

Cloud Service Engine

FAQs

Issue 01
Date 2024-05-16



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

1 Precautions When Using Huawei Cloud CSE.....	1
1.1 Why Can't I See Information About Cloud Services?.....	1
1.2 What Should I Do If an Agency Creation Failed?.....	1
2 Nacos Engines.....	3
2.1 Why Do I Get an Error When I Request gRPC?.....	3
3 ServiceComb Engines.....	4
3.1 How Do I Perform Local Development and Testing?.....	4
3.2 How Can I Handle a Certificate Loading Error?.....	5
3.3 What If the Header Name Is Invalid?.....	6
3.4 What Is the Performance Loss of Mesher?.....	6
3.5 Why Is "Version validate failed" Displayed When I Attempt to Connect to the Service Center?.....	7
3.6 Why Is "Not enough quota" Displayed When I Attempt to Connect to the Service Center?.....	7
3.7 What Should I Do If the Service Registration Fails After IPv6 Is Enabled for the Exclusive ServiceComb Engine with Security Authentication Enabled?.....	8
3.8 What Is Service Name Duplication Check?.....	8
3.9 Why Do I Have to Define Service Contracts?.....	9
3.10 Why Are Microservice Development Framework and Netty Versions Unmatched?.....	9
3.11 What Do I Need to Know Before Upgrading an Exclusive ServiceComb Engine?.....	10
3.12 What Must I Check Before Migrating Services from the Professional to the Exclusive Microservice Engine?.....	12
3.13 Why Is "Duplicate cluster name" Displayed?.....	13
3.14 Error Message "the subnet could not be found" Is Displayed When the Access Address Fails to Be Processed During Engine Creation.....	13
3.15 Why Is Error "does not match rule: {Max: 100, Regexp: ^[a-zA-Z0-9]{1,160}\$ ^-[a-zA-Z0-9][a-zA-Z0-9_-.]{0,158}[a-zA-Z0-9]\$}" Reported?.....	14
3.16 What Should I Do If SpringCloud Applications Fail to Connect to the Configuration Center of ServiceComb Engine 2.x?.....	15
3.17 Why Could My the Global Configuration Not Be Modified?.....	15
3.18 Obtain Configurations Failed.....	16

1 Precautions When Using Huawei Cloud CSE

1.1 Why Can't I See Information About Cloud Services?

Symptom

When viewing the ServiceComb engine and nacos engine, I can't see information about cloud services such as VPC and ELB.

Possible Cause

You do not have the permission to view the enterprise project the cloud services (such as VPC and ELB) belong to. Contact the tenant administrator to assign the permission.

1.2 What Should I Do If an Agency Creation Failed?

Symptom

After I log in to the CSE console, a message is displayed prompting me to enable and authorize the service. After I click **Authorize**, another message is displayed indicating that the authorization failed.

Possible Cause

You have no permission to create an agency, or your agency quota is insufficient.

Solution

- If the system displays a message indicating insufficient permissions when you log in to the IAM console, you have no permission to create an agency. In this case, see [How Can I Obtain Permissions to Create an Agency?](#)

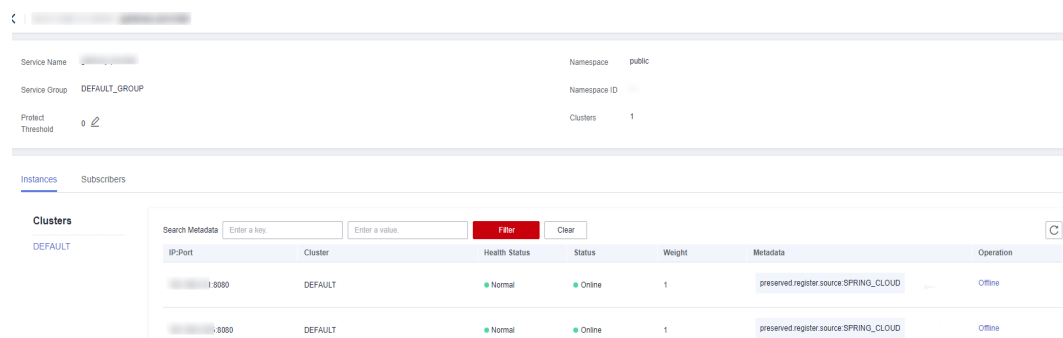
- If you have the permission to create an agency, your agency quota is insufficient. Log in to the IAM console and choose **Agencies** to check the agency quota and either adjust it or delete agencies that are no longer in use.

