

# Web 3.0 Node Engine Service (NES)

## User Guide (Staking Nodes)

**Issue** 01  
**Date** 2024-11-28



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## **Huawei Cloud Computing Technologies Co., Ltd.**

Address: Huawei Cloud Data Center Jiaoxinggong Road  
Qianzhong Avenue  
Gui'an New District  
Gui Zhou 550029  
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

---

# Contents

---

<b>1 Overview.....</b>	<b>1</b>
<b>2 Usage Principle.....</b>	<b>4</b>
<b>3 Staking Operations.....</b>	<b>6</b>
3.1 Obtaining the Key Using the Staking Launchpad.....	6
3.2 Creating a Staking Node.....	13
3.3 Creating and Obtaining an API Key.....	14
3.4 Starting the Staking Node.....	15
3.5 Monitoring Staking Nodes.....	17
<b>4 VPC Endpoint (VPCEP) Connection.....</b>	<b>19</b>

# 1 Overview

---

Staking is one of the core mechanisms of Ethereum 2.0. It aims to implement the transition of network consensus algorithm from proof-of-work (PoW) to proof-of-stake (PoS). Staking is the act of depositing 32 ETH to activate validator software. As a validator, you will be responsible for storing data, processing transactions, and adding new blocks to the blockchain. This will keep Ethereum secure for everyone and earn you new ETH in the process.

## Benefits of Staking

- **Earn rewards:** Rewards are given for actions that help the network reach consensus. You will get rewards for running software that properly batches transactions into new blocks and checks the work of other validators because that is what keeps the chain running securely.
- **Better security:** The Ethereum network gets stronger against attacks as more ETH is staked, as it then requires more ETH to control a majority of the network. Validators are responsible for safeguarding the network and protecting their own interests. This entails that in the event of staking nodes violating regulations or launching network attacks, their ETH will be diminished.
- **More sustainable:** Stakers do not need to do energy-intensive PoW computations to secure the network, as they rely on staked ETH rather than computing power. This allows Ethereum to efficiently validate and process transactions, resulting in faster overall transaction speeds and throughputs.

## Staking Options

Selecting a staking solution depends on how much you are willing to stake. You will need 32 ETH to activate your own validator, but it is possible to stake less.

**Table 1-1** Staking options

Item	Solo Home Staking	Staking as a Service	Pooled Staking
Description	<p>Solo staking on Ethereum is the gold standard for staking. It provides full participation rewards, improves the decentralization of the network, and never requires trusting anyone else with your funds.</p> <p>Those considering solo staking should have at least 32 ETH and a dedicated computer connected to the Internet 24/7.</p>	<p>If you do not want or do not feel comfortable dealing with hardware but still want to stake your 32 ETH, this option allows you to delegate the hard part while you earn native block rewards. This option usually walks you through creating a set of validator credentials, uploading your signing keys to them, and depositing your 32 ETH. This allows the service to validate on your behalf.</p> <p>This method of staking requires a certain level of trust in the provider. To limit counter-party risks, the keys to withdrawal your ETH are usually kept in your possession.</p>	<p>Several pooling solutions now exist to assist users who do not have to or do not want to stake 32 ETH. Many of these options include what is known as "liquid staking" which involves an ERC-20 liquidity token that represents your staked ETH.</p> <p>Liquid staking enables easy and anytime exiting and makes staking as simple as a token swap. This option also allows users to hold custody of their assets in their own Ethereum wallet. Note that pooled staking is not native to the Ethereum network.</p>
Rewards	<p>Solo stakers receive full rewards directly from the protocol by batching transactions into a new block and checking the work of other validators.</p>	<p>This usually involves full protocol rewards minus monthly fee for node operations. Dashboards are often available to easily track your validator client.</p>	<p>Pooled stakers accrue rewards differently, depending on which method of pooled staking is chosen. Many pooled staking services offer one or more liquidity tokens that represent your staked ETH plus your share of the validator rewards. Liquidity tokens can be held in your own wallet, used in DeFi and sold if you decide to exit.</p>

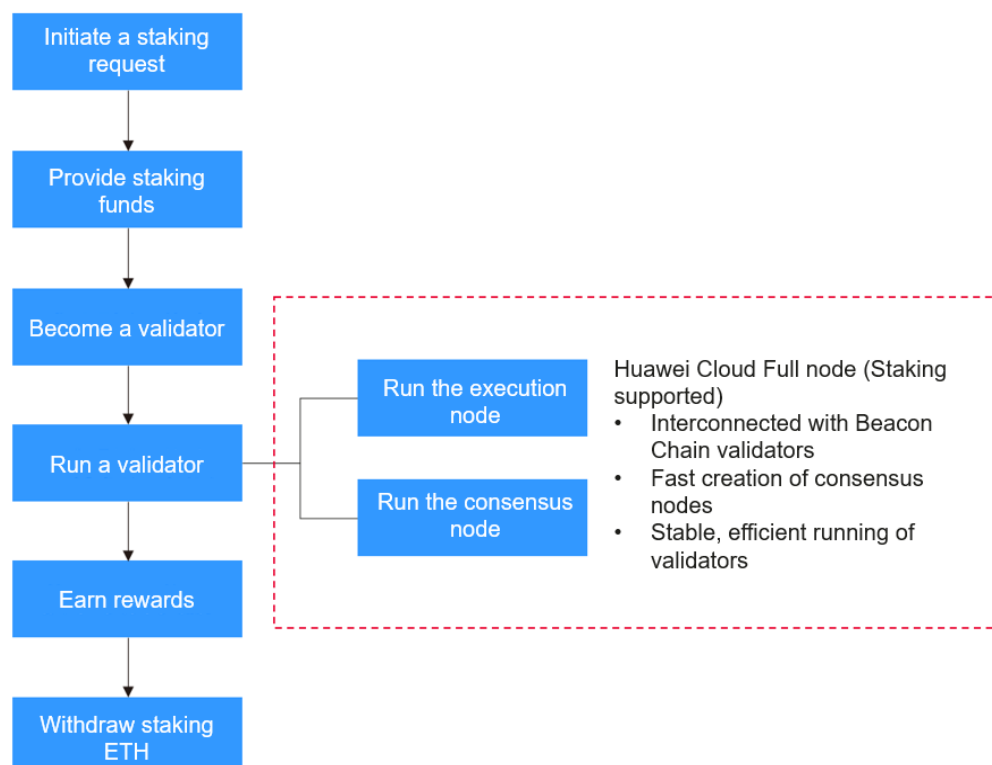
Item	Solo Home Staking	Staking as a Service	Pooled Staking
Risks	ETH is at stake and cannot be traded. Going offline or other malicious behaviors can result in "slashing" of larger amounts of ETH and forced ejection from the network.	Same risks as solo staking plus counter-party risk of the service provider.	Risks vary depending on the method used. In general, risks consist of a combination of counter-party, smart contract, and execution risk.
Requirements	<ul style="list-style-type: none"> <li>• Deposit 32 ETH.</li> <li>• Maintain hardware that runs both an Ethereum execution client and consensus client while connected to the Internet.</li> </ul>	<ul style="list-style-type: none"> <li>• Deposit 32 ETH and generate your keys with assistance.</li> <li>• Securely keep your keys.</li> <li>• Perform operations according to the service provider.</li> </ul>	<ul style="list-style-type: none"> <li>• This requires low-ETH commitment as some projects need as little as 0.01 ETH.</li> <li>• Deposit directly from your wallet to different pooled staking platforms or simply trade for one of the staking liquidity tokens.</li> </ul>

