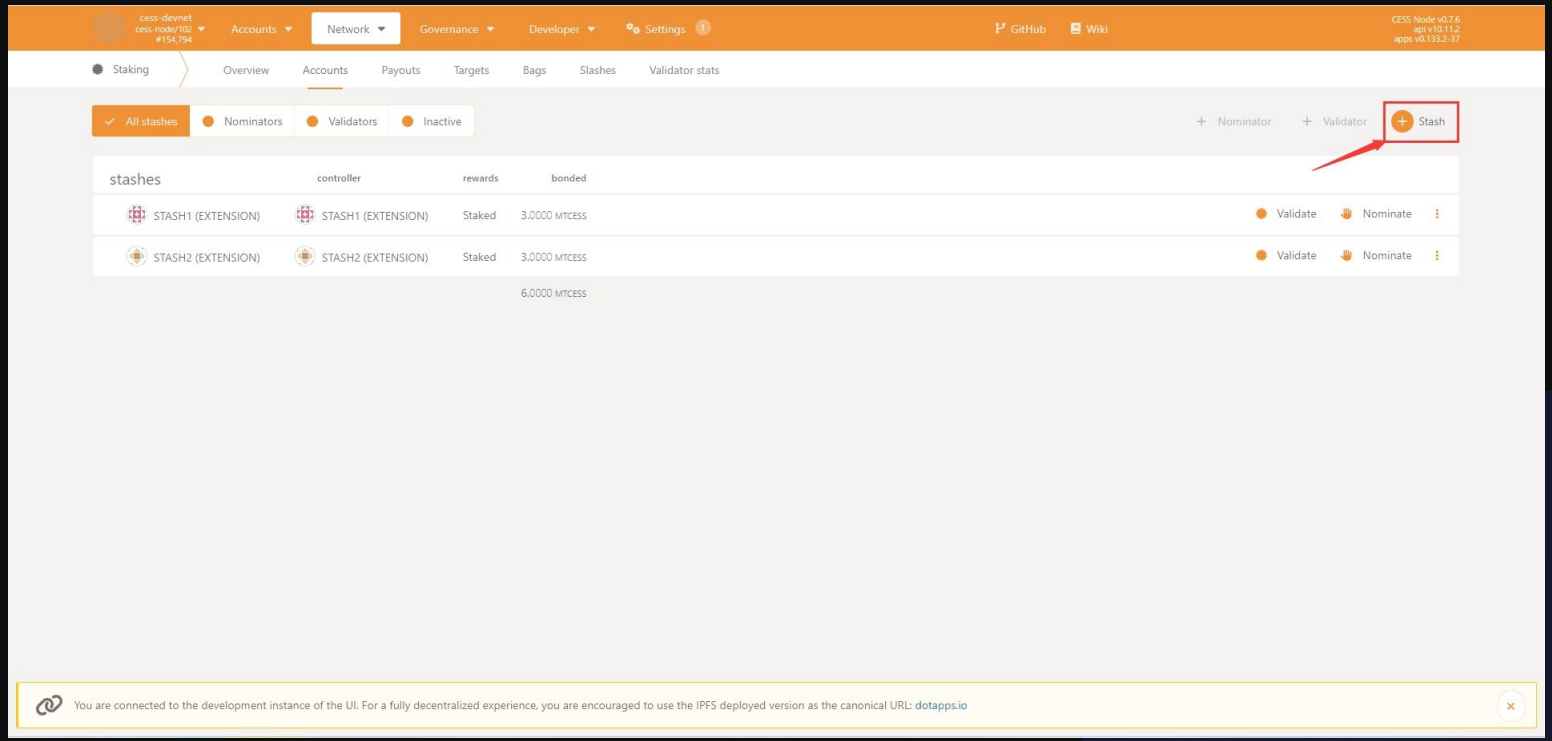
Marker Operational Capacity

- Requires only one account.
- Does not require Binding Funds operation.

Additional Notes: Using existing stash accounts with Bounded tokens or using another user's stash account bypasses the need for the Binding Funds operation.