2 Nacos Engines

2.1 Why Do I Get an Error When I Request gRPC?

Symptom

When the service starts, two instances with ports 8080 and 9090 are registered as HTTP and gRPC ports. However, since only the instance with port 8080 is displayed in the service discovery list, it returns an error when gRPC is requested.



Possible Cause

When a client installed with Go SDK connects to Nacos engines, two instances are registered with a microservice at the same time. This dual registry is currently not available for Nacos engines.

Solution

Do not register two instances with the same microservice at the same time.

3 ServiceComb Engines

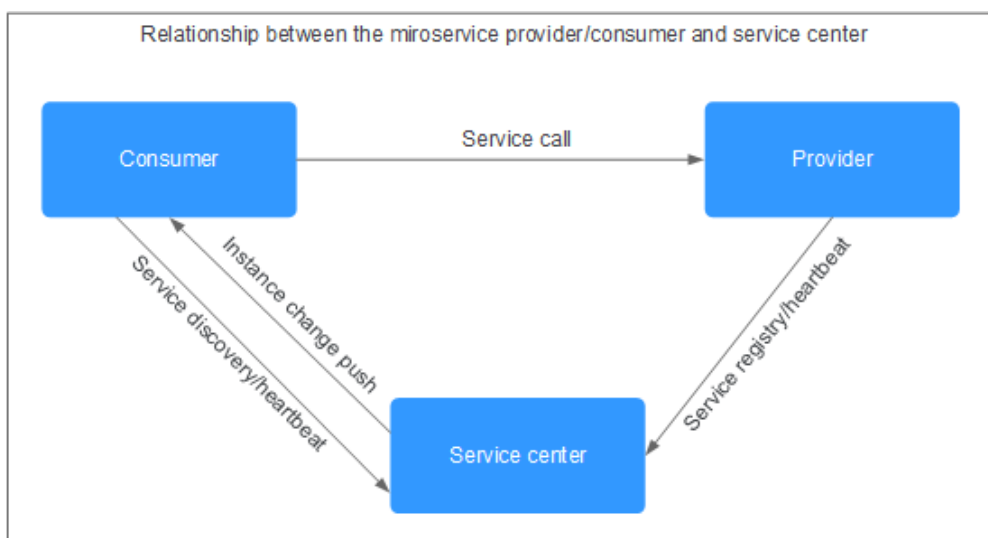
3.1 How Do I Perform Local Development and Testing?

The local development tool provides you with service discovery, registration, and query functions during development, commissioning, and testing.

Concept

This section describes how to locally develop and commission consumer and provider applications. Both service providers and consumers need to connect to a remote service center. This section describes how to start the local service center for local microservice commissioning.

The service center is an important component in the microservice framework. It is used to manage, register, and discover service metadata and service instance metadata. The following figure shows the logical relationship between the service center and microservice provider/consumer.



Prerequisites

- Starting the local service center will occupy ports 30100, 30110, and 30103 of the server, which indicate the backend service port of the service center and the backend and frontend service ports of the configuration center. Ensure that these ports are not in use.
- Before using the local service center, ensure that the environment meets the following requirements:
 - Operating system: Linux/Unix, Windows 64-bit
 - CPU architecture: x86/Arm
 - Browser: Chrome, Safari, Edge

Starting the Local Service Center

Step 1 Download the local development tool package of the required version to the local PC based on the operating system and CPU architecture, and decompress the package to the installation directory. For details, see [Local Development Tool](#).

Step 2 Start CSE.

- On Linux or Unix, go to the root installation directory and run the following command:

```
nohup sh start.sh >/dev/null 2>&1 &
```
- On Windows, go to the root installation directory and double-click **cse.exe**.

Step 3 Stop CSE.

- On Linux or Unix, go to the root installation directory and run the following command:

```
sh stop.sh
```
- On Windows, close the command-line interface.

