

阿里云 Serverless 应用引擎

故障处理

文档版本：20200415

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务时间约十分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 注意： 权重设置为0，该服务器不会再接受新请求。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	单击 设置 > 网络 > 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
Courier字体	命令。	执行cd /d C:/window命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid Instance_ID
[]或者[a b]	表示可选项，至多选择一个。	ipconfig [-all -t]
{ }或者{a b}	表示必选项，至多选择一个。	switch {active stand}

目录

法律声明.....	I
通用约定.....	I
1 微服务配置.....	1
1.1 SAE上的应用与其数据库链接失败，如何处理？	1
1.2 应用删除失败，如何处理？	1
1.3 在配置管理中SAE的自定义命名空间怎么配置？	1
1.4 在SAE配置中心创建properties格式的配置后，使用Nacos SDK可以获取但是yaml访问不了？	2
1.5 SAE的服务列表中查询不到服务信息，如何处理？	2
1.6 API调用次过多导致限流，怎么处理？	3
1.7 应用后台接收到请求后产生中文乱码，如何处理？	4
2 应用实例.....	5
2.1 Pod问题.....	5
3 网络.....	7
3.1 SAE中的应用为什么公网请求总超时？	7
4 SLB.....	8
4.1 SAE应用添加不了SLB怎么办？	8
5 日志、监控和诊断.....	9
5.1 查看文件日志时，在SLS控制台无数据展示，如何处理？	9
6 Webshell.....	10
6.1 Webshell登录应用失败如何处理？	10
7 应用开发.....	11
7.1 HSF 问题.....	11
7.1.1 定位及解决 HSF 问题.....	11
7.1.2 HSF 如何设置超时时间？	11
7.1.3 HSF 其它.....	12
7.1.4 com.ali.unit.rule.Router 初始化失败.....	13
7.1.5 如何对 HSF 服务进行单元测试？	13
7.1.6 HSF invalid call is removed.....	13
7.1.7 单个 HSF 应用发布服务数超过 800 个限制的报错.....	14
7.1.8 使用 Ali-Tomcat 启动多个 HSF 失败.....	15
7.1.9 HSF 启动时绑定 IP、端口失败.....	16
7.1.10 HSF 应用注册或订阅服务数多导致启动比较慢.....	16
7.1.11 服务认证失败.....	19
7.2 HSF 错误码.....	19
7.2.1 错误编码：HSF-0001.....	19
7.2.2 错误编码：HSF-0002.....	23
7.2.3 错误编码：HSF-0003.....	24

7.2.4 错误编码: HSF-0005.....	24
7.2.5 错误编码: HSF-0006.....	24
7.2.6 错误编码: HSF-0007.....	24
7.2.7 错误编码: HSF-0008.....	25
7.2.8 错误编码: HSF-0009.....	25
7.2.9 错误编码: HSF-0012.....	25
7.2.10 错误编码: HSF-0013.....	25
7.2.11 错误编码: HSF-0014.....	26
7.2.12 错误编码: HSF-0016.....	26
7.2.13 错误编码: HSF-0017.....	26
7.2.14 错误编码: HSF-0018.....	26
7.2.15 错误编码: HSF-0019.....	27
7.2.16 错误编码: HSF-0020.....	27
7.2.17 错误编码: HSF-0021.....	27
7.2.18 错误编码: HSF-0022.....	28
7.2.19 错误编码: HSF-0023.....	28
7.2.20 错误编码: HSF-0024.....	28
7.2.21 错误编码: HSF-0027.....	32
7.2.22 错误编码: HSF-0030.....	32
7.2.23 错误编码: HSF-0029.....	32
7.2.24 错误编码: HSF-0031.....	34
7.2.25 错误编码: HSF-0032.....	34
7.2.26 错误编码: HSF-0033.....	34
7.2.27 错误编码: HSF-0034.....	34
7.2.28 错误编码: HSF-0035.....	35
7.2.29 错误编码: HSF-0038.....	35
7.2.30 错误编码: HSF-0051.....	35
7.2.31 错误编码: HSF-0077.....	35
7.2.32 错误编码: HSF-0089.....	35
7.3 Ali-Tomcat 问题.....	36
7.3.1 Ali-Tomcat 启动失败.....	36
7.3.2 com.taobao.diamond.client.impl.DiamondEnvRepo 初始化失败.....	36
7.3.3 未知实例异常.....	36
7.3.4 为什么我在代码中配置的 Tomcat 端口不生效?	37
7.4 Pandora 问题.....	38
7.4.1 如何设置用户日志不被覆盖.....	38
7.4.2 Pandora 启动失败, QoS 端口绑定异常.....	39
7.4.3 Pandora 版本问题.....	39
7.5 轻量级配置及注册中心问题.....	41
7.5.1 更新配置后, 相关应用中的配置未生效.....	41
7.5.2 服务调用不通.....	43
7.5.3 服务注册慢.....	44