Binding Funds

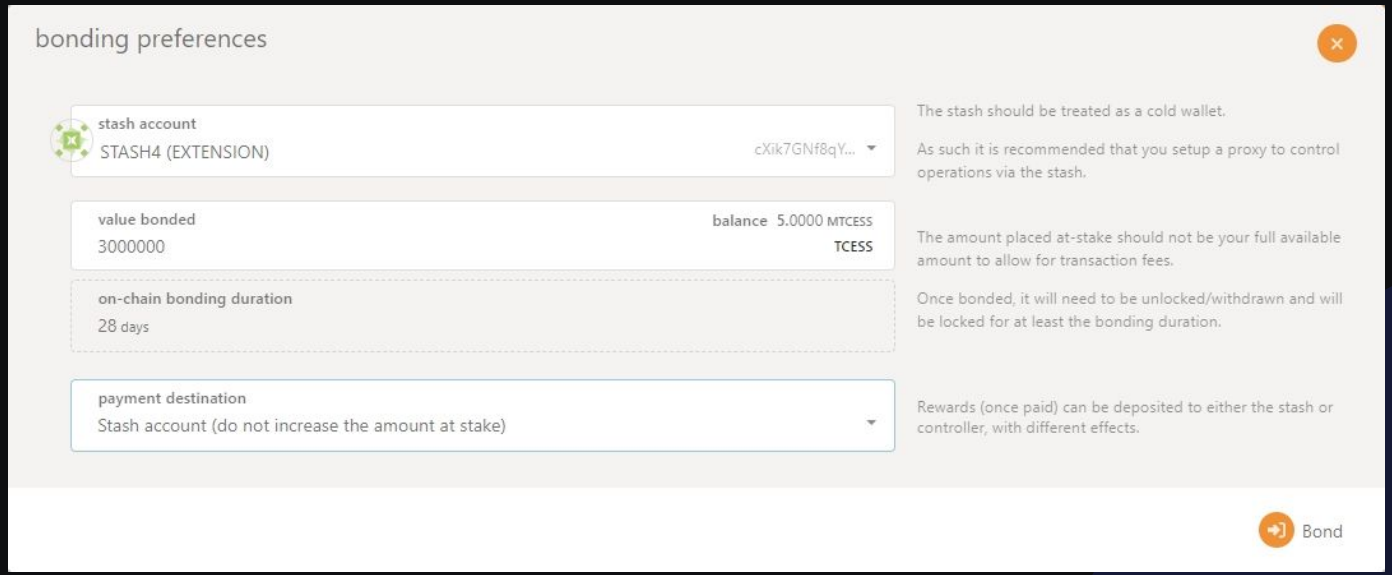
Step 1: Open CESS Explorer and Select Network > Staking > Accounts > Stash



