

Intelligent EdgeFabric

Edgectl User Guide

Issue 01
Date 2022-01-11



Copyright © Huawei 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 Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks 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 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.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Overview	1
2 Installing edgectl	3
3 Example: Using edgectl to Run Commands	5
4 edgectl Commands	6
4.1 Querying the edgectl Version	6
4.2 Managing the Product Lifecycle	7
4.2.1 Checking Installation Requirements or Failures	7
4.2.2 Checking Nodes	9
4.2.3 Checking Containerized Application Instances	10
4.2.4 Collecting Node Data	11
4.3 Checking an Item	14
4.3.1 Checking the Hardware Architecture	14
4.3.2 Checking the CPU	15
4.3.3 Checking the Memory	15
4.3.4 Checking Hard Disks	16
4.3.5 Checking the Domain Name Resolution Function	16
4.3.6 Checking Docker	17
4.3.7 Checking Network Connectivity	17
4.3.8 Checking the Installation Status of the IEF Software	18
4.3.9 Checking the Running Status of the IEF Software	19
4.3.10 Checking GPUs	19
4.3.11 Checking NPUs	20
4.3.12 Checking Processes	21
5 Error Codes	22
5.1 Error Code Overview	22
5.2 Common Errors	24
5.2.1 ERROR1001 Invalid parameters	24
5.2.2 ERROR1002 Failed to compress the folder	24
5.2.3 ERROR1003 Failed to decompress the file	24
5.2.4 ERROR1004 Failed to create the folder	25
5.2.5 ERROR1005 Failed to delete the cache directory	25
5.2.6 ERROR1006 Invalid input	26

5.3 Permission Errors.....	26
5.3.1 ERROR2001 No permission.....	26
5.3.2 ERROR2002 No user authorization.....	26
5.4 OS Errors.....	27
5.4.1 ERROR3001 Failed to obtain the hardware architecture.....	27
5.4.2 ERROR3002 Unsupported hardware architecture.....	27
5.4.3 ERROR3101 Failed to obtain the CPU information of the node.....	27
5.4.4 ERROR3102 Failed to obtain the real-time CPU status of the node.....	28
5.4.5 ERROR3103 Insufficient node CPU.....	28
5.4.6 ERROR3201 Failed to obtain the memory information of the node.....	28
5.4.7 ERROR3202 Insufficient node memory.....	28
5.4.8 ERROR3301 Failed to obtain the disk information of the node.....	29
5.4.9 ERROR3302 Insufficient disk space on the node.....	29
5.4.10 ERROR3401 Failed to resolve the domain name.....	30
5.4.11 ERROR3501 Failed to check the network.....	30
5.4.12 ERROR3601 Failed to obtain the process information of the node.....	31
5.4.13 ERROR3602 Failed to obtain the total number of processes on the node.....	31
5.4.14 ERROR3603 Number of remaining processes on the node is insufficient.....	31
5.5 IEF Software Errors.....	32
5.5.1 ERROR5001 Incomplete IEF software installation.....	32
5.5.2 ERROR5002 IEF software is not completely running.....	32
5.5.3 ERROR5003 Failed to read the IEF software configuration file.....	33
5.5.4 ERROR5004 Docker is not enabled for the IEF software.....	33
5.5.5 ERROR5005 Failed to start the IEF software.....	33
5.5.6 ERROR5006 Failed to stop the IEF software.....	34
5.5.7 ERROR5101 No application data found.....	35
5.5.8 ERROR5102 No status data of the application found.....	35
5.5.9 ERROR5103 Container is not started.....	35
5.5.10 ERROR5104 Container exits abnormally.....	36
5.5.11 ERROR5105 Failed to pull the container image.....	36
5.5.12 ERROR5106 Container exits.....	36
5.6 Third-Party Dependency Errors.....	36
5.6.1 ERROR6101 Docker does not exist.....	37
5.6.2 ERROR6102 Docker is not running.....	37
5.6.3 ERROR6103 Incorrect Docker Cgroup Driver configuration.....	37
5.6.4 ERROR6201 No GPUs available.....	38
5.6.5 ERROR6202 The GPU driver is not installed.....	38
5.6.6 ERROR6203 The GPU driver is not started.....	39
5.6.7 ERROR6301 No NPUs available.....	39
5.7 Data Collection Errors.....	39
5.7.1 ERROR7001 Failed to collect fault data.....	39
5.8 Internal Errors.....	40

5.8.1 ERROR8001 Unexpected error..... 40

1 Overview

edgectl is an O&M tool that helps you diagnose faults at the edge side. You can use edgectl to locate common problems such as IEF software installation failures, IEF software faults, and application faults. This tool also allows you to collect all IEF-related data on nodes by one click for remote fault diagnosis. In addition, you can use edgectl to migrate IEF software data to other available nodes when the node is faulty.

Concepts

- **diagnose:** Diagnoses a specific fault scenario to locate the fault cause.
- **collect:** Collects all IEF-related information for IEF O&M personnel to locate and resolve problems.
- **check:** Checks whether a system meets the installation and running requirements of the IEF software.

Commands

Table 1-1 Commands

Category	Syntax	Description
Querying the edgectl version	edgectl version	List the version information about edgectl.
Managing the product lifecycle	edgectl diagnose installation	Check whether a node meets the requirements of the IEF edge software.
	edgectl diagnose node-fault	Diagnose a node fault.
	edgectl diagnose pod	Diagnose the fault cause of a specific pod.
	edgectl collect	Collect IEF-related data on a node.
Checking an item	edgectl check arch	Check whether IEF supports the node hardware architecture.