# 2 Usage Principle

Huawei Cloud nodes can be staked separately. To stake a node, you need to purchase a Huawei Cloud staking node, activate a validator, and interconnect the node with the validator. Huawei Cloud keeps the Ethereum nodes running stably by managing the Execution Layer (EL) and Consensus Layer (CL) clients. Note that Huawei Cloud will not keep your keys.

The following figure shows the staking process.

**Figure 2-1** Staking process



The operations in the red box are performed by NES. Other operations are performed by you.

The following explains the details.

1. Initiate a staking request.  
You initiate a staking request and learn about the advisories provided by Ethereum.
2. Provide staking funds.  
You deposit a certain amount of ETH to the staking contract. These ETH will support the validator running.
3. Become a validator.  
By depositing funds, you will be able to participate in reaching consensus. To become a validator, an individual must stake a specific amount of ETH, for instance, solo staking requires 32 ETH to be staked, and has some technical skills such as knowing how to set up and start a validator client.
4. Run a validator.  
You need to verify transaction validity and batch blocks as a validator. To ensure that your validator runs and operates properly, usually, you will need to start an execution node and a consensus node. Huawei Cloud NES provides open gRPC for Beacon Chain validator interconnection. With just a few clicks, EL/CL nodes with the 8 vCPUs | 32 GB flavor can be created effortlessly, eliminating the need for O&M. Additionally, Huawei Cloud-developed algorithms ensure efficient operations of validators.
5. Earn rewards.  
Validators will receive block rewards and earn transaction fees. These awards will be allocated based on the staking funds and contribution from the participants.
6. Withdraw staking ETH.  
This is optional. The Shanghai/Capella upgrade enabled staking withdrawals on Ethereum.



# 3 Staking Operations

---

## 3.1 Obtaining the Key Using the Staking Launchpad

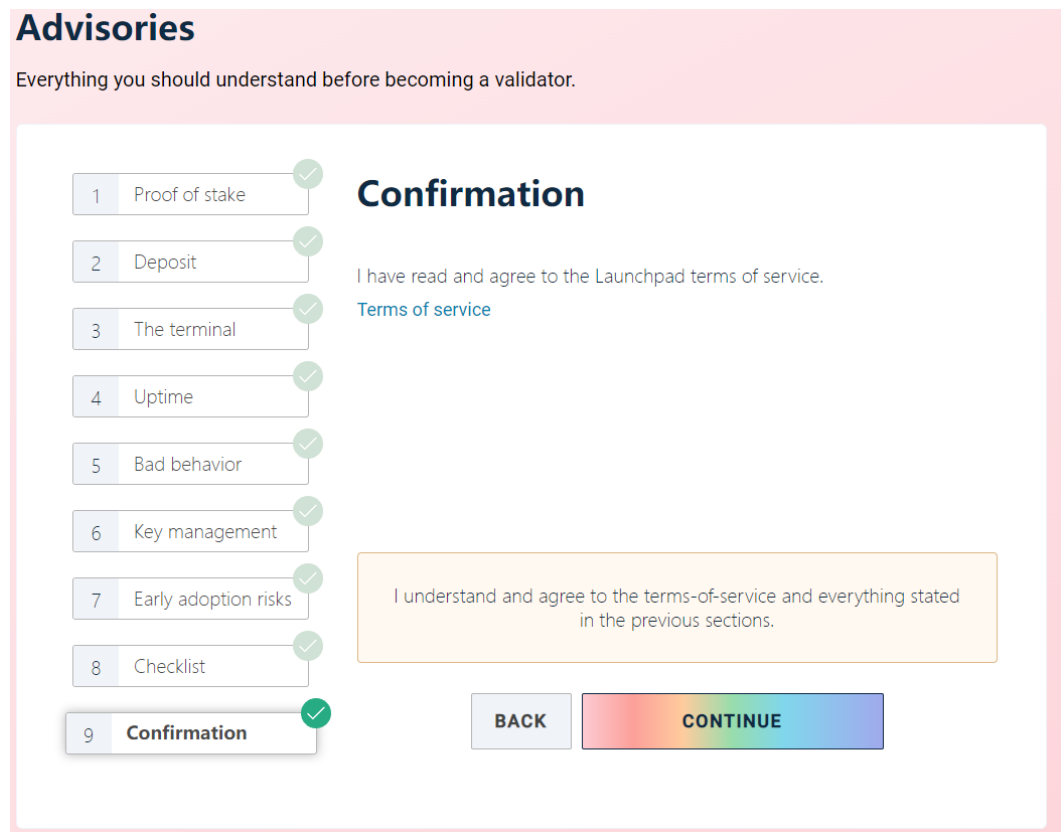
The Staking Launchpad is an open source application that will help you become a staker. It will guide you through choosing your clients, generating your keys, and depositing your ETH to the staking deposit contract. A checklist is provided to make sure you have covered everything to get your validator set up safely.

Mainnet and Holesky testnet are supported on the Staking Launchpad. It is recommended to test your setup and operational skills on the Holesky testnet.

**Step 1** Learn about staking.

The Staking Launchpad provides advisories for you to learn about staking.

Figure 3-1 Advisories



**Step 2** Choose your execution client and consensus client.

You can choose the clients from Eth1 and Eth2 providers and set up your nodes accordingly. If you have purchased Huawei Cloud staking nodes, NES will help you create execution nodes and consensus nodes, so that you can perform staking efficiently.

**NOTE**





NES uses Geth as the execution client and Prysm and Lighthouse as the consensus client.

