Web 3.0 Node Engine Service (NES)

Service Overview

Issue 01

Date 2024-05-10

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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.

Contents

1 What Is Web3 NES?

Web3 Node Engine Service (NES) is a blockchain node engine platform developed by Huawei Cloud. It simplifies blockchain network management, resource management, and authentication, while also providing developers with the ability to connect to mainstream blockchains like Ethereum. NES offers a stable, efficient, and secure infrastructure for Web3 services.

The dedicated edition of NES offers full-node services with complete hosting capabilities and can be used by:

- DApp developers and users: They can configure nodes to interact with blockchains with dispatch.
- Staking node carriers and individuals: They can use NES to host Ethereum full nodes, including execution and beacon nodes, and run validator nodes to connect with the hosted nodes.

The shared edition of NES provides packages for you to access different blockchains flexibly and cost-effectively, without the need to configure nodes.

NOTE

Currently, NES is available only in the AP-Singapore region.

2 Functions

This section describes the functions on the NES console.

Table 2-1

Edition	Module	Description
Dedicated	Dashboard	Check the quick start process and NES resources under the current Huawei Cloud account, including the number of public blockchains, nodes, full nodes, full nodes (staking), and API calls. For details, see **Figure 2-1**.
	Network Management	Manage the created public blockchain nodes and monitor them in real time. For details, see **Figure 2-2**.
	API Keys	Obtain the keys used by the management nodes easily and securely. For details, see **Figure 2-3**.
Shared	Dashboard	Check the quick start process, service uptime, and resources under the current Huawei Cloud account, including the specifications, projects, and API calls of the package. For details, see **Figure 2-4**.
	Package Management	Manage purchased packages. For details, see **Figure 2-6**.
	DApp Project Management	Manage the created projects under the current project. Check their API keys, HTTPS, WebSocket, and real-time statuses. For details, see **Figure 2-5**.