Category	Syntax	Description
	<code>edgectl check cpu</code>	Check whether the number of CPU cores meets the IEF requirements.
	<code>edgectl check memory</code>	Check the total and remaining memory size of a node to determine whether the memory size meets the IEF requirements.
	<code>edgectl check disk</code>	Check whether the system hard disks meet the IEF requirements.
	<code>edgectl check dns</code>	Check whether the domain name resolution function of a node is normal.
	<code>edgectl check docker</code>	Check whether the Docker function of a node is normal.
	<code>edgectl check network</code>	Check whether a node can communicate with IEF cloud endpoints.
	<code>edgectl check ief-system-configure</code>	Check whether IEF is correctly installed.
	<code>edgectl check ief-system-status</code>	Check the running status of the IEF software.
	<code>edgectl check gpu</code>	Check whether a node is equipped with GPUs and whether the GPU driver is installed and running properly.
	<code>edgectl check npu</code>	Check whether a node is equipped with NPUs.
	<code>edgectl check pid</code>	Check whether there are too many processes. If the percentage of available processes is less than 5%, the number of processes is regarded as insufficient.

2 Installing edgectl

Downloading edgectl

Check whether the edgectl package exists in the `/opt/edge-installer` directory. If the package exists, you do not need to download it again. If the package does not exist, log in to the IEF console and download the tool of the corresponding version, for example, `edgectl_x.x.x_x86_64.tar.gz`, from **Edge O&M Tool** on the right of the **Dashboard** page. `edgectl` in the downloaded package is the edge O&M tool.

edgectl supports the Linux operating systems that use the x86_64, Arm64, or Arm32 hardware architecture. It is installation-free. You only need to download it and place it in any directory of the operating system.

Configuring Environment Variables

If edgectl already exists in the `/opt/edge-installer` directory, run the following command to grant the execute permission on edgectl:

```
export PATH=/opt/edge-installer:$PATH
```

```
chmod +x /opt/edge-installer/edgectl
```

If the edgectl package is manually downloaded, perform the following steps to decompress the package and grant the execute permission on edgectl:

Step 1 Decompress the edgectl package.

```
sudo mkdir -p DIR_TO_IEF
```

```
sudo tar -zxvf edgectl_x.x.x_x86_64.tar.gz -C DIR_TO_IEF
```

In the preceding command, `DIR_TO_IEF` indicates the directory where the edgectl package is decompressed, and `edgectl_x.x.x_x86_64.tar.gz` indicates the name of the edgectl package. Replace them with the actual values.

Step 2 Set the path of edgectl as an environment variable and grant the execute permission on the path.

```
export PATH= DIR_TO_IEF:$PATH
```

```
chmod +x DIR_TO_IEF/edgectl  
----End
```

3 Example: Using edgectl to Run Commands

edgectl provides the functions of preliminarily diagnosing edge node faults, migrating nodes, and collecting data in typical scenarios.

Prerequisites

edgectl has been installed. If not, install edgectl by referring to [Installing edgectl](#).

Checking Whether IEF Edge Software Can Be Installed on a Node

The IEF edge software is required for performing IEF functions at the edge.

You can run the following command to check whether your edge node meets the requirements for installing the IEF edge software:

```
edgectl diagnose installation
```

Identify the cause of a node fault.

When a user uses the IEF function at the edge, the node may be disconnected from the cloud due to the environment of the edge node.

The following example shows how to check whether the IEF edge software is faulty.

```
edgectl diagnose node-fault
```

Collecting Node Data

If a fault that cannot be rectified occurs on a node and needs to be handled by IEF O&M personnel, you can run the following command to collect IEF-related data of the edge node in one-click mode and then send the data to IEF O&M personnel:

```
edgectl collect -p /tmp/data/
```

4 edgectl Commands

4.1 Querying the edgectl Version

Function

This command is used to check the software version and compilation platform of edgectl.

Syntax

edgectl version

version can be abbreviated as **ver**, that is:

edgectl ver

Parameters

Table 4-1 Parameter description

Parameter	Mandatory	Short hand	Description
--short	No	-s	Whether to display only the version number.
--output	No	-o	Output format. The value can be json or yaml .

Example

Query the software version of edgectl.

edgectl version

The following is an example of the command output:

Version: &version.VersionInfo{Version:"1.0.0", GitCommit:"02e24c38a0120e5738d63e8289bc88424cdad35e", GoVersion:"go1.12.1", Compiler:"gc", Platform:"linux/amd64"}

4.2 Managing the Product Lifecycle

4.2.1 Checking Installation Requirements or Failures

Function

This command is used to check whether a node meets the IEF software installation requirements. If the node does not meet the requirements, a message is displayed, indicating the reasons. The command can also be used to locate the cause of an IEF software installation failure.

Syntax

edgectl diagnose installation [params] CHECKITEMS

- **diagnose** can be abbreviated as **diag**.
- **installation** can be abbreviated as **install**.