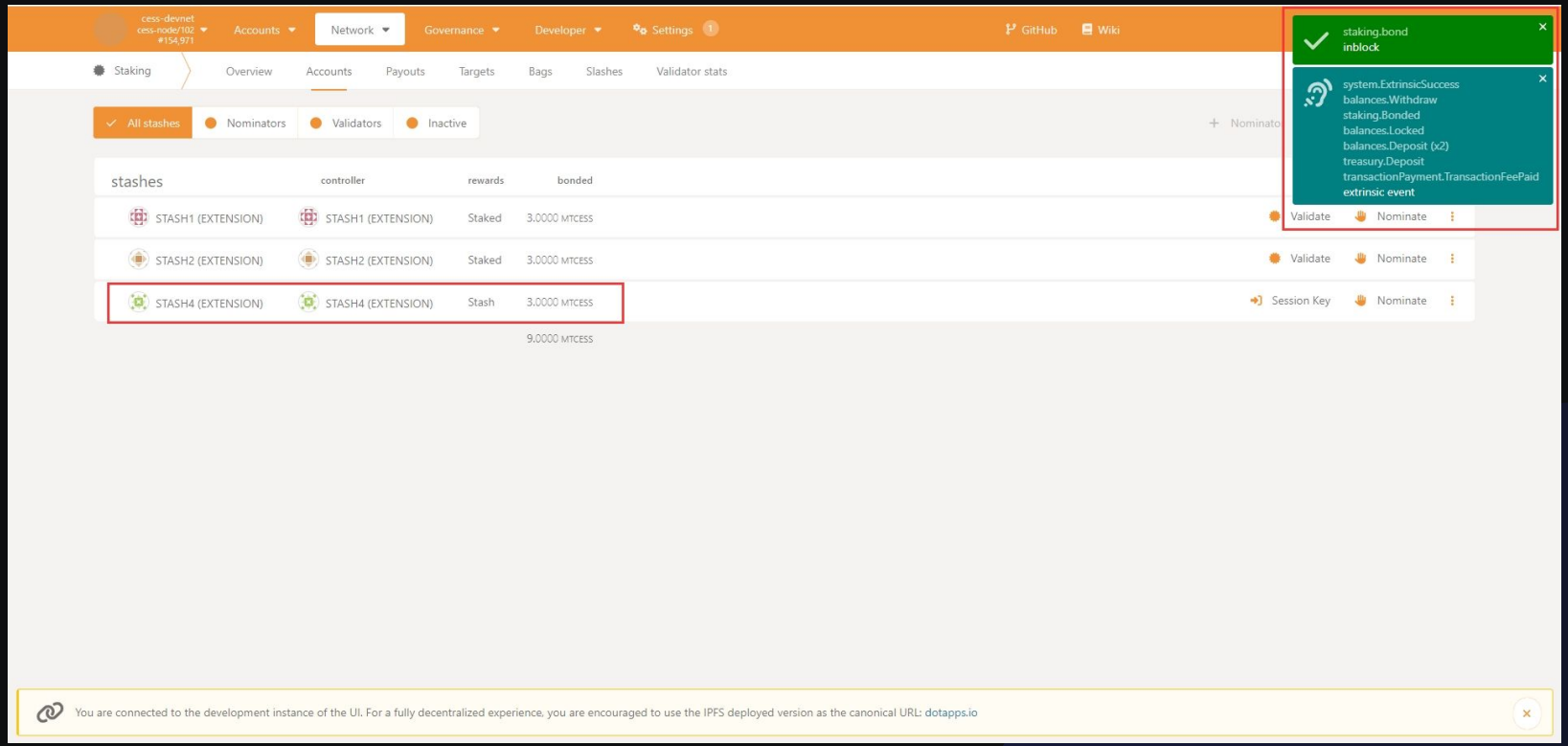
Binding Funds

Step 2:

1. Select appropriate “stash account”
2. Enter at least 3,000,000 TCESS in “value bounded”
3. Select “do not increase the amount at stake” from “payment destination”
4. Click Bond > Sign and Submit to link the Stash Account.



Binding Funds Success



Consensus Node Installation and Configuration

CESS

Install CESS Client: nodeadm

```
wget https://github.com/CESSProject/cess-nodeadm/archive/refs/tags/v0.5.5.tar.gz
tar -xvf v0.5.5.tar.gz
cd cess-nodeadm-0.5.5
sudo ./install.sh
```

Client Configuration for a Full node

sudo cess config set

```
Enter cess node mode from 'authority/storage/rpcnode' (current: authority, press enter to skip): authority
Enter cess node name (current: cess, press enter to skip):
Enter cess chain ws url (default: ws://cess-chain:9944):
Enter the public port for TEE worker (current: 19999, press enter to skip):
Enter the TEE worker endpoint (current: http://xx.xxx.xx.xx:19999, press enter to skip)
Enter cess validator stash account (current: null, press enter to skip):
Enter what kind of tee worker would you want to be [Full/Verifier]: Full
Enter cess validator controller phrase: xxxxxxxxxxxxxx
```

Consensus Node Set Up

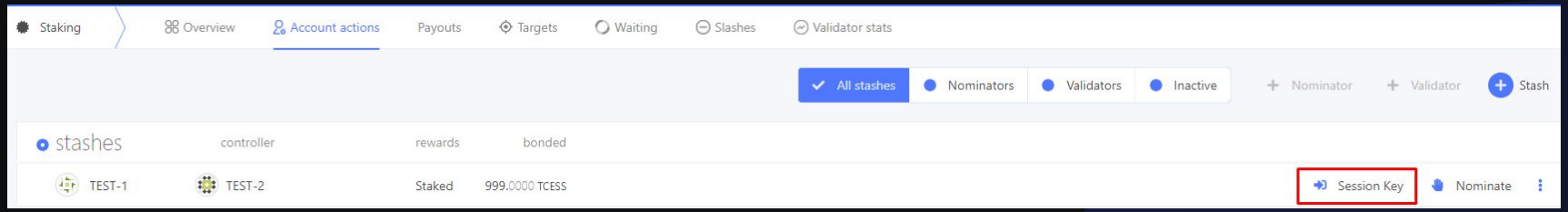
Setting up Consensus Node to Become a Validator

1. Start Consensus Node ---- cess start

2. Generate a session key ---- cess tools rotate-keys

```
root@ubuntu:~/cess-nodeadm-0.1.0#
root@ubuntu:~/cess-nodeadm-0.1.0# curl -H "Content-Type: application/json" -d '{"id":1, "jsonrpc":"2.0", "method": "author_rotateKeys", "params":[]}' http://localhost:9933
{"jsonrpc":"2.0","result":"0x70f0f4ef909c730e0057cfd6d6011f52d110[illegible]2eaa8ad5ad8a1c20b98f0e568e63a1310b95a210a619b7c5e29a477072c9ad811107eb944f3974070ac0433f1416e6ad41fd9f6208f089b964340
62743eae316a664426defed4e256b47b0405d8a064ac3b572e327f77d5a1dfe86f[illegible]9207643f117ebf1a","id":1}
root@ubuntu:~/cess-nodeadm-0.1.0#
```