1 微服务配置

1.1 SAE上的应用与其数据库链接失败，如何处理？

问题现象

部署在SAE上的应用于数链接失败。

解决方案

检查部署在SAE上的应用与数据库是否处于同一VPC内。

- 是，具体操作请参见[#unique_5/unique_5_Connect_42_section_03m_fax_xys](#)。
- 否，具体操作请参见[#unique_5/unique_5_Connect_42_section_oig_j0g_gkw](#)。

1.2 应用删除失败，如何处理？

问题现象

在SAE控制台删除应用失败。

可能原因

由于您的账号设置了操作保护，所以应用删除失败。

解决方案

1. 在SAE控制台单击右上角帐户头像。
2. 在安全设置页面单击操作保护所在行设置，并采用手机验证方式完成应用删除。

1.3 在配置管理中SAE的自定义命名空间怎么配置？

问题现象

在配置管理中命名空间IDspring.cloud.nacos.config.namespace失败。

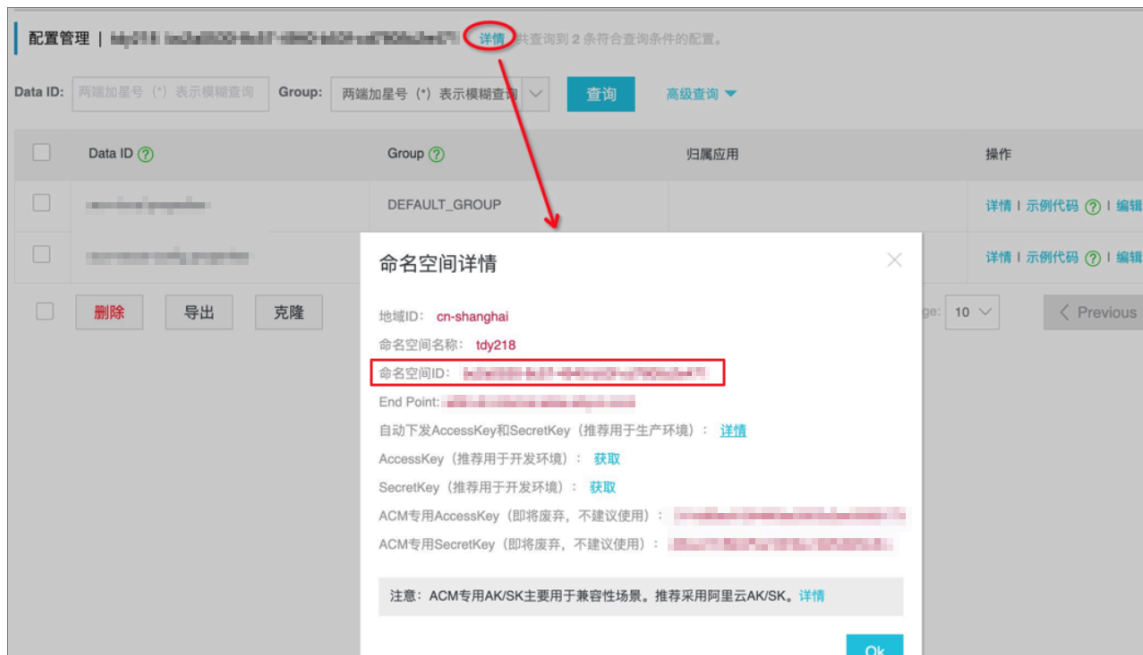
可能原因

spring.cloud.nacos.config.namespace 参数值是命名空间 ID，不是命名空间名称。

解决方案

命名空间ID的获取途径有以下两种。

- 在配置管理页面获取。



更多详情，请参见[#unique_8](#)。

- 通过配置管理读取API，详情参见[#unique_9](#)。

1.4 在SAE配置中心创建properties格式的配置后，使用Nacos SDK可以获取但是yaml访问不了？

问题现象

在SAE配置中心创建properties格式的配置后，使用Nacos SDK可以获取，但使用yaml访问不了。

可能原因

Nacos SDK支持解析properties格式配置，对yaml格式暂时不支持。

解决方案

遇到该问题，需要您自己处理返回值，Nacos SDK无法对返回值格式化。

1.5 SAE的服务列表中查询不到服务信息，如何处理？

问题现象

SAE的服务列表中查询不到服务信息。

可能原因

- 可能原因1：没有使用SAE注册中心。
- 可能原因2：SDK版本过低。
- 可能原因3：应用程序中有ANS依赖。

解决方案

- 可能原因1的解决方案：将您的服务注册中心改为SAE注册中心。
 - Spring Cloud：具体请参见[#unique_12](#)。
 - Dubbo：具体请参见[#unique_13](#)或者[#unique_14](#)。
- 可能原因2的解决方案：将SDK进行升级，使用最新版本。
- 可能原因3的解决方案：在应用程序代码中移除ANS，切换为[nacos-discovery](#)。

1.6 API调用次过多导致限流，怎么处理？

问题现象

提示 “Failed to get change order info: Throttling.User: Request was denied due to user flow control .RequestID:XXXXXXXX” 。

可能原因

当前的限流阈值为 “单用户一分钟60次Api调用” ， 同时部署多个时容易被限流。

解决方案

请提供工单或者在钉钉群联系技术支持。



1.7 应用后台接收到请求后产生中文乱码，如何处理？