----End

3.2 How Can I Handle a Certificate Loading Error?

Symptom

```
019/02/21 09:04:16 read ca cert file failed
019/02/21 09:04:16 Init chassis fail: read ca cert file failed
019/02/21 09:04:16 chassis init failed.
019/02/21 09:04:16 SetUp:RoomService init fail: read ca cert file failed
```

If the preceding error is displayed, the `verifyPeer` value is set to **true** but no certificate is configured. By default, the `verifyPeer` value in Go-chassis is **false** indicating that no certificate is configured and is not changed automatically.

Solution

- If a certificate is not necessary but a certificate loading error occurs, the `verifyPeer` value changes during development or continuous integration. In this case, check the code from committing to deployment, and change value **true** to its default value **false**.

- If a certificate is necessary, configure it.

3.3 What If the Header Name Is Invalid?

Symptom

```
{\"level\": \"[0:31mERROR-[0m\", \"timestamp\": \"2018-12-19 10:28:43.145 +08:00\", \"file\": \"handler/transport_handler.go:46\", \"msg\": \"Call  
got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \\\"\\\"]\"}  
{\"level\": \"[0:31mERROR-[0m\", \"timestamp\": \"2018-12-19 10:28:43.145 +08:00\", \"file\": \"handler/transport_handler.go:46\", \"msg\": \"Call  
got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \\\"\\\"]\"}  
{\"level\": \"[0:31mERROR-[0m\", \"timestamp\": \"2018-12-19 10:28:43.146 +08:00\", \"file\": \"handler/transport_handler.go:46\", \"msg\": \"Call  
got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \\\"\\\"]\"}  
{\"level\": \"[0:31mERROR-[0m\", \"timestamp\": \"2018-12-19 10:28:43.146 +08:00\", \"file\": \"handler/transport_handler.go:46\", \"msg\": \"Call  
got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \\\"\\\"]\"}
```

This problem is irrelevant to Go-chassis.

Solution

Check the service code, particularly the customized Go-chassis handler. Specifically, check whether there is an empty header or a non-standard header name.

3.4 What Is the Performance Loss of Mesher?

The service mesh technology uses network traffic hijacking to manage inter-service traffic. In addition to the performance loss in internal logic processing of Mesher, service mesh also causes extra conversion between the user mode and kernel mode (the CPU consumes extra resources). Compared with the latter, the former has little impact on performance. Therefore, the performance loss depends on the size of the payload transmitted over the network. For example, the size of the header and body affects the HTTP transmission speed. Latency of an end-to-end calling of Mesher is 1 ms. For example, if Mesher performance loss is added to the latency in an actual service calling, the total latency is 4 ms longer. However, such a latency is acceptable to the user.

The following table lists the performance test results before and after Mesher is used. The payload is small, that is, the character string **helloworld**. However, certain code is added to increase the computing time of the server to simulate the service code execution time.

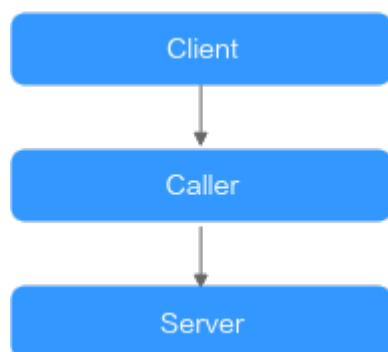


Table 3-1 Test results

Metric	Before	After
TPS	1749	1496
Latency	2.8 ms	3.34 ms
CPU	50%	100%
Concurrency	5	5

The preceding results show that the performance loss of Mesher is low. The performance bottleneck lies in the service code. If the payload content is increased, the performance deteriorates further.

It is recommended that this technology be used to call services during initial technical selection and POC to test the real performance loss.

3.5 Why Is "Version validate failed" Displayed When I Attempt to Connect to the Service Center?

Abnormal Message

```
{"errorCode":"400001","errorMessage":"Invalid parameter(s)","detail":{"Version validate failed, rule: {Length: 64,Length: ^[a-zA-Z0-9_\-]*$}}"}
```

Possible Cause

A new SDK is used, but an earlier SDK of the service center is connected.

Fault Locating

Check the version of the service center. You can download the latest service center from the official website of the public cloud or ServiceComb.