That is, the following abbreviated commands are supported:

edgectl diag installation [params] CHECKITEMS

edgectl diag install [params] CHECKITEMS

Parameters

Table 4-2 Parameter description

Parameter	Mandatory	Short hand	Description
CHECKITEMS	No	N/A	<p>Check item list. The following values are supported:</p> <ul style="list-style-type: none"> • default: Check the hardware architecture, CPU, memory, hard disks, domain name resolution, network connectivity, and Docker. • arch: Check whether the hardware architecture is supported by IEF. • cpu: Check whether the CPU meets the IEF requirements. • mem or memory: Check whether the memory meets the IEF requirements. • disk: Check whether the hard disks meet the IEF requirements. • dns: Check whether the domain name resolution function is normal. • network: Check whether the node can be connected to IEF. • docker: Check whether Docker is running properly. • gpu or nvidia: Check whether the node is equipped with GPUs and whether the GPU driver is running properly. • npu: Checks whether the node is equipped with NPUs. <p>The check items are separated by commas (,). For example: edgectl diag install [params] cpu,disk Check items can also be deleted by adding a hyphen (-) before a check item. For example: edgectl diag install [params] default,-docker</p>
--domain-name	No	No	Domain name to be resolved. It is required if you configure edgectl to check the domain name resolution function of the node.
--edge-hub-url	No	No	Endpoint address of IEF. It is used to check the connectivity between a node and IEF.
--node-id	No	No	Node ID. The default value is fake-node-id .

Parameter	Mandatory	Short hand	Description
--project-id	No	None	Project ID. The default value is fake-project-id .
--detail	No	-x	Whether to print internal logs.

Example

Check whether a node meets the IEF software installation requirements.

```
edgectl diagnose installation
```

If the check is successful, the following information is displayed:

```
+-----+
| Installation diagnose succeed. |
+-----+
```

If the check fails, the information similar to the following is displayed:

```
Failed to get Docker version.
ERROR6102: Make sure that Docker is running.
For more information, see IEF documentation.

+-----+
| Installation diagnose failed. |
+-----+
```

In addition, an error code is displayed. You can obtain the help information from the documentation provided in the error message.

4.2.2 Checking Nodes

Function

This command is used to locate the cause of a fault on the edge node where the IEF software has been installed.

Syntax

```
edgectl diagnose node-fault [params]
```

- **diagnose** can be abbreviated as **diag**.
- **node-fault** can be abbreviated as **node** or **no**.

That is, the following abbreviated commands are supported:

```
edgectl diag node-fault [params]
```

```
edgectl diag node [params]
```

```
edgectl diag no [params]
```

Parameters

Table 4-3 Parameter description

Parameter	Mandatory	Short hand	Description
--detail	No	-x	Whether to print internal logs.

Example

Check the cause of a node fault.

```
edgectl diagnose node-fault
```

If the check is successful, the following information is displayed:

```
+-----+
| Node diagnose succeed. |
+-----+
```

If the check fails, the information similar to the following is displayed:

```
Failed to get Docker version.
ERROR6102: Make sure that Docker is running.
For more information, see IEF documentation.

+-----+
| Node diagnose failed. |
+-----+
```

In addition, an error code is displayed. You can obtain the help information from the documentation provided in the error message.

4.2.3 Checking Containerized Application Instances

Function

This command is used to check the cause of a fault occurred on a containerized application instance on a node.

Syntax

```
edgectl diagnose pod [params]PODNAME
```

- **diagnose** can be abbreviated as **diag**.
- **pod** can be abbreviated as **po**.

That is, the following abbreviated commands are supported:

```
edgectl diag pod [params]PODNAME
```

```
edgectl diag po [params]PODNAME
```


Parameters

Table 4-4 Parameter description

Parameter	Mandatory	Shorthand	Description
PODNAME	Yes	N/A	Name of a containerized application, that is, the name of the application created on the IEF console.
--detail	No	-x	Whether to print internal logs.

Precautions

PODNAME specifies the name of a containerized application, that is, the name of the application created on the IEF console.

This command diagnoses all the pods whose prefix matches the specified pod name.

Example

Diagnose application exception causes.

```
Edgectl diagnose pod edge-cs
```

If the check is successful, the information similar to the following is displayed:

```
table `meta` already exists, skip
Pod edge-cs-759d46f887-p9kgf is Ready.
```

```
+-----+
| Pod diagnose succeed. |
+-----+
```

If the check fails, the information similar to the following is displayed:

```
table `meta` already exists, skip
Failed to get Docker version.
ERROR6102: Make sure that Docker is running.
For more information, see IEF documentation.
```

```
+-----+
| Pod diagnose failed. |
+-----+
```

In addition, an error code is displayed. You can obtain the help information from the documentation provided in the error message.

4.2.4 Collecting Node Data

Function

This command is used to collect all data related to IEF on a node for IEF O&M personnel to locate faults.

The following table lists the data to be collected.