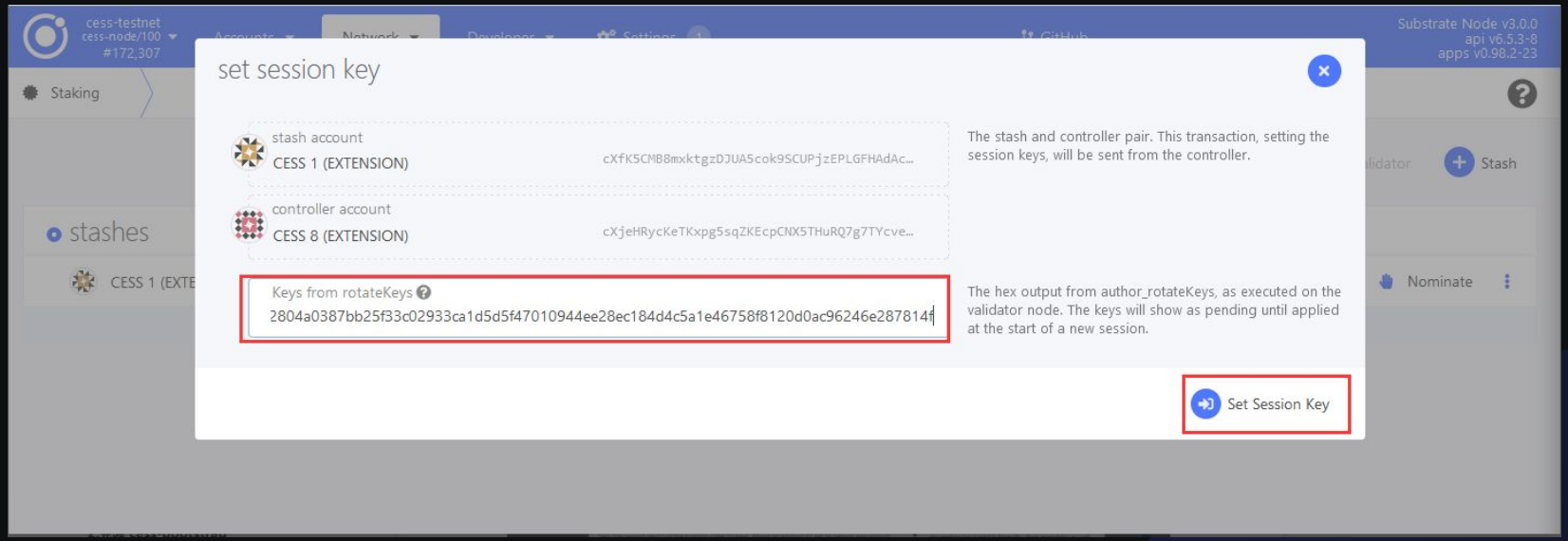
3. Setup a Session Key ---- Navigate to CESS Explorer, choose Network > Staking > Accounts > Session Key