Figure 2-1 Dashboard

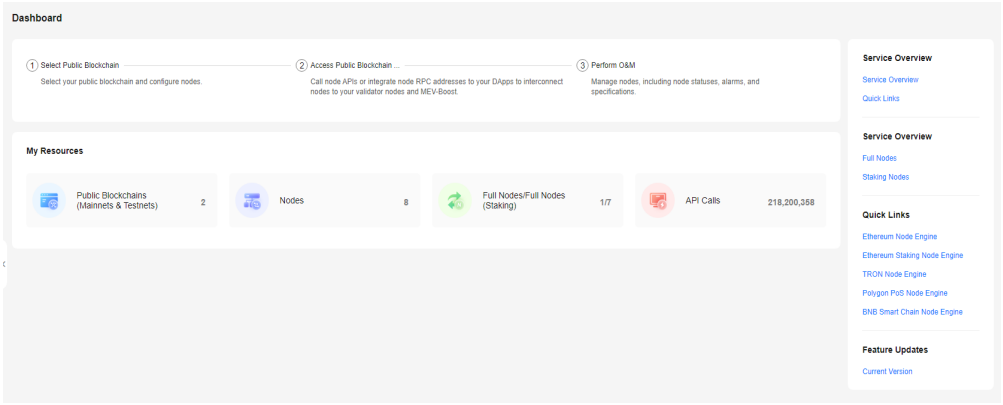


Figure 2-2 Network Management

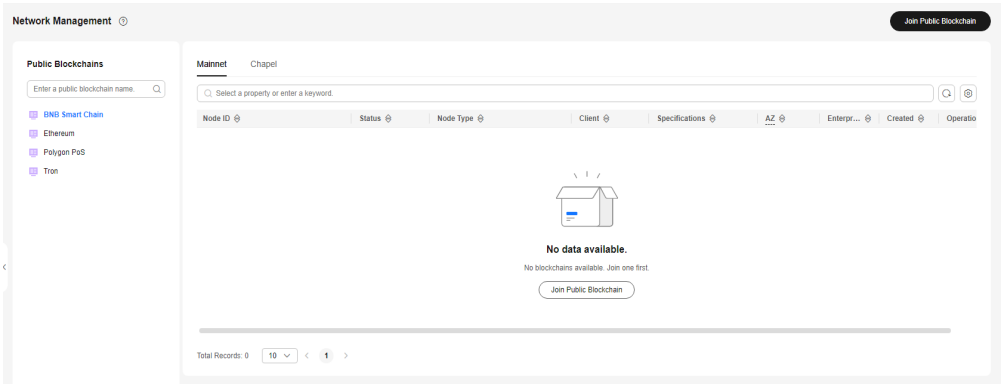


Figure 2-3 API Keys

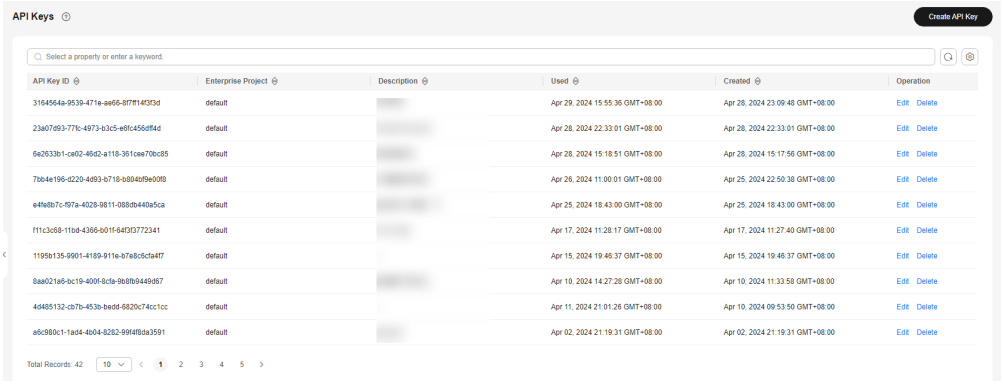


Figure 2-4 Dashboard

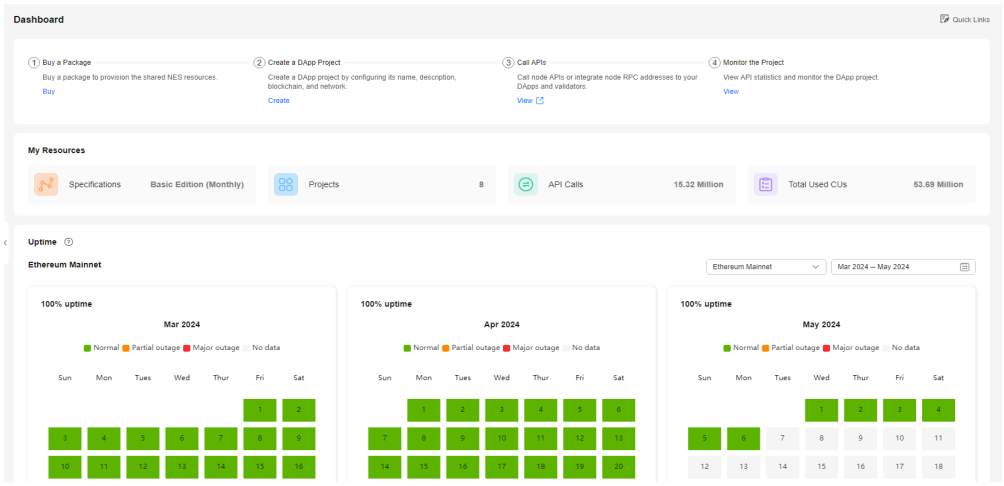


Figure 2-5 DApp Project Management

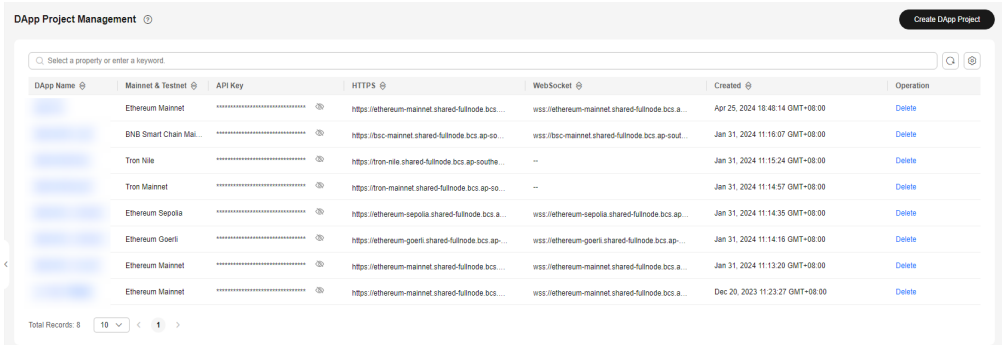
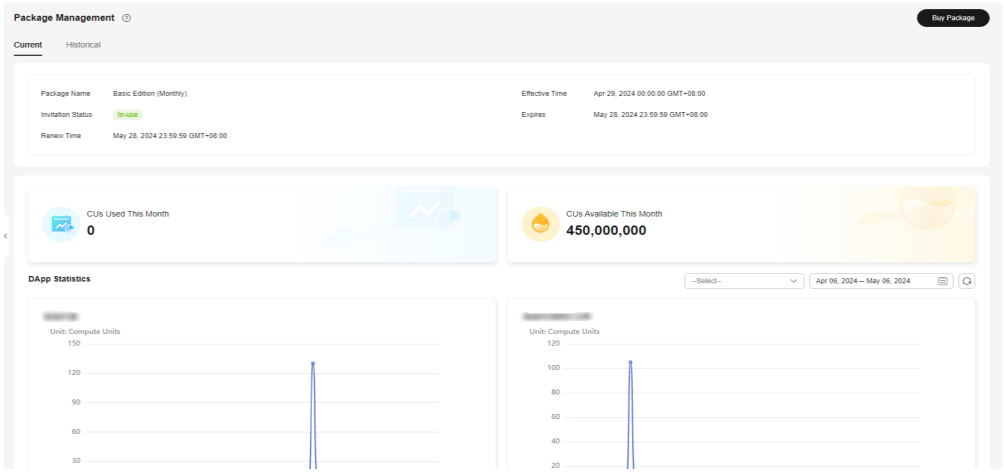


Figure 2-6 Package Management



3 Advantages

Easy to Use

Creating nodes is easy and quick. Node ledgers can be started instantly without the need for maintenance and operations.

Stable and Reliable

99.99% reliability and second-level node fault detection and recovery

Popular Blockchains Interconnected

Popular blockchain networks are supported, including Ethereum, TRON, Polygon, Arbitrum, Starknet, Optimism, and other L2 solutions, with more blockchains coming soon.

High-Performance Dual-Engine Drive

Staking carriers/individuals can have access to their Ethereum validator nodes, thanks to predictive analytics and topology algorithm optimization engines. These engines monitor global node statuses and block generation rates in real time to create an optimal task execution plan. By creating a distributed node network with low latency, these engines ensure timely and accurate validation tasks, and power validator nodes with up to 99% effectiveness.

Distributed Validator Technology (DVT)

An open-source, distributed verification technology solution that supports one-click deployment of verification nodes and multiple low-latency consensus algorithms. It offers flexible configuration and switchover, ensuring stable operation of verification nodes and maximizing benefits for users.

4 Application Scenarios

DApp Development and Usage

Effortlessly create and manage popular blockchain nodes with speed and ease, without the need for operations or maintenance.