Figure 3-2 Choosing an execution client

## Choose execution client

### Choose your execution client and set up a node

To process incoming validator deposits from the execution layer (formerly 'Eth1' chain), you'll need to run an execution client as well as your consensus client (formerly 'Eth2').

 <b>Nethermind</b> C#, .NET	 <b>Besu</b> Java	 <b>Erigon</b> Go	 <b>Geth</b> Go
--	--	---	--

[View extensive client comparison ↗](#)

Currently Geth is used by >66% of the network.

Client diversity is extremely important for the network health of Ethereum: A bug in a client with a share of over 33% can cause Ethereum to go offline. If the client has a supermajority (>66%), a bug could cause the chain to incorrectly split, potentially leading to slashing.

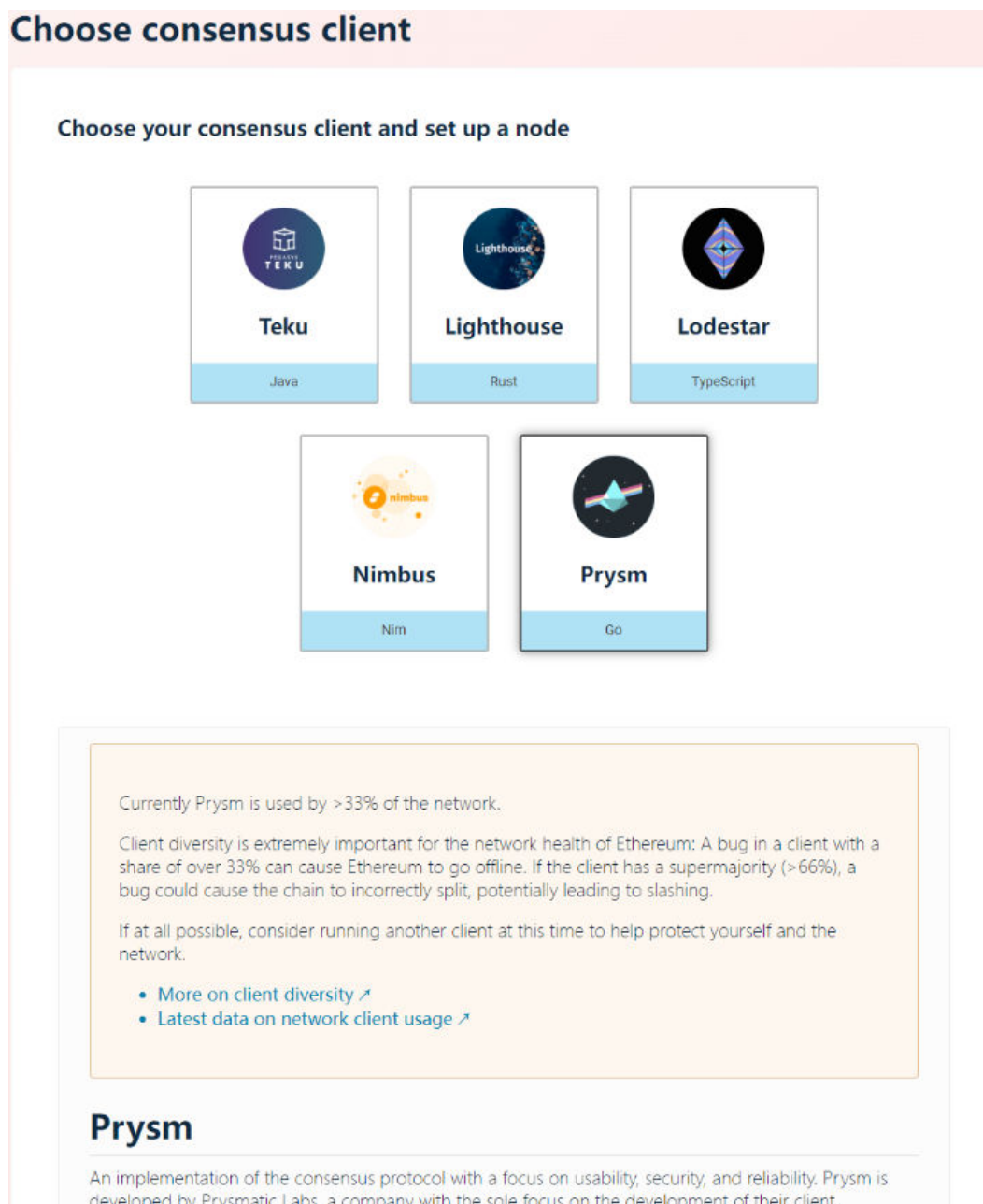
If at all possible, consider running another client at this time to help protect yourself and the network.

- [More on client diversity ↗](#)
- [Latest data on network client usage ↗](#)

## Geth

One of the three original implementations of the Ethereum protocol.

Figure 3-3 Choosing a consensus client



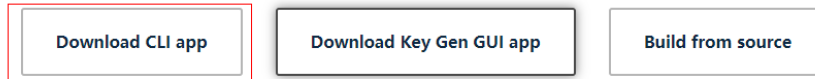
**Step 3** Generate keys.

Generate keys using the [key generation tool](#) provided by Ethereum, and keep the keys safe.

The following shows how to generate keys by downloading the CLI app, that is, the deposit command line interface app.

Figure 3-4 Choosing a key generation tool

### How do you want to generate your keys?

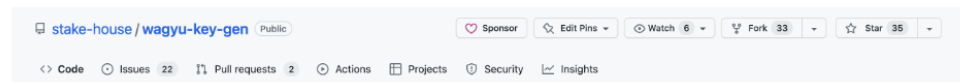
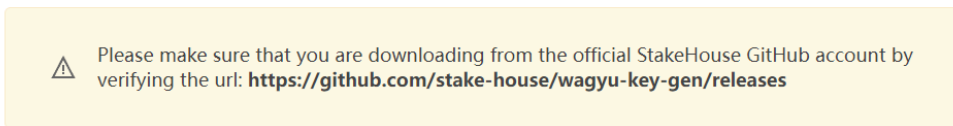