问题现象

使用URL方式前端向后端发送请求后，后端接收到中文乱码。

可能原因

应用较老且其使用的Tomcat版本较旧低于Tomcat 8版本。

解决方案

使用Tomcat 8及其以上版本。

2 应用实例

2.1 Pod问题

部署在SAE上的应用运行时，出现的问题一般为Pod问题。本文介绍常见的Pod问题以及处理方法。

Pod常见问题列表

- [ImagePullBackOff](#)
- [CrashLoopBackOff](#)
- [RunContainerError](#)
- [处于Pending状态的Pod](#)
- [处于未就绪状态的Pod](#)

ImagePullBackOff

当Kubernetes无法获取到Pod中某个容器的镜像时，将出现此错误。

可能原因：

- 镜像名称无效，例如拼错镜像名称，或者镜像不存在。
- 您为image指定了不存在的标签。
- 您检索的镜像是私有registry，而Kubernetes没有凭据可以访问它。

解决方法：

- 前两种情况可以通过修改镜像名称和标记来解决该问题。
- 第三种情况，您需要将私有registry的访问凭证，通过Secret添加到Kubernetes中并在Pod中引用它。

CrashLoopBackOff

如果容器无法启动，则Kubernetes将显示错误状态为：CrashLoopBackOff。

可能原因：

- 应用程序中存在错误，导致无法启动。
- 未正确配置容器。
- Liveness探针失败太多次。

解决方法：

您可以尝试从该容器中检索日志以调查其失败的原因。如果容器重新启动太快而看不到日志，则可以使用命令：`$ kubectl logs <pod-name> --previous`。

RunContainerError

当容器无法启动时，出现此错误。

可能原因：

- 挂载不存在的卷，例如ConfigMap或Secrets。
- 将只读卷安装为可读写。

解决方法：

请使用`kubectl describe pod`命令收集和分析错误。

处于Pending状态的Pod

当创建应用是时，在变更记录中该Pod一直Pending状态。

可能原因：

- 集群没有足够的资源（例如CPU和内存）来运行Pod。
- 当前的命名空间具有ResourceQuota对象，创建Pod将使命名空间超过配额。
- 该Pod绑定到一个处于pending状态的 PersistentVolumeClaim。

解决方法：

- 检查`$ kubectl describe pod <pod name>`命令输出的“事件”部分内容或者在SAE控制台查看应用事件。
- 对于因ResourceQuotas而导致的错误，可以使用`$ kubectl get events --sort-by=.metadata.creationTimestamp`命令检查集群的日志。

处于未就绪状态的Pod

如果Pod正在运行但未就绪（not ready），则表示readiness就绪探针失败。

可能原因：

当“就绪”探针失败时，Pod未连接到服务，并且没有流量转发到该实例。

解决方法：

就绪探针失败是应用程序的特定错误，请检查`$ kubectl describe pod <pod name>`命令输出的“事件”部分内容或者在SAE控制台查看对应的应用事件。