Table 4-5 Data to be collected

Category	Item	Description
System data	Hardware architecture	Collect the arch command output and determine the type of the installed IEF software.
	CPU information	Parse the /proc/cpuinfo file and generate a CPU information file.
	Memory information	Collect the free -h command output.
	Hardware information	Collect the outputs of the df -h and mount commands.
	Network information	Collect the netstat -anp command output and copy the /etc/resolv.conf and /etc/hosts files.
	Process information	Collect the ps -aux command output.
	Time information	Collect the outputs of the date and uptime commands.
	Historical commands	Collects all commands entered by the user.
IEF data	IEF database data	Copy the /opt/IEF/Edge-core/bin/edge.db file.
	IEF log files	Copy all files in the /var/IEF/sys/log directory.
	IEF service files	Copy the edgecore.service , edgelogger.service , edgemonitor.service , and edgedaemon.service files in the /lib/systemd/system/ directory.
	IEF software version	Copy the /opt/IEF/version file.
	IEF certificates	Copy all files in the /opt/IEF/Cert/ directory.
	IEF encryption materials	Copy all files in the /opt/material/ directory.
	Edge-Core (including Edge-daemon) configuration file	Copy all files in the /opt/IEF/Edge-core/conf/ directory.
	Edge-Logger configuration file	Copy all files in the /opt/IEF/Edge-logger/conf/ directory.
	Edge-Monitor configuration file	Copy all files in the /opt/IEF/Edge-monitor/conf/ directory.

Category	Item	Description
Docker data (required if Docker is enabled in IEF)	Docker version information	Collect the docker version command output.
	Docker information	Collect the docker info command output.
	Docker log information	Collect the journalctl -u docker.service command output.
	Docker container information	Collect the docker ps -a command output.
	Docker container configurations and logs	Copy all files in the /var/lib/docker/containers directory.
	Docker image information	Collect the docker images command output.
	GPU device information	Collect the /dev/nvidiactl /dev/nvidia-uvm /dev/nvidia? 2>&1 command output.
	GPU kernel module information	Collect the lsmod grep -e nvidia -e nvidia-uvm 2>&1 command output.
	GPU information	Collect the /var/IEF/nvidia/bin/nvidia-smi 2>&1 command output.
	GPU driver status information	Collect the systemctl status nvidia-drivers-loader command output.
	NPU device information	Collect the ls /dev/davinci_manager /dev/hisi_hdc /dev/davinci? command output.

Syntax

edgectl collect [params]

Parameters

Table 4-6 Parameter description

Parameter	Mandatory	Short hand	Description
--path	No	-p	Directory for caching IEF data and storing IEF data packages. The default value is the current directory.
--detail	No	-x	Whether to print internal logs.

Precautions

- The command must be run as user **root**. Otherwise, the command will fail.
- During the command execution, you will be prompted to enter **yes** or **y** to confirm the node data collection.
- The collected data is stored in the package named **data_{timestamp}.tar.gz** in a specific directory.

Example

Collect IEF data.

```
edgectl collect
```

If the following information is displayed, enter **yes** or **y**:

```
Do you agree that edgectl collects data on this machine (yes/no)? y
```

The following is an example of the command output:

```
Data packet name: data_2020-03-18-10-47-47.tar.gz.
```

```
+-----+  
| Data collection process succeed. |  
+-----+
```

The name of the cached package is displayed in the command output.

4.3 Checking an Item

4.3.1 Checking the Hardware Architecture

Function

This command is used to check whether IEF supports the hardware architecture used by a node.

Syntax

```
edgectl check arch
```

Parameters

None

Example

Check the hardware architecture.

```
edgectl check arch
```

The following is an example of the command output:

```
The node architecture is x86_64 and supports installation of IEF software using x86_64.
```

```
| System architecture check succeed. |  
+-----+
```

The command output displays the hardware architecture of the node and the architecture in which the IEF software will be installed.

4.3.2 Checking the CPU

Function

This command is used to check whether the number of CPU cores meets the IEF requirements.

Syntax

```
edgectl check cpu
```

Parameters

None

Example

Check the CPU information.

```
edgectl check cpu
```

The following is an example of the command output:

```
The node has 8 CPUs. CPU usage: 0.25%.
```

```
+-----+  
| CPU check succeed. |  
+-----+
```

4.3.3 Checking the Memory

Function

This command is used to check the total and remaining sizes of the system memory to determine whether the memory meets the IEF requirements.

Syntax

```
edgectl check memory
```

memory can be abbreviated as **mem**, that is:

```
edgectl check mem
```

Parameters

None

Example

Check the memory.

```
edgectl check mem
```

The following is an example of the command output:

```
Total memory: 32011 MB; Free memory: 537 MB; Used percent: 54.97%.
```

```
+-----+  
| Memory check succeed. |  
+-----+
```

4.3.4 Checking Hard Disks

Function

This command is used to check whether the system hard disks meet the IEF requirements.

Syntax

```
edgectl check disk
```

Parameters

None

Example

Check hard disks.

```
edgectl check disk
```

The following is an example of the command output:

```
IEF runtime (/opt/): Total space: 42141691904 B; Used%: 67.0%; Total inodes: 2616320; Used%: 8.2%.
```

```
IEF log (/var/): Total space: 42141691904 B; Used%: 67.0%; Total inodes: 2616320; Used%: 8.2%.
```

```
+-----+  
| Disk check succeed. |  
+-----+
```

4.3.5 Checking the Domain Name Resolution Function

Function

This command is used to check whether the domain name resolution function of a node is normal.

Syntax

```
edgectl check dns [params]
```

Parameters

Table 4-7 Parameter description

Parameter	Mandatory	Shorthand	Description
--domain-name	No	-d	Domain name to be resolved by edgectl.

Example

Check the domain name resolution function.

```
edgectl check dns
```

The following is an example of the command output:

4.3.6 Checking Docker

Function

This command is used to check whether Docker runs normally on a node.