3.6 Why Is "Not enough quota" Displayed When I Attempt to Connect to the Service Center?

Abnormal Message

```
{"errorCode":"400100","errorMessage":"Not enough quota","detail":{"no quota to create instance, ...}}"
```

Possible Cause

There is insufficient quota to add a service instance.

Fault Locating

Log in to the public cloud and view the instance quota on the microservice engine page. If the quota is sufficient, check the service center address and region information configured in the code.

3.7 What Should I Do If the Service Registration Fails After IPv6 Is Enabled for the Exclusive ServiceComb Engine with Security Authentication Enabled?

Symptom

A microservice developed based on Java Chassis is registered with the exclusive ServiceComb engine with security authentication enabled. The registry center address of the microservice is the IPv4 address of the microservice engine registry center. The microservice can be successfully registered and started.

If the registry center address of the microservice is changed to the IPv6 address of the ServiceComb engine registry center, the registration fails and the error "java.net.SocketException: Protocol family unavailable" is reported.

Possible Cause

If you select the VPC network with IPv6 enabled when creating an exclusive ServiceComb engine, the engine supports the IPv6 network. If the service is deployed using a container through an IPv6 network segment, the IPv6 dual-stack function must be enabled for the selected CCE cluster.

If IPv6 is not enabled for the selected cluster, the service network is disconnected, and the error "java.net.SocketException: Protocol family unavailable" is reported.

Solution

Step 1 Modify the environment where the microservice application is deployed by adding a CCE cluster with the IPv6 dual-stack function enabled.

Modify the environment. For details, see [Modifying an Environment](#).

Step 2 Redeploy the application. For details, see [Creating and Deploying Components](#).

----End

3.8 What Is Service Name Duplication Check?

Question

What Is Service Name Duplication Check?

Answer

Microservice names, applications, versions, and environments are checked.

A primary key uniquely identifies a microservice.
Ensure that each primary key is unique.

3.9 Why Do I Have to Define Service Contracts?

The enterprise-level systems are in large scale use and involve many microservice components. Therefore, unified API management is a key requirement of enterprises. CSE uses contract management to meet this requirement.

For management: Through contract management, API definition files that comply with API description standards for microservices are defined. In this way, API development of multiple development teams can be standardized and coordinated. This reduces communication costs and facilitating management.

For development: During microservice development, different teams or even different independent software vendors (ISVs) can develop the same application or system based on the unified API definition file. This facilitates consistency maintenance for the overall system. Specifically, modules in a monolithic application are called using code. Therefore, any API incompatibility can be resolved during the early compilation with low bug fixing costs. When microservices are decoupled, services are remotely called. Therefore, API inconsistency cannot be found during early compilation, resulting in high bug fixing costs. Using service contracts, contract design can be assured, changes can be strictly reviewed, and code can be reversely generated. This ensures API compatibility.

In addition, for small-scale systems that do not have high requirements on unified management, API definition files can be automatically generated through API code.

3.10 Why Are Microservice Development Framework and Netty Versions Unmatched?

Symptom

During development of a microservice application, the following error is displayed:

```
"Caused by: java.lang.NoSuchMethodError:
io.netty.handler.codec.http.websocketx.WebSocketClientHandshakerFactory.newHandshaker(Ljava/net/
URI;Lio/netty/handler/codec/http/websocketx/WebSocketVersion;Ljava/lang/String;ZLio/netty/handler/codec/
http/HttpHeaders;IJJ)Lio/netty/handler/codec/http/websocketx/WebSocketClientHandshaker;"
```

Possible Cause

Third-party software introduced a mismatched version dependency.

Solution

Run the **mvn dependency:tree** command in the development environment to view the dependency tree and check whether the microservice development framework and Netty versions are matched.

For ServiceComb 2.0.1 development framework, the matched Netty version is 4.1.45.Final.

3.11 What Do I Need to Know Before Upgrading an Exclusive ServiceComb Engine?

There are several problems that may occur during and after the upgrade from 1.x to 2.x. The following are their symptoms and solutions.