### Download Wagyu Key Gen app

Step 1: Download the Wagyu Key Gen app for your operating system

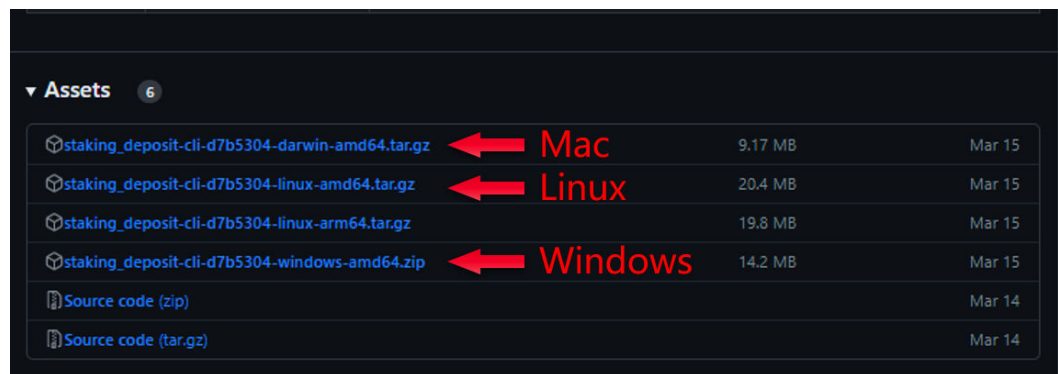


View Wagyu Key Gen audit by HashCloak



Download the [deposit command line interface](#) app from GitHub.

Figure 3-5 Downloading the tool



Decompress the file you just downloaded. Use the terminal/PowerShell to move into the directory that contains the tool and run the following commands:

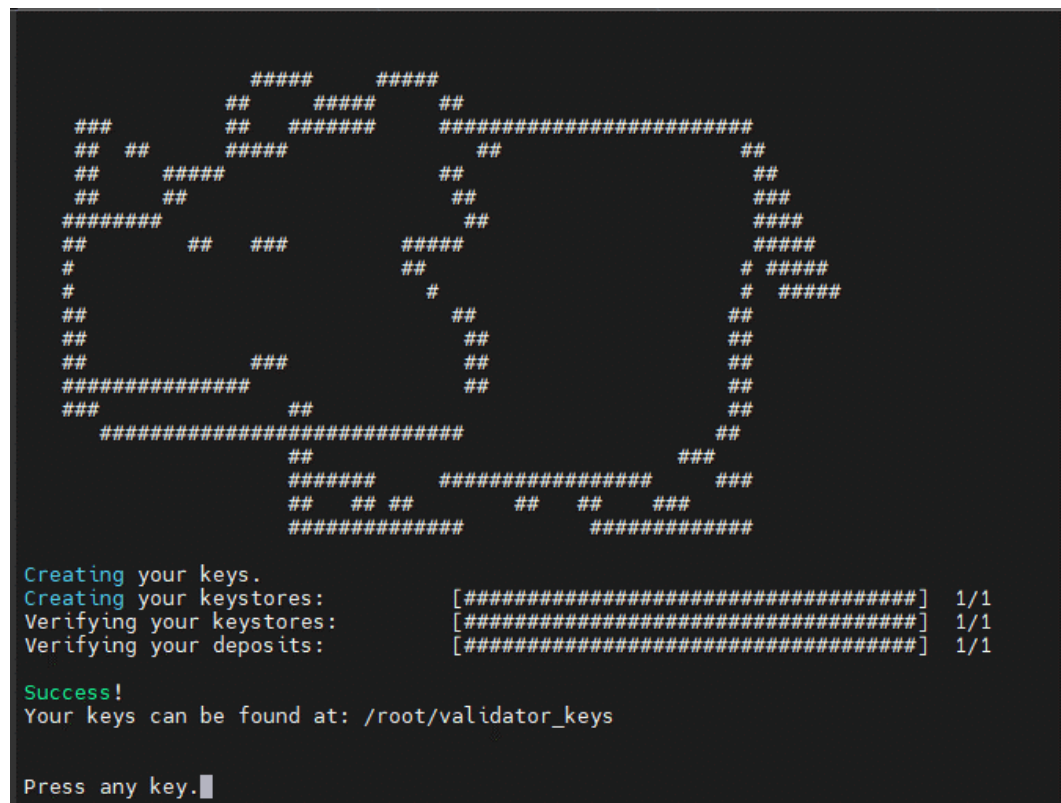
```
Linux/Mac
./deposit new-mnemonic
Windows
.\deposit.exe new-mnemonic
```

Then, follow the instructions to generate your keys.

```
[root@ecs-devnet01 ~]# ./deposit new-mnemonic
***Using the tool on an offline and secure device is highly recommended to keep your mnemonic safe.***
Please choose your language [ '1. 3' ', العربية, English', '4. Français', '5. Bahasa melayu', '6. Italiano', ]: [English]:
Please choose the language of the mnemonic word list [ '4. English', '5. Italiano', ]: [english]:
```

Please choose how many new validators you wish to run: 1  
Please choose the (mainnet or testnet) network/chain name ['mainnet', 'goerli', 'sepolia', 'zhejiang']:  
[mainnet]: goerli  
Create a password that secures your validator keystore(s). You will need to re-enter this to decrypt them when you set up your Ethereum validators:  
Repeat your keystore password for confirmation:  
This is your mnemonic (seed phrase). Write it down and store it safely. It is the ONLY way to retrieve your deposit.  
warm merry prevent invalid crash potato tongue portion rapid render chair outdoor summer red chief  
lesson firm culture infant or island bread fade property  
Press any key when you have written down your mnemonic.  
Please type your mnemonic (separated by spaces) to confirm you have written it down. Note: you only need to enter the first 4 letters of each word if you'd prefer.

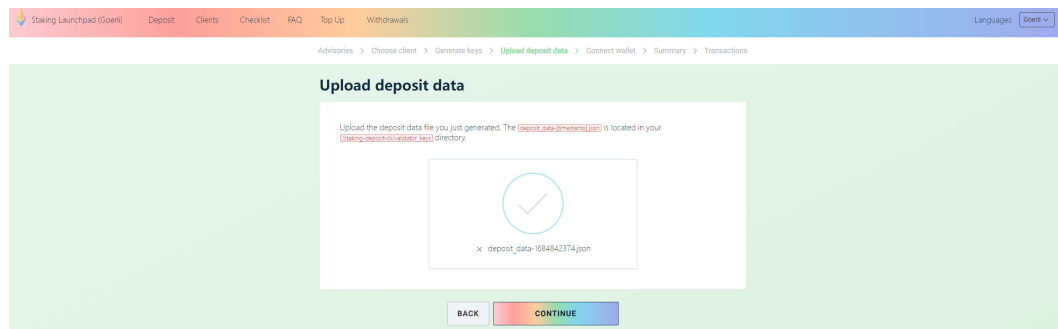
**Figure 3-6** Completing the key generation process



**Step 4** Upload deposit data.

Upload the deposit data file you just generated. Ensure that the network appeared in the upper right corner is correct. If you generate a key on the wrong network, the console will prevent you from moving on.