The nodes hosted on NES are interconnected with popular blockchains like Ethereum and TRON, allowing for seamless development and use of DApps without the need to worry about node operations.

Validator Node Interconnected

NES is interconnected with Ethereum validator nodes.

Ethereum nodes at the execution and consensus layers can be hosted to connect with validator nodes. You can run Ethereum staking nodes on NES, and manage their own node keys for continuous benefits.

5 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your NES resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your Huawei Cloud resources.

With IAM, you can use your Huawei Cloud account to create IAM users, and assign permissions to the users to control their access to specific resources. For details about permission configurations, see **Permissions Management**.

You can skip this section if you do not need fine-grained permissions management.

IAM is free of charge. You pay only for the resources in your account.

You can grant users permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you also need to assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant Elastic Cloud Server (ECS) users only the permissions for managing a certain type of ECSs.

NES Permissions

By default, new IAM users do not have any permissions assigned. You must add them to user groups and assign permissions policies or roles to these groups. Users then inherit permissions from the groups. This process is called authorization. Users can perform specified operations on cloud services based on their assigned permissions.

NES is a project-level service deployed for specific regions. To assign NES permissions to a user group, specify the scope as region-specific projects and

select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing NES, the users need to switch to the authorized region.

The following table lists all system-defined policies supported by NES.