3 网络

3.1 SAE中的应用为什么公网请求总超时？

问题现象

部署在SAE应用访问公网超时。

可能原因

SAE中的应用默认是不能出公网。如果有公网访问需求，需要进行网关配置。

解决方案

SAE上的应用如果需要访问公网，请参见[#unique_21](#)。

4 SLB

4.1 SAE应用添加不了SLB怎么办？

问题现象

当前账号已经有60个SLB，在为SAE上的应用添加SLB添加不了。

可能原因

SLB配置达到配额阈值。

解决方案

- 方法一：在SLB控制台申请提升SLB配额，详情参见[管理配额](#)。
- 方法二：删除不使用的SLB。

5 日志、监控和诊断

5.1 查看文件日志时，在SLS控制台无数据展示，如何处理？

问题现象

查看文件日志时，在SLS控制台无数据展示。

可能原因

通常SLS默认查询最近15分钟内的查询结果。

解决方案

如果SLS无数据显示，建议使用Webshell查看您的应用是否有文件日志信息。

- 如果有，请联提工单。
- 如果没有，请对您的应用进行检查。

6 Webshell

6.1 Webshell登录应用失败如何处理？

问题现象

使用Webshell登录应用失败。

可能原因

Webshell运行依赖/bin/bash，您的镜像中没有该执行文件。

解决方案

- 方法一：在Dockerfile文件中指定带有/bin/bash的基础镜像。
- 方法二：在Dockerfile中建立软连接/bin/bash -> /bin/sh。

7 应用开发

7.1 HSF 问题

7.1.1 定位及解决 HSF 问题

HSF 的问题描述会记录在 `/home/admin/logs/hsf/hsf.log` 中，如果出现与 HSF 相关的问题，请查询此文件进行错误定位。

HSF 的错误一般都会有对应的错误码，根据错误码及文档，也可找到对应的解决方案。HSF 错误码列表如下：

- [HSF-0001](#)
- [HSF-0002](#)
- [HSF-0003](#)
- [HSF-0005](#)
- [HSF-0007](#)
- [HSF-0008](#)
- [HSF-0009](#)
- [HSF-0014](#)
- [HSF-0016](#)
- [HSF-0017](#)
- [HSF-0020](#)
- [HSF-0021](#)
- [HSF-0027](#)
- [HSF-0030](#)
- [HSF-0031](#)
- [HSF-0032](#)
- [HSF-0033](#)
- [HSF-0035](#)

7.1.2 HSF 如何设置超时时间？

解决方案

可通过 HSF 标签：**methodSpecials** 和 **clientTimeout** 进行配置。

- `methodSpecials`: 为方法单独配置超时时间 (单位 ms)
- `clientTimeout`: 为接口中所有方法设置统一的超时时间 (单位 ms)

超时时间配置的优先级由高到低。

消费者 (客户端) `methodSpecials` > 消费者 (客户端) `clientTimeout` > 提供者 (服务端) `methodSpecials` > 提供者 (服务端) `clientTimeout`

Consumer 标签配置示例:

```
<hsf:consumer id="service" interface="com.taobao.edas.service.SimpleService"
version="1.1.0" group="test1" clientTimeout="3000"
target="10.1.6.57:12200?_TIMEOUT=1000" maxWaitTimeForCsAddress="5000">
<hsf:methodSpecials>
  <hsf:methodSpecial name="sum" timeout="2000" ></hsf:methodSpecial>
</hsf:methodSpecials>
</hsf:consumer>
```

7.1.3 HSF 其它

问题一

报错信息:

启动时, `java.lang.IllegalArgumentException: HSFApiConsumerBean.ServiceMetadata.ifClazz is null.`

解决方案:

接口对应的类没有加载成功, 请确认接口类在 `class path` 中可以被加载到。

问题二

报错信息:

`failure to connect 10.10.1.1.`

解决方案:

1. 确认是否在同一个 VPC、同一个地区。

如果不是那不能连通。

2. 确认是否在同一个安全组。

如果不在同一个安全组则需要开通对应的 12200 端口。

3. 执行 `telnet 10.10.1.1 12200` 查看端口是否可以连通。

如果不能连通, 请确认 10.10.1.1 机器的防火墙设置。

7.1.4 com.ali.unit.rule.Router 初始化失败

错误信息

SEVERE: Context initialization failed java.lang.NoClassDefFoundError: Could not initialize class com.ali.unit.rule.Router