Consensus Node Set Up

CESS

4. Fill in the Session key

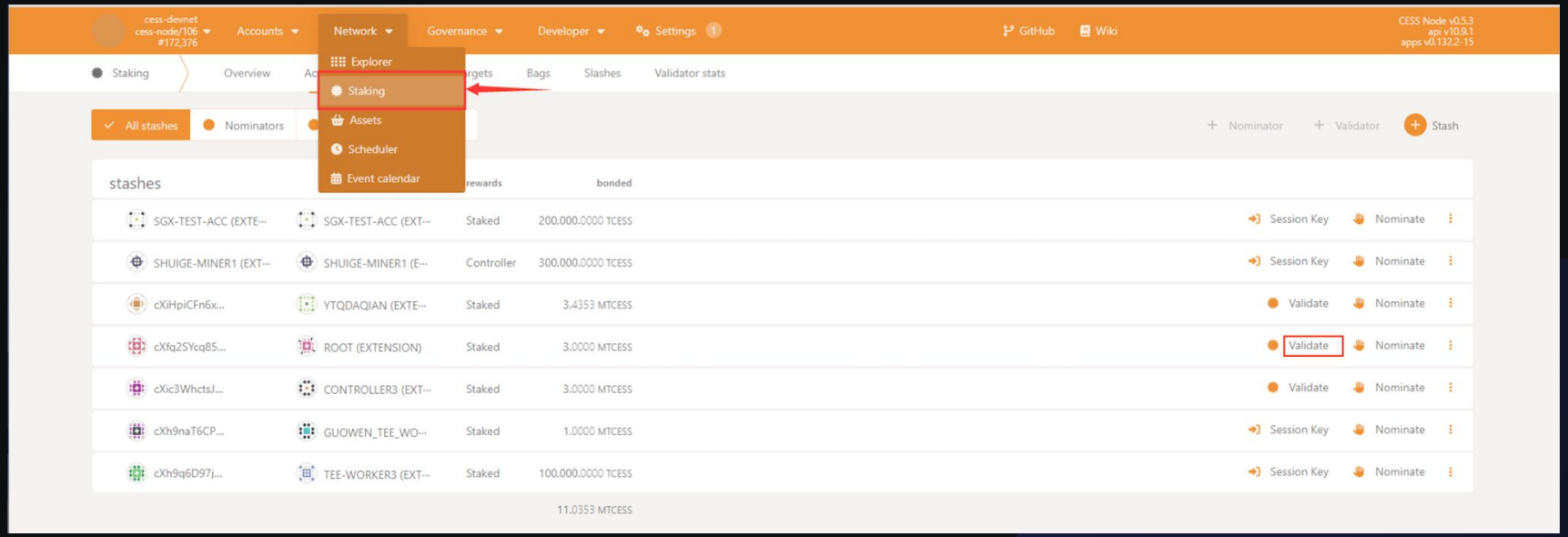


5. Sign and Submit the transaction

Consensus Node Set Up

Becoming a Validator

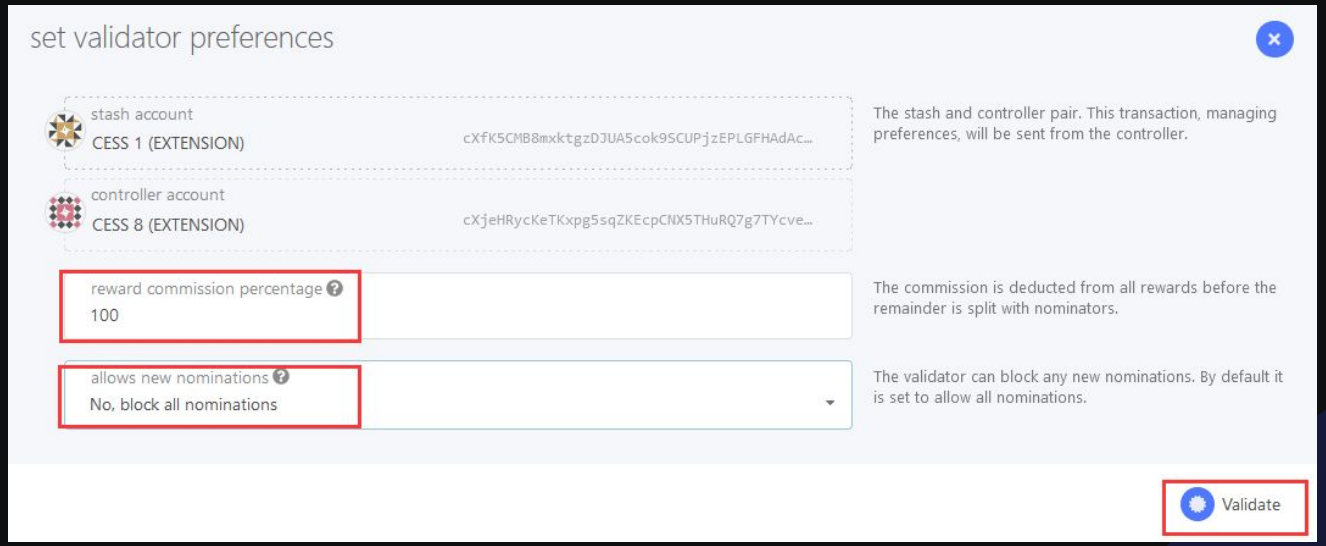
Step 1. Navigate to CESS Explorer, click Network > Staking > Accounts > Validate



Consensus Node Set Up

Becoming a Validator

Step 2. Enter Reward Commission Percentage as “100” and Select Nominations as “No, block all nominations”