- Symptom 1:** During the upgrade, the configuration could not be obtained or updated using the API, showing the error "Connection refused" or "Connection was closed". Error examples:


```
[ERROR] Config update from xxx.xxx.xxx.xx failed. Error message is [Connection refused: xxx.xxx.xxx.xx]. org.apache.servicecomb.config.client.ConfigCenterClient$ConfigRefresh.lambda$null $13(ConfigCenterClient.java:428)
```

```
[ERROR]Config update from xxx.xxx.xxx.xx failed. Error message is [Connection was closed]. org.apache.servicecomb.config.client.ConfigCenterClient$ConfigRefresh.lambda$null $13(ConfigCenterClient.java:428)
```

Solution: The configuration center restarts for a short period of time when upgrading. During this restart, the configuration is obtained or updated, causing a disconnection error. To avoid this error, do not update the configuration during the upgrade.
- Symptom 2:** Service scenario governance is unavailable when using an engine 1.x to access the configuration center.

Solution: The configuration center of the engine 2.x has changed to kie, so switch the access mode of the configuration center to kie. For details, see [Using the Configuration Center in Spring Cloud](#).

- Symptom 3:** When you import a configuration file in the original 2.x configuration center format, the import fails and a message is displayed indicating that the file is empty or the format is incorrect.

Solution: Change the format of the configuration items in the configuration file as required by the engine 2.x. The new configuration file is a JSON file with the following format:

```
{
  "data":[
    {
      "key":"xxx",
      "labels":{"environment":"xxx"},
      "service": "xxx",
      "app": "xxx",
      "version": "xxx"
    },
    {"value":"xxx",
      "value_type":"text",
      "status":"enabled"
    }
  ],
  {
    "key":"xxx",
    "labels":{"environment":"xxx"}
  },
  {"value":"xxx",
    "value_type":"text",
```

```

    "status":"enabled"
  },
  {
    "key":"xxx",
    "labels":{"
      "environment":"xxx",
    "service": "xxx"
    },
    "value":"xxx",
    "value_type":"text",
    "status":"enabled"
  },
  {
    "key":"xxx",
    "labels":{"
      "environment":"xxx",
    "service": "xxx",
    "app": "xxx"
    },
    "value":"xxx",
    "value_type":"text",
    "status":"enabled"
  }
]
}

```

where,

- **key** and **value** are mandatory: key and value of the configuration item
 - **labels** is mandatory: configuration range, determined by setting the **environment**, **service**, **app**, and **version** fields
 - **value_type** is mandatory: configuration item type. Value: **ini**, **json**, **text** (default), **yaml**, **properties**, or **xml**
 - **status** is optional: whether to enable the configuration. Value: **enabled** or **disabled** (default)
- **Symptom 4:** If the global configuration is set in the configuration center of ServiceComb engine version 1.x, after the version is upgraded to 2.x, the global configuration automatically adjusts the application scope **environment=\${environmentName}**. The value of *environmentName* can be **empty**, **development**, **testing**, **acceptance**, or **production**. If the SDK uses Kie as the configuration center, you need to add a custom tag to the project configuration file to obtain the configuration. The following uses **environment=production** as an example:

spring-cloud-huawei framework:

```

spring:
  cloud:
    servicecomb:
      config:
        serverType: kie
      kie:
        customLabel: environment
        customLabelValue: production

```

servicecomb-java-chassis framework:

```

servicecomb:
  kie:
    customLabel: environment
    customLabelValue: production

```

3.12 What Must I Check Before Migrating Services from the Professional to the Exclusive Microservice Engine?

Background

The professional microservice engine is a logical multi-tenant engine. Since all tenants share one engine, any engine fault will interrupt all registered services. To prevent such fault and ensure service continuity, switch services from the professional microservice engine to the commercial exclusive microservice engine.

Switching has the following advantages:

1. Physical isolation. An exclusive microservice engine is physically isolated, since one tenant uses one engine exclusively. If the engine is faulty, other engines are not affected.
2. Multiple AZs. An exclusive microservice engine can be deployed in multiple AZs to improve reliability.
3. Large capacity. You can create many exclusive microservice engines and each one supports 2000 instances, compared to 500 instances per tenant of a professional microservice engine.

After you switch from the professional edition (the engine name is **Cloud Service Engine**) to the exclusive edition (the engine name can be customized), the engine functions remain the same and user configurations and service data have been migrated to the new engine.

