



CESS Network

The Decentralized Data Infrastructure

Episode 6 Demo: Running a Storage Node



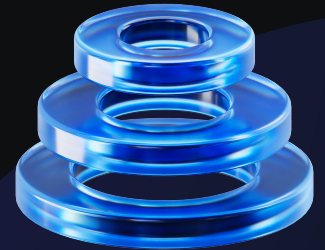
<https://www.cess.network>



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



Resource

Specification

- | | |
|---------------------------------|--------------------------|
| • Recommended OS | Linux 64-bit Intel / AMD |
| • # of CPU Cores | ≥ 4 |
| • Memory | ≥ 8 GB |
| • Bandwidth | ≥ 5 Mbps |
| • Public Network IP | Required |
| • Linux Kernel Version | 5.11 or higher |
| • HDD (Testnet) / SSD (Mainnet) | ≥ 50 GB (Testnet) |



Prerequisites



Software

- Docker and Docker Compose
[Official Documentation](#)
- Firewall Port 4001: For Ubuntu “*ufw allow 4001*”

Static Public IP

- `curl -4 ifconfig.co`

CESS Wallet Accounts

- Staking Account: 4000 TCESS / TB

Wallet Accounts

Earning Account

- Used to receive rewards.

Staking Account

- Used for Staking TCESS.

Signature Account

- Used to sign blockchain transactions.
- If no staking account is specified, Signature account will also be used for staking TCESS.

NOTE: Create Unique Signature Account for Each Storage Node or Exceptions May Occur



Storage Node Installation and Configuration



Install the CESS Client - nodeadm

```
wget https://github.com/CESSProject/cess-nodeadm/archive/refs/tags/v0.5.7.tar.gz  
tar -xvf v0.5.5.tar.gz  
cd cess-nodeadm-0.5.5  
sudo ./install.sh
```

Client Configuration

Set the desired network Devnet or Testnet

```
sudo cess profile devnet/testnet
```

Set Configuration

```
sudo cess config set
```



Storage Node Installation and Configuration



Enter cess node mode from 'authority/storage/watcher': storage

Enter cess storage listener port (current: 15001, press enter to skip):

Enter cess storage earnings account: <"ACCOUNT_ADDRESS">

Enter cess storage signature account phrase: <ACCOUNT_MNEMONICS>

Enter cess storage disk path: <DISK_PATH>

Enter cess storage space, by GB unit (current: 300, press enter to skip):

Enter the number of CPU cores used for mining; Your CPU cores are 4

(current: 3, 0 means all cores are used; press enter to skip):

Enter the staker's payment account if you have another (if it is the same as the signature account, press enter to

skip): <ACCOUNT_ADDRESS>

Enter the reserved TEE worker endpoints (separate multiple values with commas, press enter to skip):

NOTE: If no TEE worker endpoints are provided Default TEE worker endpoints will be used.

This doesn't affect your reward as a storage node.

Running the Storage Node



Command: `sudo cess start`

Output:

[+] Running 3/0

✓	Container chain	Running	0.0s
✓	Container bucket	Running	0.0s
✓	Container watchtower	Running	0.0s

NOTE: If you have access to Intel SGX with FLC Support you can also speed up your earnings by deploying a TEE-Worker as Market Type and Specifying it in your Storage Node. [TEE Worker User Guide](#)

Common Operations



Check Logs

`docker logs chain`: To check CESS Blockchain Status

```
2023-07-05 05:52:08 [ ] Idle (25 peers), best: #15590 (0xadf4...16f3), finalized #15588 (0xb289...d82e), ↓ 10.6kiB/s ↑ 14.5kiB/s
2023-07-05 05:52:12 +* Imported #15591 (0x1177...bc3e)
2023-07-05 05:52:13 [ ] Idle (26 peers), best: #15591 (0x1177...bc3e), finalized #15589 (0x368e...b4bb), ↓ 10.7kiB/s ↑ 12.2kiB/s
2023-07-05 05:52:18 +* Imported #15592 (0xe67a...b89f)
```

`docker logs bucket`: To check CESS Storage Node Logs

```
++ 2023-07-05 05:51:46 Start node discovery service
++ 2023-07-05 05:51:46 /kldr-testnet
OK 2023-07-05 05:51:46 Start successfully
2023/07/05 05:51:49 Connected to the bootstrap node: 12D3KoolWAdyc4qPwFh5xMtXvSrm7CXNFhUmKPQdoXukQXki69qBo
2023/07/05 05:51:49 Connected to the bootstrap node: 12D3KoolWLTpEaPbJhTyC8qpRp8PrjHy4ou7TCSY3XEtzNsh4Jatb
2023/07/05 05:52:05 Connected to the bootstrap node: 12D3KoolWHY6BRu2MtG9SempACgYCcGHRSEai2Zkwy3E4VKDYrqh9
```


Common Operations



View Storage Node Status

sudo ccess bucket stat

```
+-----+-----+
| name      | storage miner |
| peer id   | 12D3KooWRC8MG91CY9KZQ6HWYyV7zY4sR98rdLYv7ei5NAR1tVXF |
| state     | positive      |
| staking amount | 4000 TCESS    |
| validated space | 0 Bytes       |
| used space  | 0 Bytes       |
| locked space | 0 Bytes       |
| staking account | cXjHRBKDQ3LhxWJEqmLv6ZLjSNSStJcAJmUHLffNsAWRVgEMef |
| earnings account | cXjHRBKDQ3LhxWJEqmLv6ZLjSNSStJcAJmUHLffNsAWRVgEMef |
+-----+-----+
```

NOTE:

If you get the message "You are not a storage node", please wait for the chain synchronization to complete.