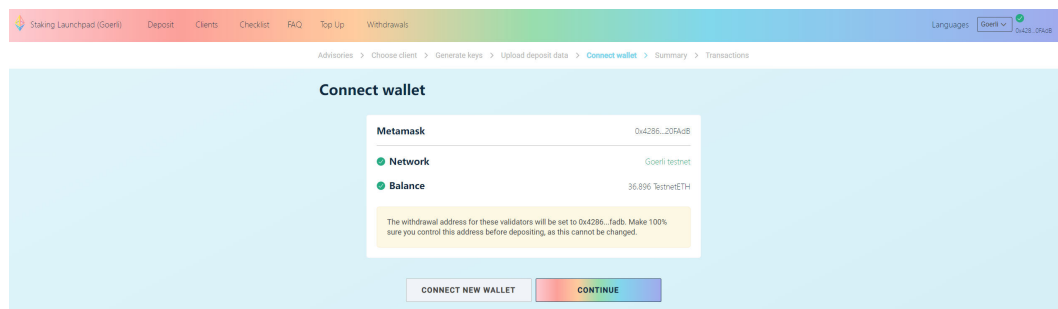
Figure 3-7 Uploading deposit data



**Step 5** Connect to the wallet.

Connect your wallet to the console and make sure you have 32 ETH in your account.

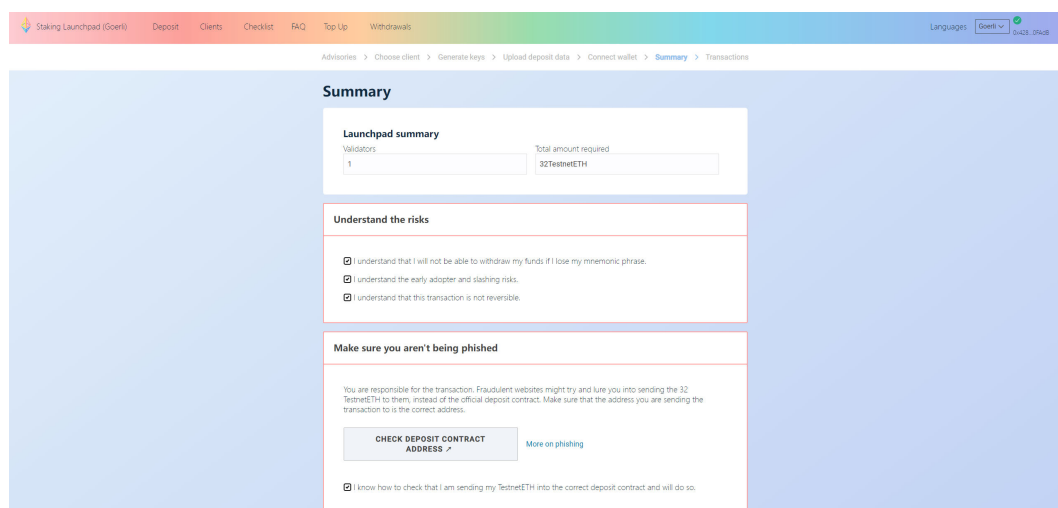
Figure 3-8 Connecting to the wallet



**Step 6** Confirm the information.

Check the information to ensure it is correct.

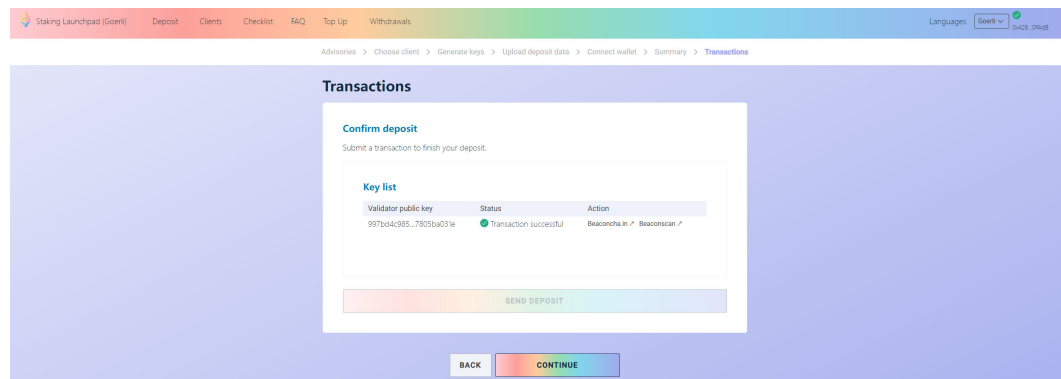
Figure 3-9 Confirming the information



**Step 7** Send the deposit.

Click **CONTINUE** to send the deposit to the Beacon Chain. Now, you have submitted a transaction. Next, you will start the client to complete staking.

Figure 3-10 Sending the deposit



**NOTE**

For details, see [Staking Launchpad \(Goerli\)](#) and [Staking Launchpad](#).