解决方案

不能链接地址服务器 jmenv.tbsite.net，请进行域名绑定。

在 hosts 文件中添加以下内容进行地址服务器域名绑定：**192.168.1.10 jmenv.tbsite.net**（其中 192.168.1.10 改为您所使用的轻量配置中心的地址），hosts 文件的路径如下：

- Windows: **C:\Windows\System32\drivers\etc\hosts**
- Linux: **/etc/hosts**

7.1.5 如何对 HSF 服务进行单元测试？

在开发环境可以使用 LightApi 进行单元测试请参见 [HSF 单元测试](#)。

7.1.6 HSF invalid call is removed

问题现象

```
invalid call is removed because of connection closed
```

可能原因

- 网络闪断：客户端和服务端建立的连接后，客户端发起调用请求，若服务端还在处理该请求，此时同时没有达到客户端超时，因网络等各类问题导致了客户端主动关闭了连接，此时会报错。
- 服务端重启：客户端发起请求后，等待服务端响应，但此时服务端重启，导致 socket 断开，客户端收到操作系统 connection closed 回调，将报错。
- 服务器端 OOM：服务器端出现 outofmemory 以前，一直在 FullGC。

解决方案

如果业务是幂等性的，可以由业务做重试机制处理。同时检查服务器网络，一般情况是网络断开（闪断）导致。

7.1.7 单个 HSF 应用发布服务数超过 800 个限制的报错

HSF 应用发布时，遇到下面的报错：java.lang.IllegalArgumentException: publisher count bigger than 800

问题现象

创建 Kubernetes 应用时，拉取镜像失败。

可能原因

这是因为 Pandora 里面的 configclient 插件对单个应用发布的 HSF 服务数有如下限制：

名称	示例	限制	是否可调整
{服务名}:{版本号}	com.alibaba.edas.testcase.api.TestCase:1.0.0	最大 192 字节	否
组名	aliware	最大 32 字节	否
一个 Pandora 应用实例发布的服务数	N/A	最大 800 个	可以，参见 解决方案 。

解决方案

1. 在应用**基本信息**页面的**应用设置**区域选择**设置 > JVM**。

2. 在**应用设置**对话框中单击**自定义**，在**自定义参数**右侧的配置内容文本框中输入 JVM 属性参数-DCC.pubCountMax=2000，然后单击**配置 JVM 参数**。

3. 重启应用使该参数生效。

7.1.8 使用 Ali-Tomcat 启动多个 HSF 失败

使用 AliTomcat 在同一台机器上同时启动多个 HSF 应用时，可能会出现无法删除临时目录的错误，或者类加载相关的问题。

可能原因

两个 HSF 进程同时启动的时候，会向相同的临时目录解压，相互造成干扰。

解决方案

- 通过 **-Dcom.taobao.pandora.tmp_path=** 指定不同的临时解压目录
- 通过 **-Dhsf.server.port=** 指定 HSF 不同的服务监听端口
- 通过 **-Dpandora.qos.port=** 指定 Pandora 不同的 QoS 端口
- 通过 **-Dproject.name=** 指定 HSF 的应用名

除了在启动参数里配置上述启动参数外，还可以在 tomcat 目录下的 conf/server.xml 中修改监听端口，样例如下：

```
<Server port="8006" shutdown="TAOBAO-TOMCAT-SHUTDOWN">  
<Connector port="8080" protocol="HTTP/1.1"
```

```
connectionTimeout="15000" redirectPort="8443" maxParameterCount="1000"
maxThreads="250" maxPostSize="2097152" acceptCount="200" useBodyEnc
odingForURI="true" />
```

7.1.9 HSF 启动时绑定 IP、端口失败

使用 AliTomcat 在同一台机器上同时启动多个 HSF 应用时，可能会出现无法删除临时目录的错误，或者类加载相关的问题。

问题现象

HSF 启动时报错，`java.net.BindException: Can't assign requested address`。

可能原因

获取当前 IP 端口失败。

解决方案

通过设置 JVM 参数解决。

除了在启动参数里配置上述启动参数外，还可以在 tomcat 目录下的 `conf/server.xml` 中修改监听端口，样例如下：

```
-Dhsf.server.ip=你本地网卡的 IP -Dhsf.server.port=12200
```