Syntax

```
edgectl check docker
```

Parameters

None

Example

Check whether Docker runs normally on a node.

```
edgectl check docker
```

The following is an example of the command output:

```
+-----+
| Docker check succeed. |
+-----+
```

4.3.7 Checking Network Connectivity

Function

This command is used to check whether a node can communicate with IEF cloud endpoints.

Syntax

edgectl check network [params]

network can be abbreviated as **net**, that is:

edgectl check net [params]

Parameters

Table 4-8 Parameter description

Parameter	Mandatory	Short hand	Description
--edge-hub-url	No	No	Endpoint address of IEF. It is used to check the connectivity between a node and IEF.
--project-id	No	No	Project ID. The default value is fake-project-id .
--node-id	No	No	Node ID. The default value is fake-node-id .

Example

Check the network connectivity of the node.

```
edgectl check network
```

The following is an example of the command output:

In the preceding example, the IEF cloud endpoint returns response code 401 because the request does not carry correct node ID, project ID, and node certificate. This is not a network problem, which does not affect the communication between the node and the IEF cloud endpoint.

4.3.8 Checking the Installation Status of the IEF Software

Function

This command is used to check whether the IEF software is correctly installed on a node.

Syntax

edgectl check ief-system-configure

ief-system-configure can be abbreviated as **ief**, that is:

edgectl check ief

Parameters

None

Example

Check whether the IEF software is installed on the node.

```
edgectl check ief
```

The following is an example of the command output:

4.3.9 Checking the Running Status of the IEF Software

Function

This command is used to check the running status of the IEF software.

Syntax

```
edgectl check ief-system-status
```

ief-system-status can be abbreviated as **ief-status** or **ief-stat**.

```
edgectl check ief-status
```

```
edgectl check ief-stat
```

Parameters

None

Example

Check whether the IEF software on the node is running properly.

```
edgectl check ief-system-status
```

The following is an example of the command output:

```
+-----+  
| IEF system status check succeed. |  
+-----+
```

4.3.10 Checking GPUs

Function

This command is used to check whether a node is equipped with GPUs and whether the GPU driver is installed and running properly.

Syntax

```
edgectl check gpu
```

Parameters

None

Example

Check GPUs of the node.

```
edgectl check gpu
```

If the check is successful, the following information is displayed:

```
+-----+  
| NVIDIA check succeed. |  
+-----+
```

If the check fails, the information similar to the following is displayed:

```
ERROR6201: Confirm that this node has NVIDIA devices.  
For more information, see IEF documentation.
```

```
+-----+  
| NVIDIA check failed. |  
+-----+
```

If the check item fails, an error code is displayed. You can obtain the help information from the documentation provided in the error message.

4.3.11 Checking NPUs

Function

This command is used to check whether a node is equipped with NPUs.

Syntax

```
edgectl check npu
```

Parameters

None

Example

Check NPUs of the node.

```
edgectl check npu
```

The following is an example of the command output:

```
ERROR6301: please confirm that this node has npu device.  
npu device check failed.  
npu check failed.
```

If the check is successful, the following information is displayed:

```
+-----+  
| NPU check succeed. |  
+-----+
```

If the check fails, the information similar to the following is displayed:

```
ERROR6301: Confirm that this node has NPU devices.  
For more information, see IEF documentation.
```

```
+-----+  
| NPU check failed. |  
+-----+
```

If the check item fails, an error code is displayed. You can obtain the help information from the documentation provided in the error message.

4.3.12 Checking Processes

Function

This command is used to check whether there are too many processes. If the percentage of available processes is less than 5%, the number of processes is insufficient.

Syntax

```
edgectl check pid
```

Parameters

None

Example

Check the number of processes on the node.

```
edgectl check pid
```

The following is an example of the command output:

```
Maximum PIDs: 32768; Running processes: 146.
```

```
+-----+  
| PID check succeed. |  
+-----+
```

5 Error Codes

5.1 Error Code Overview

Common Errors

- [ERROR1001 Invalid parameters](#)
- [ERROR1002 Failed to compress the folder](#)
- [ERROR1003 Failed to decompress the file](#)
- [ERROR1004 Failed to create the folder](#)
- [ERROR1005 Failed to delete the cache directory](#)
- [ERROR1006 Invalid input](#)

Permission Errors

- [ERROR2001 No permission](#)
- [ERROR2002 No user authorization](#)

OS Errors

- [ERROR3001 Failed to obtain the hardware architecture](#)
- [ERROR3002 Unsupported hardware architecture](#)
- [ERROR3101 Failed to obtain the CPU information of the node](#)
- [ERROR3102 Failed to obtain the real-time CPU status of the node](#)
- [ERROR3103 Insufficient node CPU](#)
- [ERROR3201 Failed to obtain the memory information of the node](#)
- [ERROR3202 Insufficient node memory](#)
- [ERROR3301 Failed to obtain the disk information of the node](#)
- [ERROR3302 Insufficient disk space on the node](#)

ERROR3401 Failed to resolve the domain name

ERROR3501 Failed to check the network

ERROR3601 Failed to obtain the process information of the node

ERROR3602 Failed to obtain the total number of processes on the node

ERROR3603 Number of remaining processes on the node is insufficient

IEF Software Errors

ERROR5001 Incomplete IEF software installation

ERROR5002 IEF software is not completely running

ERROR5003 Failed to read the IEF software configuration file

ERROR5004 Docker is not enabled for the IEF software

ERROR5005 Failed to start the IEF software

ERROR5006 Failed to stop the IEF software

ERROR5101 No application data found

ERROR5102 No status data of the application found

ERROR5103 Container is not started

ERROR5104 Container exits abnormally

ERROR5105 Failed to pull the container image

ERROR5106 Container exits

Third-Party Dependency Errors

ERROR6101 Docker does not exist

ERROR6102 Docker is not running

ERROR6103 Incorrect Docker Cgroup Driver configuration

ERROR6201 No GPUs available

ERROR6202 The GPU driver is not installed

ERROR6203 The GPU driver is not started

ERROR6301 No NPUs available

Data Collection Errors

ERROR7001 Failed to collect fault data

Internal Errors

ERROR8001 Unexpected error

5.2 Common Errors

5.2.1 ERROR1001 Invalid parameters

Description

Invalid parameters.