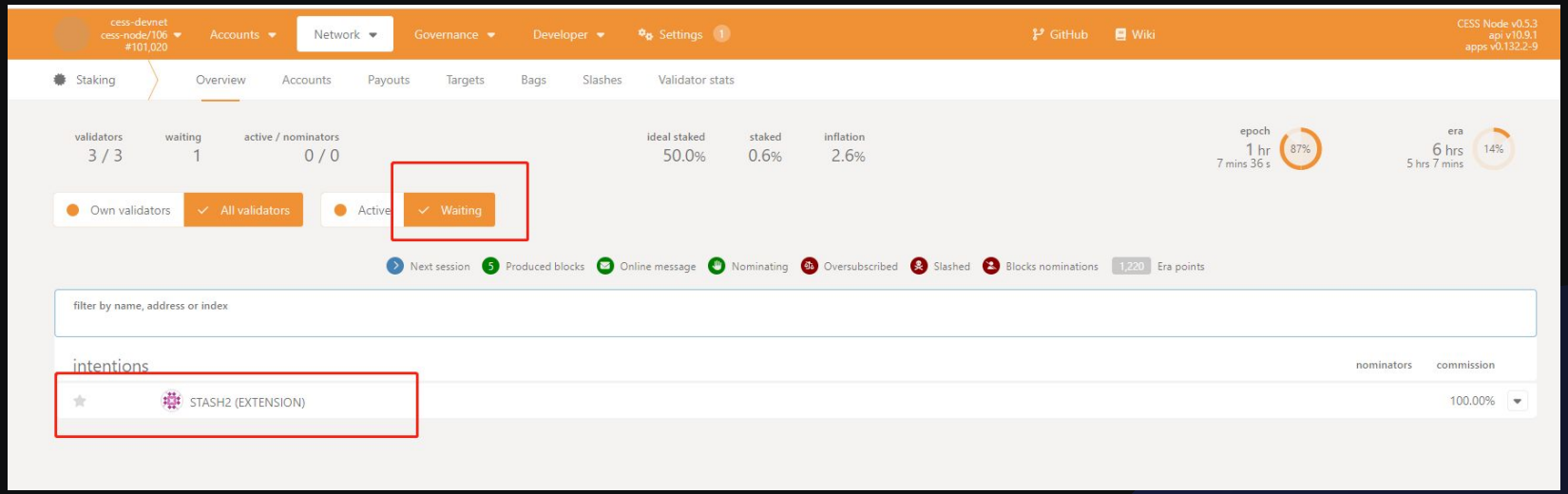


Click on Validate, and Sign and Submit the transaction

Consensus Node Set Up

Becoming a Validator

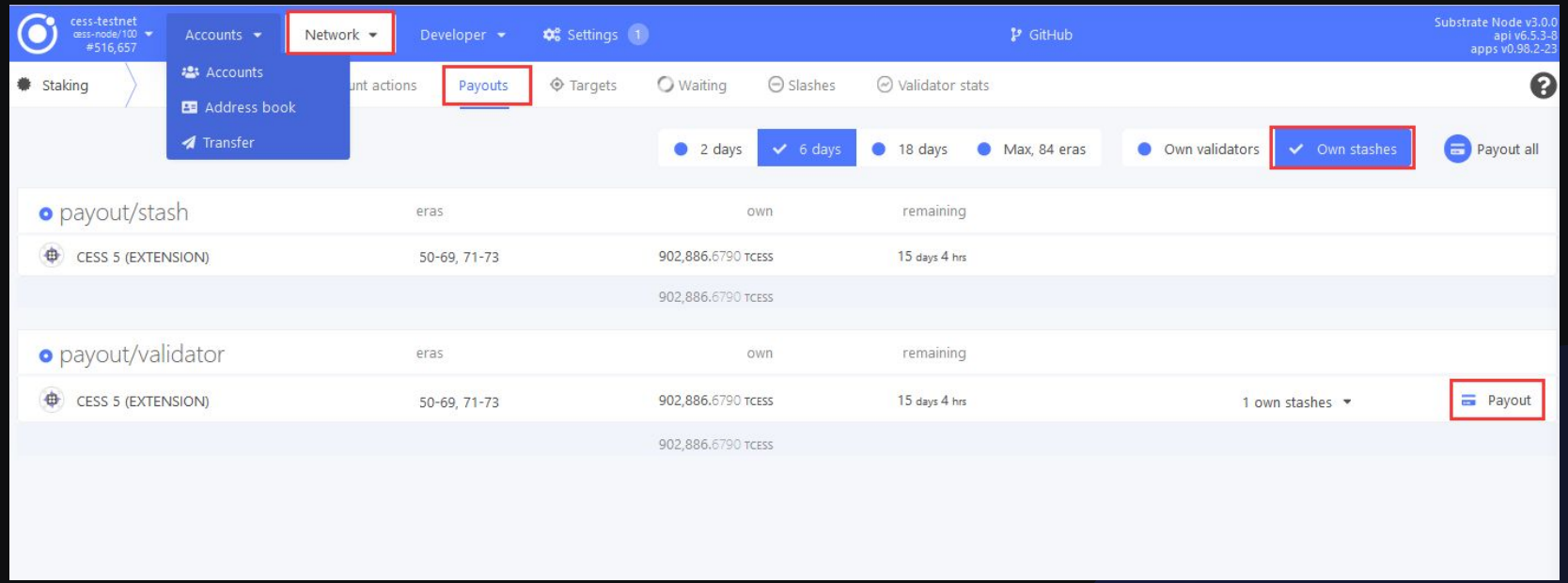
Step 3. Navigate to CESS Explorer, click Network > Staking > Waiting