Table 5-1 NES system permissions

Role/Policy Name	Description	Permission Type	Dependency
BCS Administrator	Full operation permissions for enhanced Hyperledger Fabric BCS	System-defined role	Tenant Guest, Server Administrator, ELB Administrator, SFS Administrator, SWR Admin, APM FullAccess, AOM FullAccess, CCE Administrator, VPC Administrator, EVS Administrator, and CCE Cluster Admin
BCS NES FullAccess	Full permissions for NES	System-defined policy	None
BCS NES ReadOnlyAccess	Read-only permissions for NES	System-defined policy	None

BCS NES FullAccess Content

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "bcs:nes*:*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

BCS NES ReadOnlyAccess Content

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "bcs:nes*:get*",
                "bcs:nes*:list*"
            ],
```



```
        "Effect": "Allow"
      }
    ]
}
```

6 Billing

6.1 Overview

This section describes the billing mode, billing items, and billing period of NES.

Billing Mode

NES supports the pay-per-use billing model. Fees are postpaid based on the actual usage of each billing item.

Table 1 Billing items lists the billing items. You can use the price calculator to calculate the price.

Billing Items

Table 6-1 Billing items

Billing Item	Type	Category	Node/Package Specifications	Description	Price (USD)	Scenario
Dedicated edition	Full node	Node fee	4U16G	Node of the mainnet and testnet, which is suitable for development.	0.384/ node/ hour	Ethereum, Polygon, Arbitrum, TRON, and BNB Smart Chain
			8U16G	Node of the testnet, which is suitable for development.	0.627/ node/ hour	

Billing Item	Type	Category	Node/Package Specifications	Description	Price (USD)	Scenario
			8U32G	Node of the mainnet and testnet, which is suitable for stable running.	0.768/ node/ hour	
			16 vCPUs | 64 GB	Node of the mainnet and testnet, which is suitable to meet premium performance requirements.	1.536/ node/ hour	
			16U32G	Node of the mainnet and testnet, which is suitable for stable running.	1.254/ node/ hour	
			32U64G	Stable running of the mainnet services	2.508/ node/ hour	
		Storage fee	-	It can be scaled based on the size of the public blockchain ledger data.	0.12/GB/ month	All
		API calling fee	-	Billed based on the time of API calls	4.81/ million calls	
	Full node (Staking supported)	Node fee	8 vCPUs | 32 GB	(Recommended by Ethereum) Node of the mainnet and testnet	1.514/ node/ hour	Ethereum staking

Billing Item	Type	Category	Node/Package Specifications	Description	Price (USD)	Scenario
		Storage fee	-	It can be scaled based on the size of the public blockchain ledger data.	0.12/GB/month	All
Shared edition	Package	Basic Edition (Monthly)	• Number of projects: 10 • Compute units (CUs) per month: 450,000,000 • CUs per second: 400	CUs consumed by API calling each month	Free	Function tests
		Professional Edition (Monthly)	• Number of projects: 20 • CUs per month: 600 million • CUs per second: 990 **NOTE** Excess: USD1.2/million CUs	CUs consumed by API calling each month	49/month	Projects at an early stage
		Enterprise Edition (Monthly)	• Number of projects: 40 • CUs per month: 2.2 billion • CUs per second: 5,000 **NOTE** Excess: USD1.0/million CUs	CUs consumed by API calling each month	289/month	Enterprise-level projects

Billing Item	Type	Category	Node/Package Specifications	Description	Price (USD)	Scenario
		Enterprise Edition (Yearly)	• Number of projects: 40 • CUs per month: 2.2 billion • CUs per second: 5,000 **NOTE** Excess: USD1.0/million CUs	CUs consumed by API calling each month	2,388/year	