Precautions

1. Check whether you are using a professional microservice engine. To do this, confirm that the instance is registered with the engine named **Cloud Service Engine**.
2. Create an exclusive microservice engine before the migration. Select one with 100, 500, or 2000 instances.
3. The new engine's VPC must be the VPC where the component to be upgraded is deployed.
4. Migration switches over registry center and configuration center addresses. Switchover causes all services registered with the old engine to be registered with the new one. During this process, there may be mixed registries, preventing services from being discovered or invoked and affecting availability.
5. Confirm the deployment mode. If it is ServiceStage application hosting, contact O&M personnel to obtain a quick migration solution. Otherwise, change the registry and configuration center addresses to the address of the new exclusive engine. In this case, you are advised to get a risk evaluation from O&M personnel and confirm a reliable solution before migrating with diversified deployment modes.

6. The migration includes both instance and configuration migration, so back up data in the configuration center first. Contact O&M personnel for any assistance. Migration includes both dynamic and global configurations. Global configuration: On the professional engine console, switch to the environments one by one and check whether a global configuration exists. If yes, export the global configuration for backup. Dynamic configuration: Check whether a dynamic configuration exists. If yes, export and save it. Ignore the microservice and microservice scope without dynamic configuration.
7. Upgrade each microservice before the migration to avoid failure by external factors.
8. Check whether the **JAVA_ARGS** parameter exists. If yes, check whether the following information exists:

```
spring.cloud.servicecomb.discovery.address  
spring.cloud.servicecomb.credentials.enabled  
spring.cloud.servicecomb.credentials.accessKey  
spring.cloud.servicecomb.credentials.secretKey  
spring.cloud.servicecomb.credentials.akskCustomCipher  
spring.cloud.servicecomb.credentials.project
```

If yes, they are no longer needed after the migration and can be deleted.
9. If importing a configuration file in the original 2.x configuration center format failed, a message indicates that the file is empty or the format is incorrect. In this case, rectify the fault by referring to [What Do I Need to Know Before Upgrading an Exclusive ServiceComb Engine?](#).

3.13 Why Is "Duplicate cluster name" Displayed?

Abnormal Message

You receive the following message when creating an engine:

```
{"error_code":"SVCSTG.00500500","error_message":{"kind":"Status":"Failure"... "message":"duplicate cluster name","reason":"Conflict"}}
```

Possible Cause

The CCE service on which the microservice engine depends is abnormal, causing engine creation, deletion, and configuration to be unavailable.

Solution

Contact O&M personnel.

3.14 Error Message "the subnet could not be found" Is Displayed When the Access Address Fails to Be Processed During Engine Creation

Symptom

During engine creation, the access address fails to be processed, and the following error message is displayed:


```
{"error_code":"SVCSTG.00500404","error_message":{"code":"VPC.0202","message":"Query resource by id xxx fail.the subnet could not be found."}}
```

Possible Cause

CSE is not authorized in the user's project.

Solution

- When you use CSE instances provisioned from ServiceStage and want to provision new instances in CSE, you need to grant permissions to CSE. For details, see [Creating a User and Granting Permissions](#).
- CSE depends on VPC. If you do not grant any permission, create a cloud service agency `cse_admin_trust` by referring to [Creating an Agency \(by a Delegating Party\)](#).

3.15 Why Is Error "does not match rule: {Max: 100, Regexp: ^[a-zA-Z0-9]{1,160}\$|^ [a-zA-Z0-9][a-zA-Z0-9_ \-]{0,158}[a-zA-Z0-9]\$" Reported?

Symptom

When registering my local lightweight service, I get the error "does not match rule: {Max: 100, Regexp: ^[a-zA-Z0-9]{1,160}\$|^ [a-zA-Z0-9][a-zA-Z0-9_ \-]{0,158}[a-zA-Z0-9]\$"."