Common Operations



Increase Node Stake

```
sudo cess bucket increase stake <DEPOSITE_AMOUNT_IN_TCESS>
```

Query Reward Information

```
sudo cess bucket reward
```

Claim Reward

```
sudo cess bucket claim
```

Withdraw Node Stake

```
sudo cess bucket withdraw
```

Update Earning Account

```
sudo cess bucket update earnings <NEW_ACCOUNT_ADDRESS>
```

Updating CESS Client

```
sudo cess stop
```

```
sudo cess down
```

```
sudo cess purge
```

Then download and install new cess-nodeadm Client and execute

```
sudo cess pulling
```

Checking Storage Node Status On-Chain



Step 1.

[CESS Explorer](#) and Select
Developer > Chain State

Step 2.

From “selected state query”, Select
sminer and minerItems(AccountId32)

Step 3.

Select your Account Address, then
click on “+”

The screenshot shows the CESS Explorer interface for checking storage node status. The 'selected state query' section is highlighted with a red box. The 'sminer' query is selected, and the 'minerItems(AccountId32): Option<PalletSminerMinerInfo>' is chosen. The 'AccountId32' is set to 'PERSONAL-01 (EXTENSION)'. The 'encoded storage key' is shown, and the 'encoded key details' are displayed. The 'sminer.minerItems' data is shown in a JSON format.

selected state query
sminer minerItems(AccountId32): Option<PalletSminerMinerInfo> The hashmap for info of storage miners. +

Option<AccountId32> include option ☒ cXjHRBKDQ...

AccountId32
PERSONAL-01 (EXTENSION)

blockhash to query at
0X...

encoded storage key
0xeef51d8e841afa77e8c7349eaa7746f7b2e25dfd1af730b06c3bbbf79259aeb7b930922890ef52d8c41adbb0b2f15be7c1553d878bcd97e5195aede2884c931cd5d28e5f62b0f6ba12f86dcb0df0f

encoded key details

module	ee51d8e841afa77e8c7349eaa7746f7
method	b2e25dfd1af730b06c3bbbf7f09259ae
blake2_128concat(accountid32)	be7b930922890ef52d8c41adbb0b2f15be7c1553d878bcd97e5195aede2884c931cd5d28e5f62b0f6ba12f86dcb0df0f

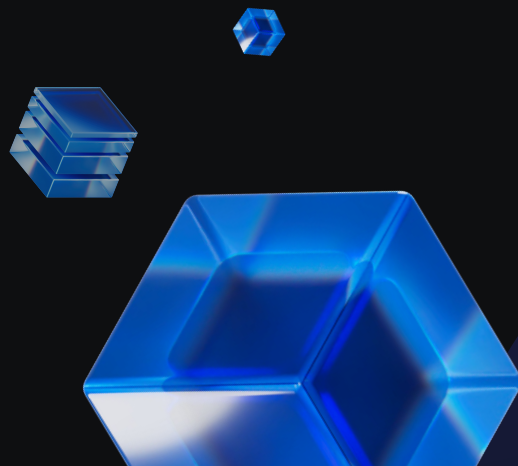
sminer.minerItems: Option<PalletSminerMinerInfo>

```
{
  beneficiary: cXjHRBKDQ3LhxWJEqmLv6ZLjSNStJcAJmUHLffNsAWRVgEMef
  peerId: 0x002408011220e47274f60c02c2befa41500bcfebbfff07408bba95cfe34d6cc5fcc79443ca82
  collaterals: 4,000,000,000,000,000
  debt: 0
  state: positive
  idleSpace: 34,359,738,368
  serviceSpace: 0
  lockSpace: 0
  spaceProofInfo: {
    miner: cXjHRBKDQ3LhxWJEqmLv6ZLjSNStJcAJmUHLffNsAWRVgEMef
    front: 0
    rear: 512
    poisKey: {
      g: 0x647cb40e5d615a684e0e242cbd870bbab75cb64669db4c38aef6c87bc03983850cf57e90c1095490f3259640e4fbf70137a9f19dde4...
      n: 0xdf6f1115a5cdd366901b2e8c50ab84d7fccbd82f43ab9ae7c9572fd0fe7bf89dbdbdb472dbfa610a50777e4f302a14e84779e6296d...
    }
    accumulator: 0xd4aef7d77cad21c202c7dd1841f91890e797cc5751eade0ebbde2b31d7b5c8afe0ced17753ff1dbf471fa2c3bb00e6fadd...
  }
}
```



Demo

Running a Storage Node





Thank you for watching

Please Join Our Community





CESS Network - Episode 6

Demo: Running a Storage Node



<https://www.cess.network>