Node appear in Candidate Node List

Redeeming Rewards

Navigate to CESS Explorer: Network > Staking > Payouts > Payout

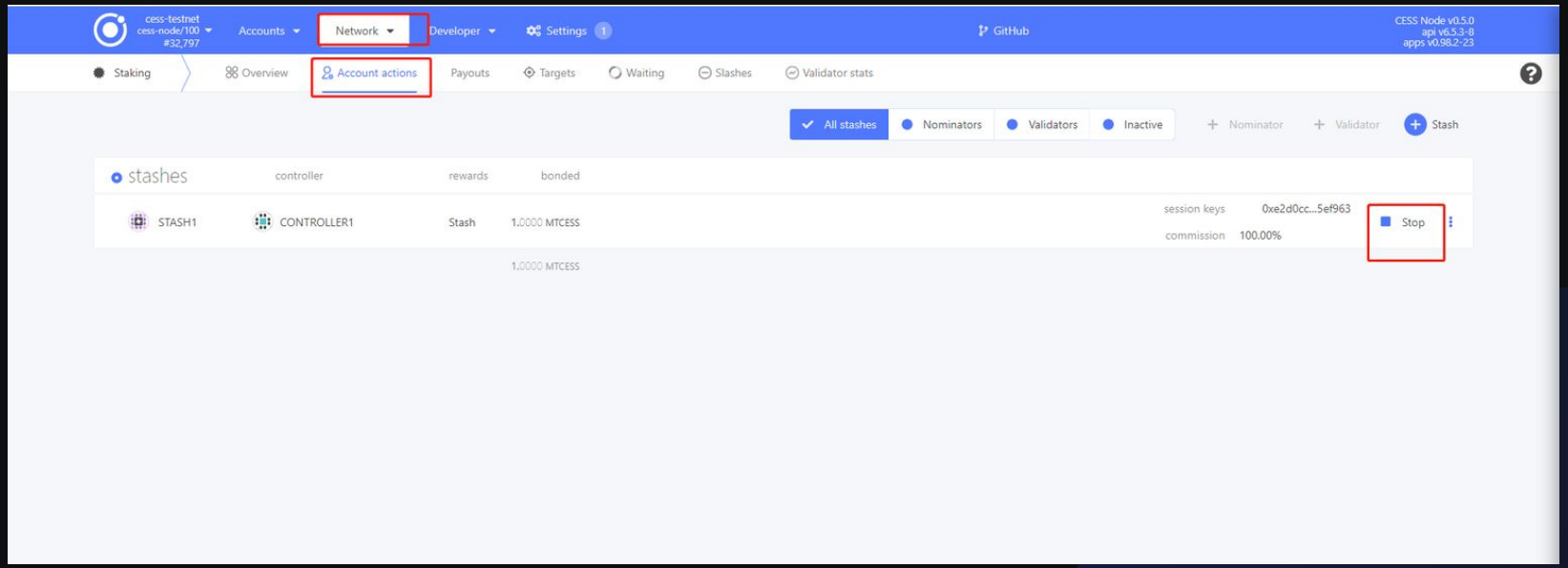


Select Desired Accounts and Click “Payout” then Sign and Submit the Transaction

Exiting Consensus Node from Validation

1. Stop the Consensus

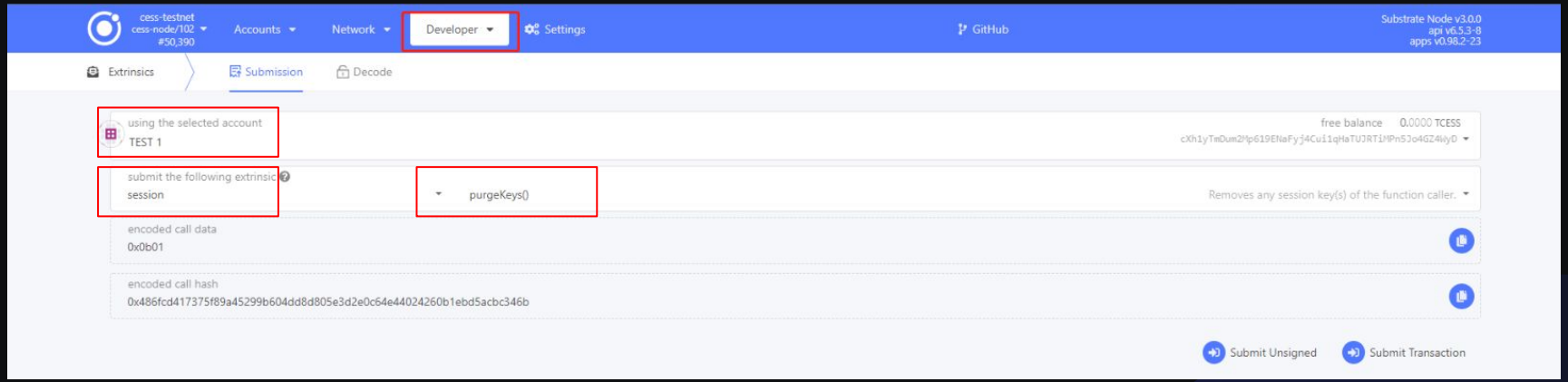
From CESS Explorer, navigate to: Network > Staking > Account Actions > Stop