7.1.10 HSF 应用注册或订阅服务数多导致启动比较慢

问题现象

使用 Pandora Boot 开发的 HSF 应用，当应用注册或订阅的服务数多时，启动比较慢。

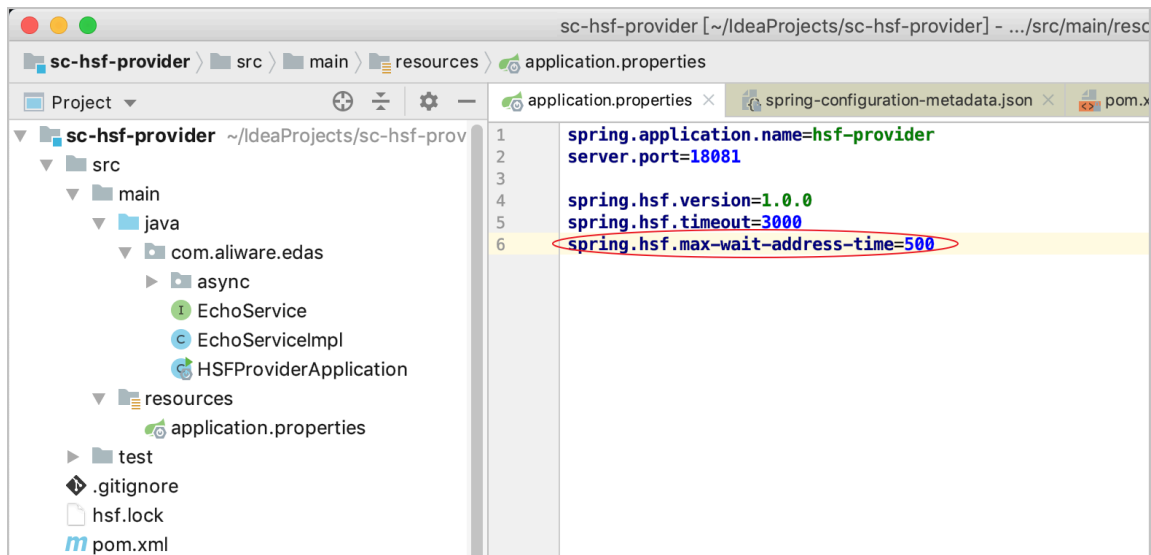
可能原因

使用 Pandora Boot 开发的 HSF 应用会为每个发布（服务提供者端）或消费（服务消费者端）的服务设置从服务注册中心（ConfigServer）同步地址的默认时间，默认为 3000 毫秒。当发布或消费的服务数量多的时候，每个服务需要串行等待同步地址，会导致应用启动很慢。

解决方案

- 打开 Pandora Boot 应用的 application.properties 文件，修改该全局默认同步时间参数 spring.hsf.max-wait-address-time 参数（单位：毫秒），例如：

```
spring.hsf.max-wait-address-time = 500
```