Possible Causes

- A mandatory parameter is not specified.
- The entered parameters are invalid.

Solution

Modify the parameters by referring to the parameter description of the corresponding command in the *edgectl User Guide*.

5.2.2 ERROR1002 Failed to compress the folder

Description

Failed to compress the folder.

Possible Causes

The disk space of the cache directory is insufficient.

Solution

Run the following command:

```
df -h
```

Check whether the disk space of the cache directory is sufficient. If insufficient, move the cache directory to a directory with sufficient space.

5.2.3 ERROR1003 Failed to decompress the file

Description

Failed to decompress the file.

Possible Causes

- The disk space of the cache directory is insufficient.
- The package is damaged.

Solution

Step 1 Check whether the disk space is insufficient.

Run the following command:

```
df -h
```

Check whether the disk space of the cache directory is sufficient. If insufficient, move the cache directory to a directory with sufficient space.

Step 2 Check whether the package is damaged.

Run the following command to manually decompress the package:

```
mkdir -p test; tar zxvf TarName -C test
```

In the preceding command, **TarName** indicates the actual name of the package. Check whether the package can be decompressed.

Step 3 If the fault is not caused by the preceding reasons, rectify the fault based on the error information reported by edgectl.

----End

5.2.4 ERROR1004 Failed to create the folder

Description

Failed to create the folder.

Possible Causes

The disk space of the cache directory is insufficient.

Solution

Run the following command:

```
df -h
```

Check whether the disk space of the cache directory is sufficient. If insufficient, move the cache directory to a directory with sufficient space.

5.2.5 ERROR1005 Failed to delete the cache directory

Description

Failed to delete the cache directory.

Possible Causes

Processing by edgectl is abnormal.

Solution

Manually delete the cache directory.

5.2.6 ERROR1006 Invalid input

Description

An exception occurs when edgectl reads user input.

Possible Causes

Processing by edgectl is abnormal.

Solution

Run the corresponding command again. If the error persists, contact IEF engineers.

5.3 Permission Errors

5.3.1 ERROR2001 No permission

Description

No permission to run the edgectl command.

Possible Causes

edgectl is not executed as user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

You can run the following command to switch to user **root**:

```
su
```

The system then prompts you to type the password of user **root**.

5.3.2 ERROR2002 No user authorization

Description

edgectl is not authorized by the user.

Possible Causes

The user does not enter **yes** when run edgectl to collect data.

Solution

Enter **yes** to authorize edgectl to collect node data.

5.4 OS Errors

5.4.1 ERROR3001 Failed to obtain the hardware architecture

Description

edgectl cannot obtain the hardware architecture of the node.

Possible Causes

edgectl is not run by user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

5.4.2 ERROR3002 Unsupported hardware architecture

Description

The hardware architecture of the node does not support the installation of the IEF software.

Possible Causes

The node hardware architecture is not x86, Arm32, or Arm64.

Solution

Install the IEF software on the node using x86, Arm32, or Arm64.

5.4.3 ERROR3101 Failed to obtain the CPU information of the node

Description

edgectl cannot obtain the CPU information of the node.

Possible Causes

edgectl is not run by user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

5.4.4 ERROR3102 Failed to obtain the real-time CPU status of the node

Description

edgectl cannot obtain the real-time CPU status of the node.

Possible Causes

edgectl is not run by user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

5.4.5 ERROR3103 Insufficient node CPU

Description

edgectl detects that the number of CPU cores on the node is insufficient for installing the IEF software.

Possible Causes

The number of CPU cores on the node is insufficient.

Solution

Install the IEF software on the node with sufficient CPU cores.

5.4.6 ERROR3201 Failed to obtain the memory information of the node

Description

edgectl cannot obtain the memory information of the node.

Possible Causes

edgectl is not run by user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

5.4.7 ERROR3202 Insufficient node memory

Description

edgectl detects that the memory of the node is insufficient for installing the IEF software.

Possible Causes

Too much software is running on the node.

Solution

Expand the node memory, or stop unnecessary processes.

5.4.8 ERROR3301 Failed to obtain the disk information of the node

Description

edgectl cannot obtain the hard disk information of the node.

Possible Causes

edgectl is not run by user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

5.4.9 ERROR3302 Insufficient disk space on the node

Description

edgectl detects that the disk space of the node is insufficient for installing the IEF software.

Possible Causes

The disk space of the corresponding directory is insufficient.

Solution

The IEF software will be installed on the **/opt** and **/var** directories on the node. The error message will prompt the directory whose disk space is insufficient. To rectify the fault, perform the following steps:

Step 1 Expand the capacity of the corresponding hard disk.