----End

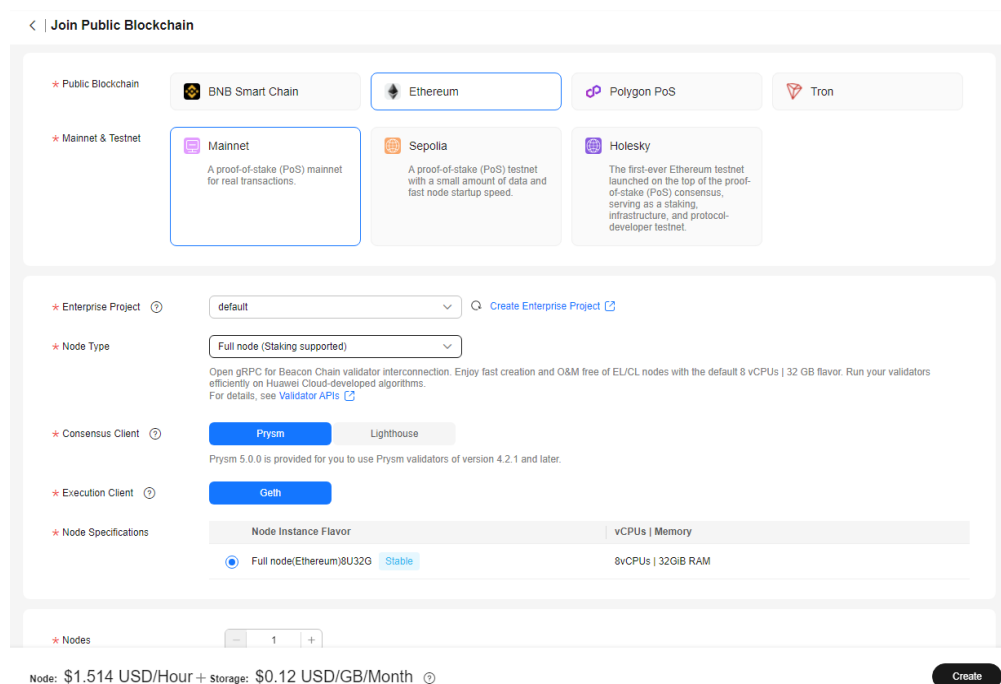
## 3.2 Creating a Staking Node

**Step 1** Log in to the NES console.

**Step 2** Click **Network Management** and click **Join Public Blockchain**.

**Step 3** Configure parameters.

Figure 3-11 Creating a staking node



**Step 4** Click **Create**.



**Step 5** Select **I have read and agree to the HUAWEI CLOUD User Agreement and Disclaimer.** and click **Submit.**

 **NOTE**

- It takes about 5 to 8 seconds to complete the process.
- Currently, only staking nodes of Ethereum mainnet, Sepolia, and Holesky are supported.

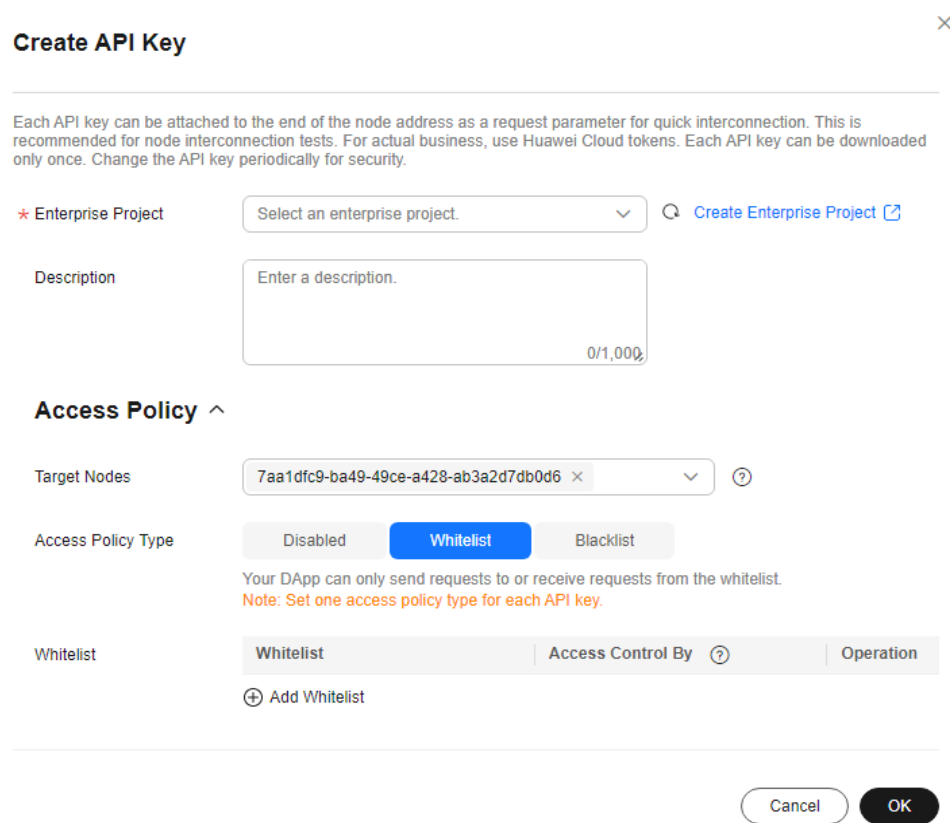
----End

## 3.3 Creating and Obtaining an API Key

**Step 1** On the NES console, choose **Dedicated > API Keys**, then click **Create API Key.**

**Step 2** Describe the API key and set the access policy.

**Figure 3-12** Creating an API key



**Create API Key** ✕

Each API key can be attached to the end of the node address as a request parameter for quick interconnection. This is recommended for node interconnection tests. For actual business, use Huawei Cloud tokens. Each API key can be downloaded only once. Change the API key periodically for security.

★ Enterprise Project  [Create Enterprise Project](#)

Description  0/1,000

**Access Policy** ^

Target Nodes  ?

Access Policy Type  Disabled  **Whitelist**  Blacklist

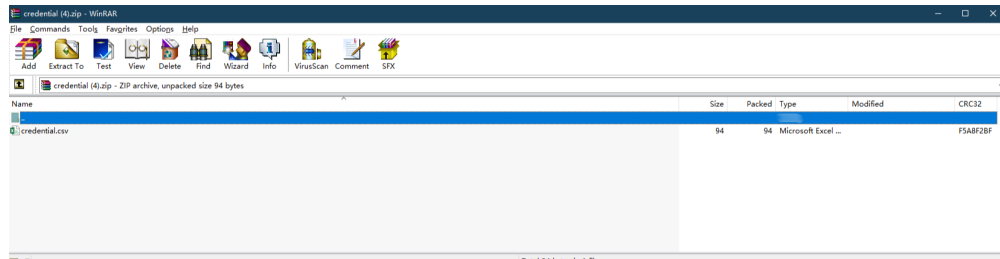
Your DApp can only send requests to or receive requests from the whitelist.  
**Note:** Set one access policy type for each API key.

Whitelist 

Whitelist	Access Control By	Operation
<span>+</span> Add Whitelist		

Cancel OK

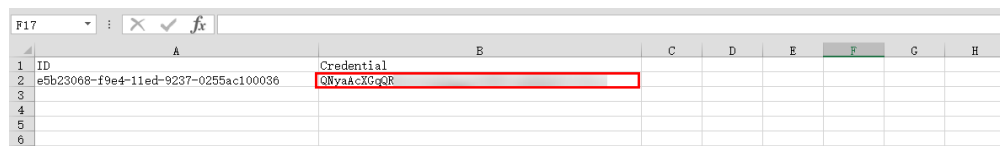
**Step 3** Click **OK.** The API key is created and then automatically downloaded as a ZIP package.



**NOTE**

Each API key can be downloaded only once. Change the API key periodically for security.