Error logs of local service registry:

```
register failed, will retry, please check config file. message=read response failed. status:400; message:Bad Request; content:{"errorCode":"400001","errorMessage":"Invalid parameter(s)","detail":{"field 'MicroService.Schemas' invalid value '[MogoManagerController ResExpandFollowController PurPurchaseController DevSupplierClassController BilTransferConfirmController RptRentAccountController ApprovalProjectParaController RptOperateInfoController ResThemeController ExportController SysBannerController SysBillingMethodController ReportSourceController DevMaintainController ReportInvoiceController SysPushMsgController ResHouseTypeController TestController UnionPayController WebMvcEndpointHandlerMapping_WebMvcLinksHandler ResSourceConfigureController BizCheckOutUserController BillInterfaceMsgController BizSettleController SysInterfaceMsgController RptBigDataController ResRepairController ContCheckInHouseController SysBranchCodeController ResProjectHotController ContRentLadderController ContContractSourceRelController ReportDepositController RptWeeklyController GlobalExceptionHandler PurPayDetailController JobController ContShortContractController ResCostConfigureDetailControl
```

```
does not match rule: {Max: 100, Regexp: ^[a-zA-Z0-9]{1,160}$|^ [a-zA-Z0-9][a-zA-Z0-9_ \-]{0,158}[a-zA-Z0-9]$"}
```

Possible Cause

There are too many default local lightweight schemas. Max: 100

Solution

Modify `quota_plugin="unlinit"` in the `/conf/app.conf` file extracted from the local lightweight tool package.

NOTE

Use this method when exceeding other quotas too.

3.16 What Should I Do If SpringCloud Applications Fail to Connect to the Configuration Center of ServiceComb Engine 2.x?

Symptom

The center is configured in the code, but the configuration items cannot be obtained.

```
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1247)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1236)
at com.huawei.financialsolutionassetcenter.console.ConsoleApplication.main(ConsoleApplication.java:29)
Caused by: java.lang.IllegalArgumentException: Create breakpoint : Could not resolve placeholder 'cse.example.key1' in value "${cse.example.key1}"
at org.springframework.util.PropertyPlaceholderHelper.parseStringValue(PropertyPlaceholderHelper.java:186)
```

Possible Cause

The type and address of the configuration center in the configuration file are incorrect.

Fault Location

1. The engine uses the 2.x key, but the configuration center uses the 1.x key.
2. The environment is specified in the configuration file, but the environment key obtained from the code is incorrect.

Solution

1. Set **spring.cloud.servicecomb.config.serverType** to **kie** in the configuration file to change the configuration center type to kie.
2. Modify **server.env** to change the environment key in the configuration file.

3.17 Why Could My the Global Configuration Not Be Modified?

Symptom

After the global configuration is modified, the configuration obtained by the service is not changed, and the keyword **changed** is not recorded in the log.

```
2021-11-15 10:13:33.710 INFO 6212 --- [ntloop-thread-0] .a.s.ConfigCenterConfigurationSourceImpl : Config value cache changed; action:set, item:{servicecomb.credentials.akskEn
2021-11-15 10:13:33.710 INFO 6212 --- [ntloop-thread-0] o.a.s.config.client.ParseConfigUtils : Updating remote config is done, revision has changed from default9602643 to 4
2021-11-15 10:13:36.438 INFO 6212 --- [etator-poller-0] o.a.s.a.c.publish.DefaultLogPublisher :
vertx:
instances:
  name      eventLoopContext-created
  registry  0
  transport 0
  config-center 0
```

Possible Cause

The dependency of config-cc was not configured.

Fault Location

1. Check for the address of the configuration center in the **application.yaml** file.
2. Check for the config-cc dependency in the **pom.xml** file.

Solution

Add the config-cc dependency to the **pom.xml** file.

```
<dependency>  
<groupId>org.apache.servicecomb</groupId>  
<artifactId>config-cc</artifactId>  
</dependency>
```

3.18 Obtain Configurations Failed

Symptom

After a microservice is connected to the corresponding microservice development framework (such as spring-cloud-huawei and java-chassis), it fails to obtain configuration items from the ServiceComb engine by calling the configuration query API through SDK.

Possible Cause

If the connection between a microservice and the registration center jitters due to network and CPU problems, the request may be abnormal.

Solution

The microservice framework has the self-healing capability. If the configuration fails to be obtained, the retry mechanism takes effect for pulling configurations. Service exceptions do not occur. Check whether configuration items can be obtained next time. If not, contact customer service.