Step 2 Run the following command to mount the corresponding directory to a directory with large available space:

```
mount /dev/xvde1 /opt
```

Step 3 Delete unnecessary files from the hard disk.

----End

5.4.10 ERROR3401 Failed to resolve the domain name

Description

edgectl failed to resolve the domain name.

Possible Causes

- The domain name does not exist.
- The DNS server is not configured on the node.
- The node cannot connect to the DNS server.
- The DNS server does not work.

Solution

To rectify the fault, perform the following steps:

Step 1 Check whether the domain name exists.

Check whether the domain name can be resolved on the node where the domain name resolution service is normal. If the domain name cannot be resolved, change it to another domain name.

Step 2 Check whether the DNS server is configured on the node.

On the node, open the **/etc/resolv.conf** file and check whether the DNS server is configured. If not, configure the DNS server. For example, add the following line about the IP address (114.114.114.114) of the DNS server on the public network to the **/etc/resolv.conf** file, and save the file.

```
nameserver 114.114.114.114
```

Step 3 Check whether the node can connect to the DNS server.

On the node, open the **/etc/resolv.conf** file, obtain the IP address of the DNS server, and ping the IP address.

```
ping 114.114.114.114
```

Step 4 Check whether the DNS server works properly.

If the DNS server does not work, replace the DNS server or contact the maintenance personnel of the DNS server to restore the DNS server.

----End

5.4.11 ERROR3501 Failed to check the network

Description

edgectl cannot obtain responses from IEF.

Possible Causes

The node cannot connect to IEF.

Solution

Check whether the node can resolve the IEF domain name.

- If the domain name cannot be resolved to an IP address, check whether the domain name resolution service is normal by referring to [ERROR3401 Failed to resolve the domain name](#).
- If the domain name can be resolved to an IP address, check the network connection and check whether the IEF domain name and port are blocked (for example, by the security group rule set for the VM).

5.4.12 ERROR3601 Failed to obtain the process information of the node

Description

edgectl cannot obtain the process information of the node.

Possible Causes

edgectl is not run by user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

5.4.13 ERROR3602 Failed to obtain the total number of processes on the node

Description

edgectl cannot obtain the total number of processes on the node.

Possible Causes

edgectl is not run by user **root**.

Solution

Switch to user **root**, and run the edgectl command again.

5.4.14 ERROR3603 Number of remaining processes on the node is insufficient

Description

edgectl detects that the number of remaining processes on the node is insufficient.

Possible Causes

Too much software is running on the node.

Solution

Run the following command to view all processes on the node:

```
ps -aux
```

Stop unnecessary software on the node.

5.5 IEF Software Errors

5.5.1 ERROR5001 Incomplete IEF software installation

Description

The IEF software is not installed on the node, or some files of the IEF software are lost.

Possible Causes

The IEF software on the node is uninstalled by mistake.

Solution

Reinstall the IEF software.

5.5.2 ERROR5002 IEF software is not completely running

Description

The IEF software is not completely running on the node.

Possible Causes

The IEF software is manually stopped on the node.

Solution

To rectify the fault, perform the following steps:

1. Check the running status of Edge-Core.
systemctl status edgecore
If Edge-Core is not started, start it.
systemctl start edgecore
2. Check the running status of Edge-Logger.
systemctl status edgelogger
If Edge-Logger is not started, start it.
systemctl start edgelogger
3. Check the running status of Edge-Monitor.
systemctl status edgemonitor

If Edge-Monitor is not started, start it.

```
systemctl start edgemonitor
```

4. Check the running status of Edge-Daemon.

```
systemctl status edgedaemon
```

If Edge-Daemon is not started, start it.

```
systemctl start edgedaemon
```

5.5.3 ERROR5003 Failed to read the IEF software configuration file

Description

edgectl failed to read the user configuration file `/opt/IEF/Cert/user_config` of the IEF software on the node.

Possible Causes

The `/opt/IEF/Cert/user_config` file is deleted or manually modified.

Solution

Reinstall the IEF software.

5.5.4 ERROR5004 Docker is not enabled for the IEF software

Description

Docker is not enabled in the IEF software configuration of the node.

Possible Causes

Docker is not enabled when the node is registered.

Solution

Check whether Docker needs to be installed on the node. If yes, reinstall the IEF software on the node and enable Docker.

5.5.5 ERROR5005 Failed to start the IEF software

Description

The IEF software on the node fails to be started.

Possible Causes

- The IEF software is deleted by mistake.
- The IEF software cannot be started.

Solution

Check whether the corresponding IEF software exists.

systemctl status edgecore

systemctl status edgelogger

systemctl status edgemonitor

systemctl status edgedaemon

- If yes, manually start Edge-Core, Edge-Logger, Edge-Monitor, and Edge-Daemon.

systemctl start edgecore

systemctl start edgelogger

systemctl start edgemonitor

systemctl start edgedaemon

- If any software cannot be started, [submit a service ticket](#).

5.5.6 ERROR5006 Failed to stop the IEF software

Description

The IEF software on the node fails to be stopped.

Possible Causes

The IEF software is deleted by mistake.

Solution

Check whether the corresponding IEF software exists.

systemctl status edgecore

systemctl status edgelogger

systemctl status edgemonitor

systemctl status edgedaemon

- If yes, manually stop Edge-Core, Edge-Logger, Edge-Monitor, and Edge-Daemon.

systemctl stop edgecore

systemctl stop edgelogger

systemctl stop edgemonitor

systemctl stop edgedaemon

- If any software cannot be stopped, [submit a service ticket](#).