**Step 4** Decompress the package and open the **credential.csv** file to obtain the API key.



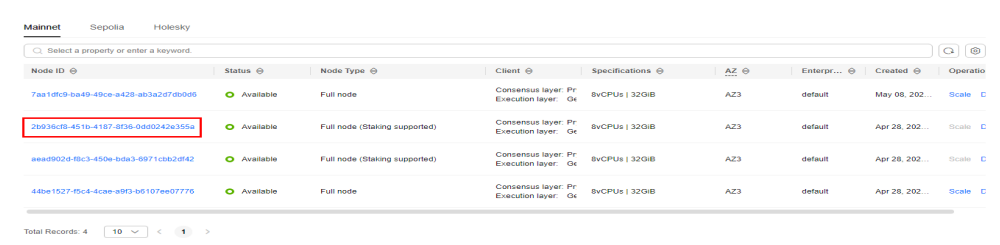
----End

## 3.4 Starting the Staking Node

**Step 1** On the NES console, click **Network Management**.

**Step 2** Click a node ID.

**Figure 3-13** Node ID

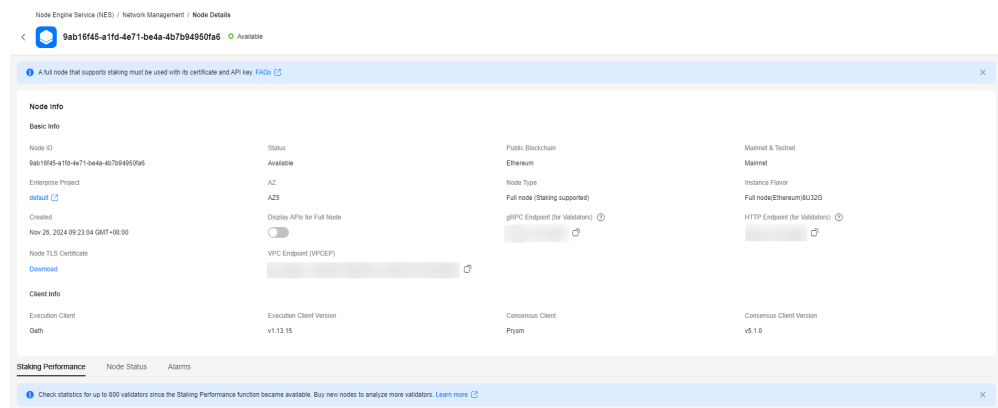


**Step 3** Obtain the node information.

For a Prysm client, you can obtain its **gRPC Endpoint**, **Node TLS Certificate**, and **HTTP Endpoint**.

For a Lighthouse client, you can obtain its **HTTP Endpoint** and **Node TLS Certificate**.

**Figure 3-14** Node details of a Prysm client



**Step 4** Paste the key and TLS certificate to the hardware machine installed with the script.

For a Prysm client, run the following command to import the key to the keystore:  
`./prysm.sh validator accounts import --keys-dir=<YOUR_FOLDER_PATH> --< NETWORK >`

*NETWORK* is the staking network and *YOUR\_FOLDER\_PATH* is the actual key file path.

For a Lighthouse client, run the following command to import the key to the keystore:

`lighthouse --network < NETWORK > account validator import --directory < YOUR_FOLDER_PATH >`

*NETWORK* is the staking network and *YOUR\_FOLDER\_PATH* is the actual key file path.

**Step 5** After the key is imported, perform the following operations for a Prysm client and Lighthouse client, respectively.

For a Prysm client, run the **prysm.sh** file, configure the following parameters, and start the staking node.

- *beacon-rpc-provider*: the value of **gRPC Endpoint**
- *grpc-headers*: the API key
- *tls-cert*: the relative path of **Node TLS Certificate**

Example:

`./prysm.sh validator --beacon-rpc-provider=xx.xx.xx.xx:30002 --grpc-headers=credential=xxxxxxxxxxxxxxxxxxxxxxxx --tls-cert=ca.crt`

For a Lighthouse client, run the **lighthouse vc** command, configure the following parameters, and start the staking node.

- *network*: the staking network
- *suggested-fee-recipient*: the suggested fee recipient
- *beacon-nodes-tls-certs*: the relative path of **Node TLS Certificate**
- *beacon-nodes*: the HTTP endpoint or API key information

`lighthouse vc --network < **NETWORK** > --suggested-fee-recipient < **YourFeeRecipientAddress** > --beacon-nodes-tls-certs ca.pem --beacon-nodes https://xx.xx.xx.xx:30000/xx`

 NOTE

These parameters are mandatory for interconnecting Huawei Cloud nodes. Check the [Prism Documentation](#) and [Lighthouse Documentation](#) to learn other parameters.