Dedicated edition: Total fee = Node fee + Storage fee + API calling fee

Specifically,

- Node fee = Duration x Price (USD)
- Storage fee = Duration x Usage (GB) x Price (USD)
- API calling fee = Number of API calls x Price (USD)

  NOTE

  The number of API calls is the number of requests. Whenever the combined size of the request body and response body reaches 32 KB or the response duration exceeds 500 ms, it is considered as one request. The system then compares the number of requests triggered by each condition and uses the higher value as the total number of API calls. For example, if the sum of the request body size and response body size is 50 KB but the response duration is only 300 ms, the number of API calls is 2.

Shared edition: The fee varies with the package edition.

- Basic edition (monthly): You can use it free of charge for one month and renew it after it expires.

  NOTE

  The package will be invalid if the CUs exceed the package quota. In this case, buy other packages to obtain more quota.

- Professional edition (billed monthly) = USD49 x Number of months

  NOTE

  CUs exceeding the package quota will be charged on a pay-per-use basis. Every 1 million CUs cost USD1.2.

- Enterprise edition (billed monthly) = USD289 x Number of months

  NOTE

  CUs exceeding the package quota will be charged on a pay-per-use basis. Every 1 million CUs cost USD1.0.

- An enterprise edition package costs USD2388 per year.

  NOTE

  CUs exceeding the package quota will be charged on a pay-per-use basis. Every 1 million CUs cost USD1.0.

NOTE

CUs per month = CUs consumed by HTTPS requests + CUs consumed by WebSocket requests

Specifically,

- CUs consumed by each HTTPS API. For details, see the API list in **NES Developer Guide**.
- CUs consumed by each WebSocket API = Number of bytes (usage) x 0.04 (CU)

Billing Period

NES reports service detail records (SDRs) every hour, collects statistics on the usage of all NES resources by hour, and calculates fees based on your usage.

6.2 Billing Cases

Dedicated Edition

Assume that you have purchased an Ethereum node with the specifications of 16 vCPUs | 64 GB, used NES for 1 month (30 days), used 1024 GB memory, and made 3 million API calls. The bill will include:

- Monthly node fee = Duration x Price (USD)

  30 days x 24 hours x USD1.536 = USD1105.92
- Monthly storage fee = Duration x Usage (GB) x Price (USD)

  1 month x 1024 GB x USD0.12 = USD122.88
- Monthly API calling fee = Number of API calls x Price (USD)

  3,000,000 x USD0.00000481 = USD144.3

Total fee = Node fee + Storage fee + API calling fee

In this case, the total fee is USD1373.1.

Shared Edition

- Calculation of the total fee

  Assume that you have purchased a professional edition (monthly) package for one month. The package takes effect at 08:00:00 on January 01, 2024 and expires at 23:59:59 on February 01, 2024. So the package includes 600 million available CUs in current month. By January 25, 2024, you have used 601.005 million CUs. The billing mode is as follows:

  – Package fee = USD49 x Number of months

    USD49 x 1 = USD49

  – Fee of excess CUs = USD1.2 x (Number of used CUs – Number of available CUs in the package)/1,000,000

USD1.2 x (601,005,000 – 600,000,000)/1,000,000 = USD1.2 (round off to one decimal place)

Total fee = Package fee + Fee of excess CUs

In this case, the total fee is USD50.2.

- Calculation of the remaining CUs in the package

  Assume that the CUs in the package are 10,000. The **eth_blockNumber** method is called 10 times via HTTP and consumes 10 CUs each time. The **newHeads** is subscribed to via WebSocket and 1 KB (1024 bytes) of subscription data is received, with each byte consumes 0.04 CUs. The bill will include:

  Number of CUs consumed via HTTP + Number of CUs consumed via WebSocket = Consumed CUs

  10 calls x 10 CUs + 1024 bytes x 0.04 CUs = 140.96 CUs

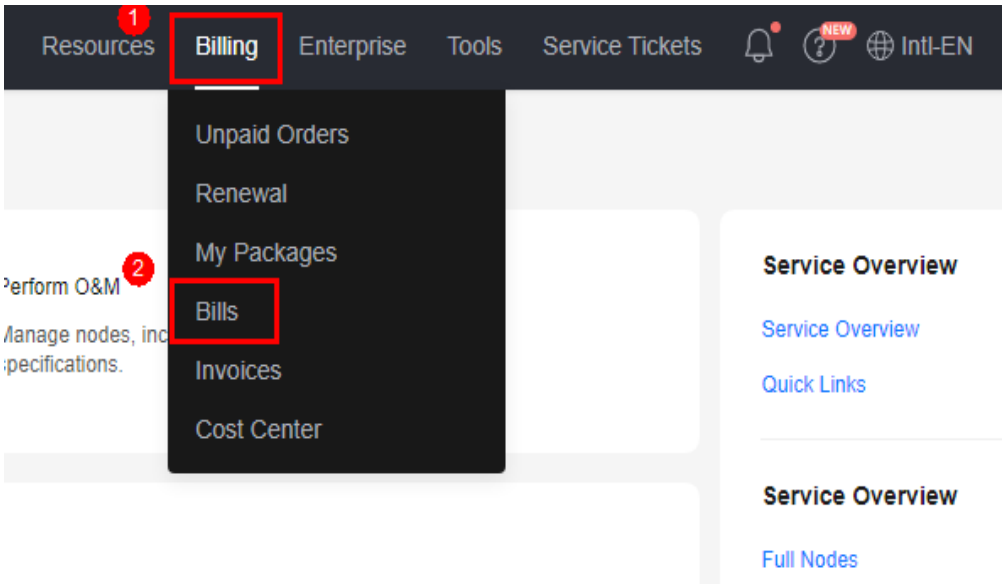
  Remaining CUs = 10,000 – 140.96 = 9859.04

6.3 Bill Query

This section describes how to query bills.

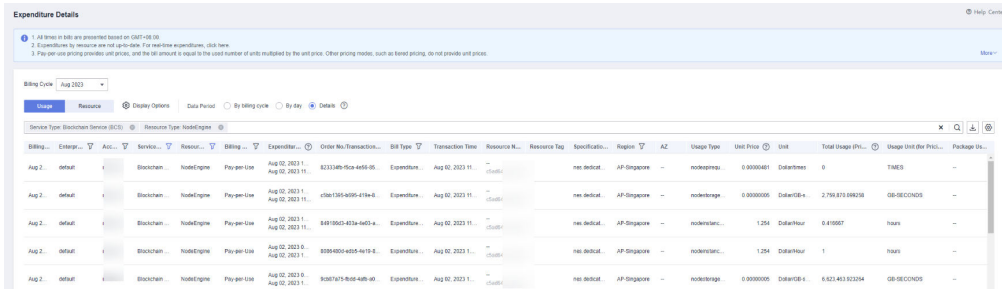
Step 1 Log in to the console.

Step 2 Choose **Billing & Costs** > **Bills** to go to the billing center.



Step 3 Choose **Billing** > **Expenditure Details**.

Step 4 On the **Expenditure Details** page, set **Service Type** to **Blockchain Service (BCS)** and **Resource Type** to **NodeEngine** to view details.


Expenditure Details
Billing Cycle
Aug 2023
Usage
Resource
Display Options
Data Period
By billing cycle
By day
Details

----End

7 Restrictions

Before using NES to host nodes, be familiar with the following precautions:

- Currently, NES is available only in the AP-Singapore region.
- NES supports both WebSocket and HTTP APIs through separate endpoints.
- You can use NES nodes to create batches of JSON-RPC requests.
- You can get an authentication credential with your NES node and construct a node URL credential to establish a connection to the node and send RPC requests.
- To modify the state of the blockchain, you can only use **eth_sendRawTransaction** to send a raw transaction to Ethereum network. This means you cannot use NES as a wallet until you send a transaction. Generate and keep the transaction and private key on a third-party platform until then.

8 Glossary

This section describes common terms used in NES to help you better understand and use NES.

Table 8-1 Glossary

Term	Sub-item	Description
BSC	Mainnet	An EVM-compatible, Proof of Staked Authority (PoSA) mainnet, with shorter block time and lower fees.
	Chapel	As a replica of mainnet, Chapel allows you to develop, test, and deploy your DApps.
Ethereum	Mainnet	It uses Proof-of-Stake (PoS). Mainnet coins have monetary value, incur gas fees, and are recorded on distributed ledgers.
	Sepolia	Sepolia has fewer deployed applications as compared to Goerli. It has a smaller state and history, allowing for faster syncing and requiring minimal disk space to run a node. The validator set is restricted, so not everyone can run a validator node. Ethereum.org recommends Sepolia as the primary choice for testing applications and smart contracts due to its restricted validator set and higher stability guarantees.
	Holesky	The first-ever Ethereum testnet launched on the top of the PoS consensus, serving as a staking, infrastructure, and protocol-developer testnet.
TRON	Mainnet	A primary TRON blockchain network. Transactions on this network have real value.
	Nile	It is used to test new features of TRON, and the code version is generally ahead of the mainnet.
Polygon PoS	Mainnet	The Proof-of-Stake (PoS) mechanism and compatibility with EVMs make mainnet the preferred choice for fast transactions and low costs.

Term	Sub-item	Description
	Mumbai	As a replica of mainnet, Mumbai allows you to develop, test, and deploy your DApps.
Arbitrum	One	An L2 optimistic rollup chain that implements the Arbitrum Rollup protocol and settles to Ethereum's L1 chain. It supports EVMs and lets you efficiently perform transactions at low cost.
	Goerli	A testnet chain that replicates the features of the Arbitrum One mainnet. It is tethered to the Goerli testnet of Ethereum, providing a secure testing ground for developers to experiment with.