5.5.7 ERROR5101 No application data found

Description

Data of a specified application is not found on the node.

Possible Causes

- The application name is incorrect.
- The IEF software is faulty.

Solution

Step 1 Check whether the application name is correct.

Step 2 Run the following command to check whether the IEF software is faulty:

```
edgectl diag node
```

Rectify the fault based on the error information.

----End

5.5.8 ERROR5102 No status data of the application found

Description

The status data of a specified application is not found on the node.

Possible Causes

The IEF software is processing the application.

Solution

Wait for several minutes and run the edgectl command again.

If the issue persists, [submit a service ticket](#).

5.5.9 ERROR5103 Container is not started

Description

The container of the application deployed on the node is not started.

Possible Causes

The Docker version is incorrect.

Solution

Determine the cause of the startup failure based on the error message and rectify the fault.

5.5.10 ERROR5104 Container exits abnormally

Description

The container of the application deployed on the node exits abnormally.

Possible Causes

An exception occurs on the application.

Solution

View the logs of the container to locate the cause.

5.5.11 ERROR5105 Failed to pull the container image

Description

The container image that will be used to deploy the application on the node fails to be pulled.

Possible Causes

- The Docker configuration is incorrect.
- The network is faulty, and the node cannot connect to SWR.

Solution

[Submit a service ticket.](#)

5.5.12 ERROR5106 Container exits

Description

The container of the application deployed on the node exits.

Possible Causes

An error occurs on the container image.

Solution

View the logs of the container to locate the cause.

5.6 Third-Party Dependency Errors

5.6.1 ERROR6101 Docker does not exist

Description

Docker commands are unavailable on the node.

Possible Causes

Docker is not installed.

Solution

Use the APT or Yum tool to install the Docker software. You can also install the Docker software by referring to the Docker official website.

5.6.2 ERROR6102 Docker is not running

Description

Docker is not running.

Possible Causes

Docker is stopped.

Solution

Run the following command to check the Docker status:

```
systemctl status docker
```

- If Docker is not started, run the following command to start it:

```
systemctl start docker
```
- If the startup fails, view Docker logs as prompted to locate the cause. For example, the Docker configuration file (`/etc/docker/daemon.json`) fails to be parsed.

5.6.3 ERROR6103 Incorrect Docker Cgroup Driver configuration

Description

The **Cgroup Driver** parameter of Docker is incorrectly configured.

Possible Causes

For IEF, the **Cgroup Driver** parameter of Docker must be set to **cgroupfs**.

Solution

Step 1 Run the following command to check the current Docker configuration:

docker info

Step 2 Check whether the **Cgroup Driver** parameter is set to **cgroupfs**.

Cgroup Driver: cgroupfs

Step 3 If not, add the following configuration to the Docker configuration file **/etc/docker/daemon.json**:

```
{  
  "exec-opts": ["native.cgroupdriver=cgroupfs"]  
}
```

For more information, see [Docker Documentation](#).

Step 4 Run the following command to restart Docker:

```
systemctl restart docker
```

Step 5 Run the **docker info** command again to check whether the **Cgroup Driver** parameter is successfully modified. If yes, run the installation command to manage the node.

```
cd /opt/edge-installer; sudo ./installer -op=install
```

----End

5.6.4 ERROR6201 No GPUs available

Description

No GPUs are detected on the node.

Possible Causes

- The GPU type is not supported by IEF. Currently, IEF supports only NVIDIA GPUs.
- The GPUs on the node are not detected.

Solution

Step 1 If a non-NVIDIA GPU is used, do as follows:

Disable GPU when installing the IEF software, or replace the existing GPU with an NVIDIA GPU.

Step 2 If no GPUs are detected, do as follows:

Restart the node.

----End

5.6.5 ERROR6202 The GPU driver is not installed

Description

A GPU driver is not installed on the node.

Possible Causes

A GPU driver is not installed.

Solution

Install the GPU driver by referring to the GPU usage guide.

5.6.6 ERROR6203 The GPU driver is not started

Description

The GPU driver is not started on the node.

Possible Causes

The GPU driver is not started.

Solution

Run the following command to check the GPU status:

systemctl status nvidia-drivers-loader

- If the NVIDIA driver is not started, run the following command to start it:
systemctl start nvidia-drivers-loader
- If the startup fails, view the GPU-related logs as prompted to locate the cause.

5.6.7 ERROR6301 No NPUs available

Description

No NPUs are detected on the node.

Possible Causes

The NPUs on the node are not detected.

Solution

Restart the node. If the check item still fails, replace the node or install the IEF software without enabling NPU.

5.7 Data Collection Errors

5.7.1 ERROR7001 Failed to collect fault data

Description

The collected fault data is incorrect.

Possible Causes

Failed to collect node data.

Solution

Rectify the fault based on the error information. If the issue persists, [submit a service ticket](#).

5.8 Internal Errors

5.8.1 ERROR8001 Unexpected error

Description

edgectl is abnormal.

Possible Causes

- edgectl is faulty.
- The IEF software is faulty.

Solution

- The edgectl software is faulty.
Use edgectl of the latest version.
- The IEF software is faulty.
[Submit a service ticket](#).