Exiting Consensus Node from Validation

2. Clear Session Keys

From CESS Explorer, navigate to: Developer > Extrinsics



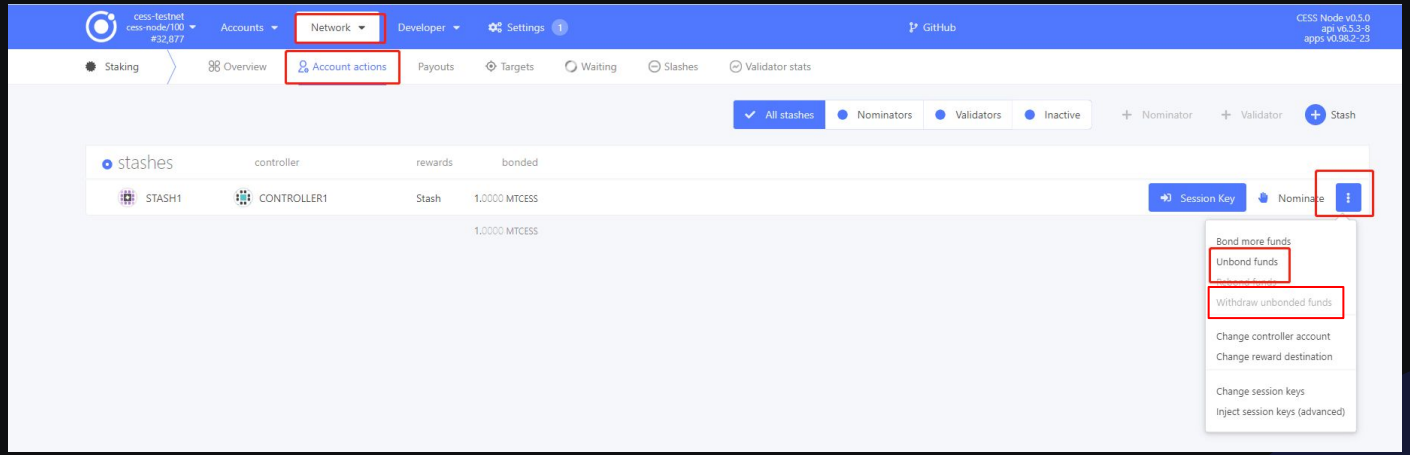
Select Appropriate Controller Account, then Select session from “submit the following extrinsic” and select purgeKeys()

Sign and Submit the transaction

Redeeming Stake

Unbound found and Withdraw

From CESS Explorer, navigate to: Network > Staking > Account Actions > Unbond Funds



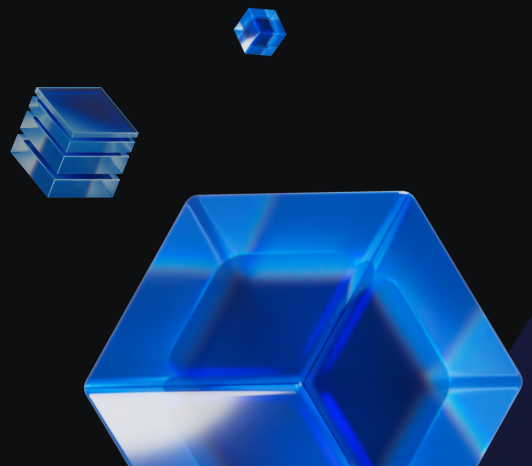
After submitting Unbound funds transaction we can withdraw unbounded funds after 28 eras (Each era is of 6 hours in testnet)

Stop the CESS Client - cess stop

CESS

Demo

Running a Consensus Node



CESS

Thank you for watching

Please Join Our Community

CESS

CESS Network - Episode 5

Demo: Running a Consensus Node

https://www.cess.network