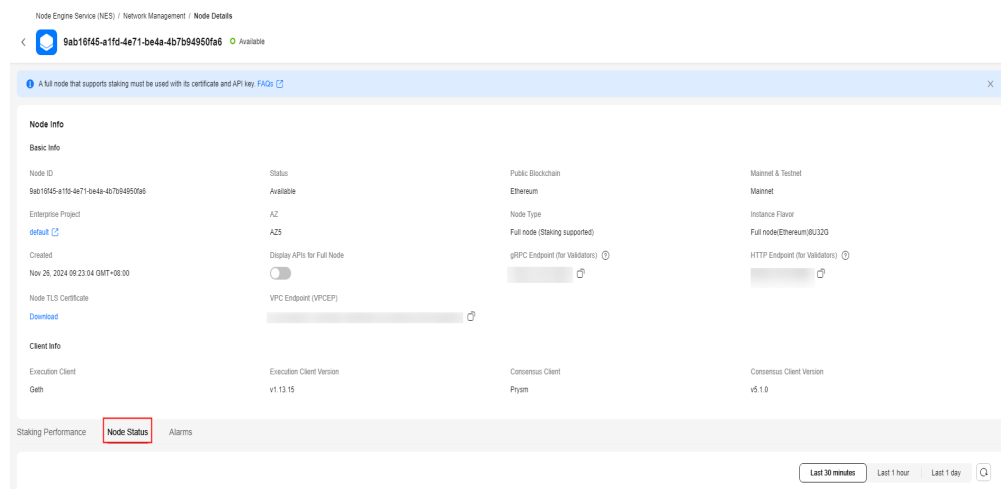
----End

## 3.5 Monitoring Staking Nodes

**Step 1** On the NES console, click **Network Management**.

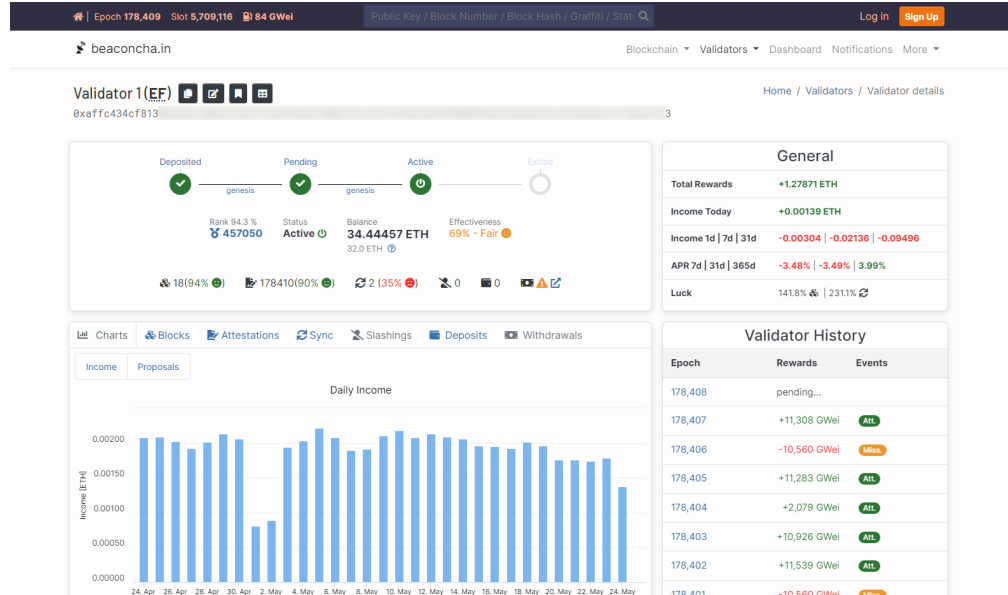
**Step 2** Click a node ID and click the **Node Status** tab page.

**Figure 3-15** Node status



 NOTE

You need to monitor and perform O&M on the validator client where a staking node has been started. You can also enter the key [on a page similar to the following](#) to check the client execution.



----End

# 4 VPC Endpoint (VPCEP) Connection

VPC endpoints are used for connecting your staking nodes with backend resources, such as Elastic Cloud Server (ECS) and Cloud Container Engine (CCE), through a Huawei Cloud private network.

## Prerequisites

You have **created at least one staking node** and enabled **VPC Endpoint (VPCEP)** for it.

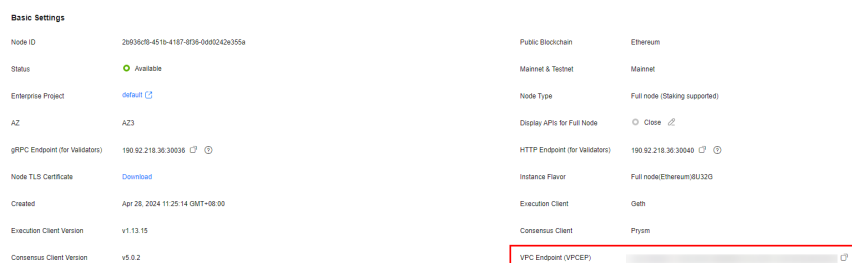
## Precautions

- The staking node, VPC endpoint, and backend resources must be in the same region, for example, AP-Singapore.
- The VPC endpoint and backend resources must be in the same Virtual Private Cloud (VPC).

## Procedure

**Step 1** Obtain the VPC endpoint service name.

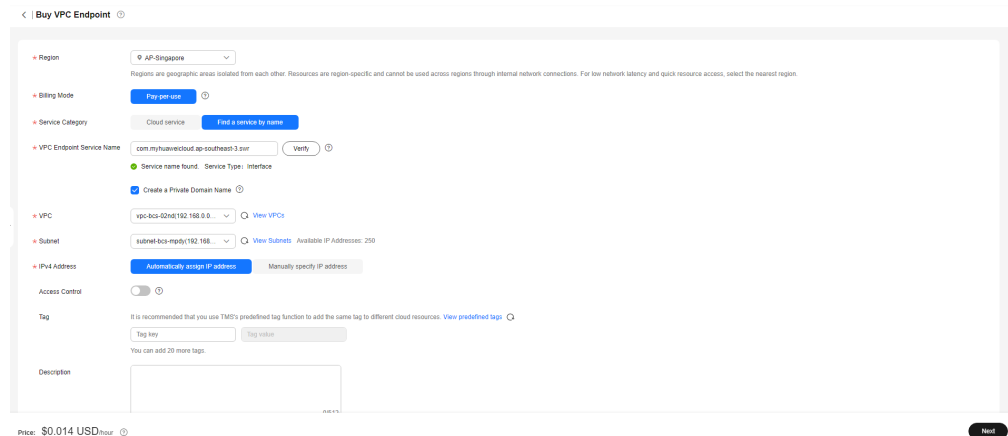
1. Log in to the NES console.
2. Choose **Dedicated > Network Management**.
3. Click a node ID and obtain the **VPC Endpoint** on the node details page.



**Step 2** Buy a VPC endpoint.

1. Log in to the **VPC Endpoint** console.
2. Choose **VPC Endpoint > VPC Endpoints**.

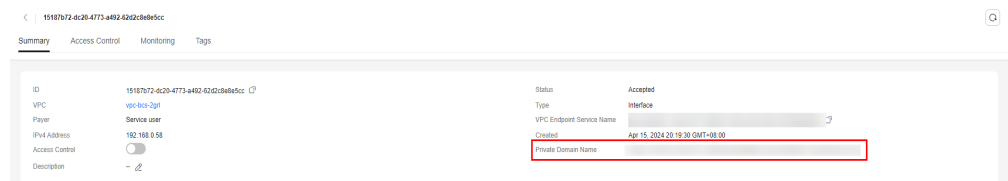
3. Click **Buy VPC Endpoint** and configure parameters. For details, see [Buying a VPC Endpoint](#).



**NOTE**

Set **Service Category** to **Find a service by name**. Then, paste the obtained VPC endpoint service name in **VPC Endpoint Service Name**.

4. Click **Next**, confirm parameters, and submit the order.
5. On the **VPC Endpoints** page, click the ID of the purchased VPC endpoint, and obtain its **Private Domain Name**.



**Step 3** Access the staking node using its port number and the private domain name of the VPC endpoint.

gRPC endpoint: *Private network domain name:Port number*

HTTP endpoint: *Private network domain name:Port number*

**NOTE**

Obtain the port numbers on the NES console. They are the values of **gRPC Endpoint (for Validators)** and **HTTP Endpoint (for Validators)** on the node details page.

----End