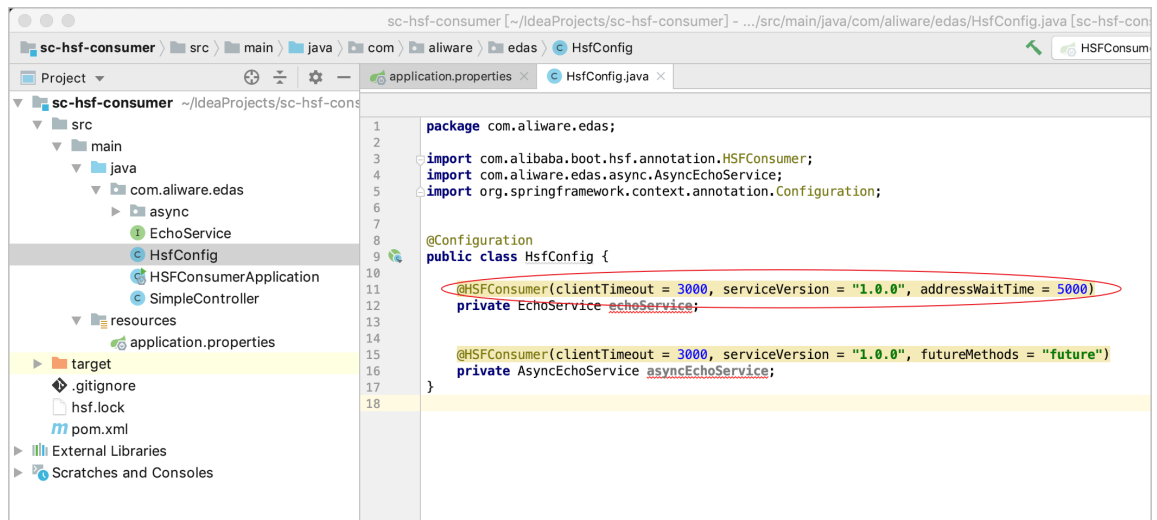


- 服务消费者端还可以针对服务接口通过参数 addressWaitTime 单独设置地址同步时间。

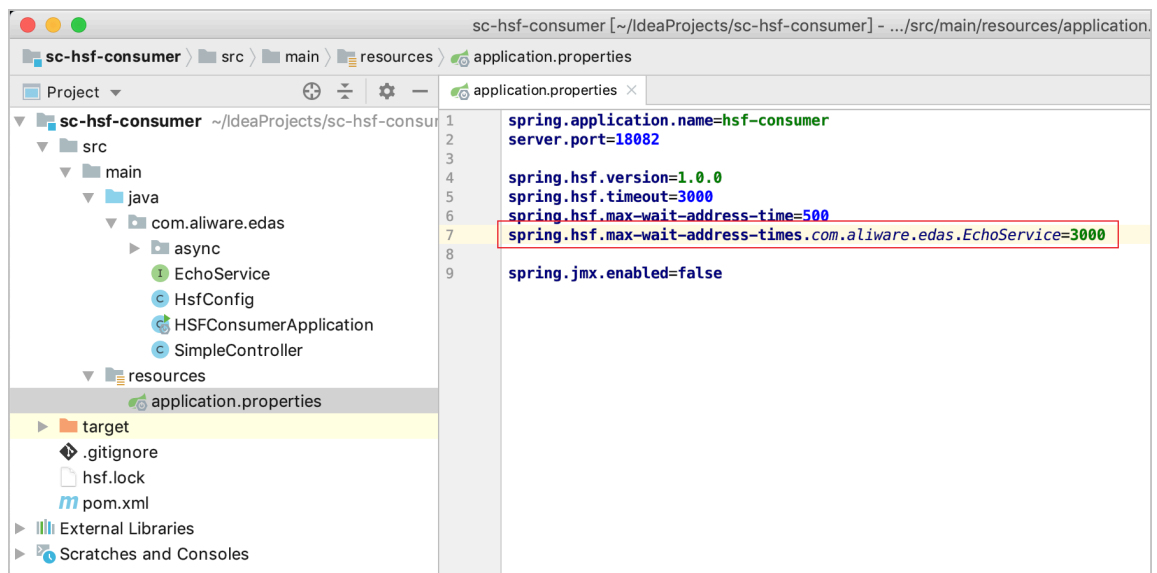
当服务进行订阅时，会在该参数指定的时间内阻塞线程，等待地址推送，避免调用该服务时因为地址为空而出现地址找不到的情况。在消费者端应用启动时就需要订阅服务的接口才需要设

置该参数，推荐设置为 3000 ~ 5000 毫秒，否则应用启动时可能遇到 [HSF-0001-HSFServiceAddressNotFoundException](#) 的异常（设置该参数的一个副作用就是应用启动会延长）。

在应用程序中对某个服务消费者接口单独设置等待地址推送时间addressWaitTime，单位：毫秒。



- 或者在应用配置文件application.properties中单独对某个服务接口设置spring.hsf.max-wait-address-times.<完整的服务接口名>=<等待地址推送时间，等待地址推送时间的单位为毫秒。



- 如果设置了 spring.hsf.max-wait-address-time 参数并重启了应用，应用启动还是有点慢，可以对正在启动的应用做线程堆栈（jstack <应用JVM进程号> >> threaddump.txt）并结合\$USER_HOME/log/configclient/config-client.log、\$USER_HOME/log/hsf/下面的日志进行分析。

7.1.11 服务认证失败

:

错误信息

java.lang.Exception: Service authentication failed （此问题只会发生在 EDAS 生产环境）

解决方案

执行 **date** 命令，查看是否时间不准确。

7.2 HSF 错误码

7.2.1 错误编码：HSF-0001

报错信息

HSFServiceAddressNotFoundException 未找到需要调用的服务目标地址。

描述信息

需要调用的目标服务为：xxxx； 组别为：xxxx。

基本问题排查

1. 服务本身的发布、调用是否正确。
 - 验证服务是否发布：在对应环境（日常、预发或线上环境）的“服务治理”控制台查询需要的服务。
 - 名称不匹配，检查代码中对应发布者和消费者 XML 配置的 interface、version、group，3 个数据要完全一致（注意大小写也要一样，同时要注意前后不能有空格）。
2. 服务提供者防火墙开启或网络不通，telnet 服务提供者 IP HSF 端口（默认12200）是否可以正常连接，如果不能正常连接说明防火墙开启或网络存在问题，需要协调相关负责人员排查。
3. 是否存在多网卡，如果存在多网卡可以使用 -Dhsf.server.ip 来指定服务端的 IP。
 - 本地开发环境可以直接 jvm 启动参数设定。
 - 生产环境需要跟开发联系确定如何解决。
4. 服务调用太快，在 **ConfigServer** 将地址推送过来之前就发起调用导致出错。在服务消费者的配置项里添加 **maxWaitTimeForCsAddress** 的相关配置，请参考[消费者订阅服务](#)。

本地开发环境排查

本地使用轻量配置中心进行开发时，服务发布、订阅均无需鉴权，故服务正常启动就可以注册、订阅成功。在排除掉前面的基本问题后，可以按照如下步骤进行排查。

1. 轻量配置中心检查。

- a) 登录轻量配置中心控制台，搜索对应的服务是否已经发布成功，服务提供者的 IP、端口是否正确。如果服务未发布成功，按照下方《服务提供者排查》步骤进行排查。
- b) 搜索对应的服务是否已经订阅成功。如果服务未订阅成功《服务消费者排查》步骤进行排查。
- c) 从订阅者机器上 Telnet 服务提供者 IP 端口，是否可以正常联通。

2. 服务提供者排查。

如果轻量配置中心控制台发现服务未能发布成功，请进行如下排查。

- a) `ping jmentv.tbsite.net` 确定地址服务器 IP 跟轻量配置中心 IP 一致。
- b) 清空 `{userhome}/logs/` 下所有日志，清空 `{userhome}/configclient/` 目录。
- c) 启动服务提供者应用，如果之前 Tomcat 已经启动，请重启。
- d) 首先查看 Tomcat 启动日志，无异常信息，最后显示启动耗时 ms。如果有异常，请解决异常。
- e) 查看 `{userhome}/configclient/logs/configclient.log` 或 `{userhome}/logs/configclient/configclient.log`（不同版本稍有区别），查找 `Connecting to remoting://{IP 地址}` 连接的注册中心 IP 跟轻量配置中心 IP 一致，如果不一致，请确认轻量配置中心是否通过 `Daddress.server.ip={可以访问的 IP}` 调整过 IP。
- f) 其中对应服务 `[Register-ok][Publish-ok]`，确定服务名、版本、分组都是预期的信息。
- g) 如果 `[Register-ok][Publish-ok]`，说明服务提供者正常注册到服务注册中心。开发环境只要启动成功就可以注册成功。



说明：

在实际开发过程中机器上可能会同时启动多个服务提供者，例如 A，B，C；每个服务提供者对外提供的 hsf 端口会在 12200 基础上依次累加。同时可以通过在 jvm 参数上指定 ip 和端口：`-Dhsf.server.ip=<ip> -Dhsf.server.port=<port>` 在轻量配置中心上可以检查服务端端口与实际启动的端口是否一致，不一致会导致消费端调用失败，可以通过轻量配置中心控制台更新服务端端口，或者删除掉服务，然后重新发布应用。

3. 服务消费者排查。

- a) ping jmentv.tbsite.net 确定地址服务器 IP 跟轻量配置中心 IP 一致。
- b) 启动服务消费者应用，如果之前 Tomcat 已经启动，请重启。
- c) 首先查看 Tomcat 启动日志，无异常信息，最后显示启动耗时 ms。如果有异常，请解决异常。
- d) 查看 `{userhome}/configclient/logs/configclient.log` 或 `{userhome}/logs/configclient/configclient.log`（不同版本稍有区别），查找 `Connecting to remoting://{IP 地址}` 连接的注册中心 IP 跟轻量配置中心 IP 一致，如果不一致，请确认轻量配置中心是否通过 `-Daddress.server.ip={可以访问的 IP}` 调整过 IP。
- e) 查看日志中对应的服务订阅情况，`[Data-received]` 是否收到服务提供者的具体信息。如果未收到服务提供者数据，需要确定服务提供者服务已经注册成功。
- f) 从订阅者机器上 Telnet 服务提供者 IP 端口，是否可以正常联通。如果不能正常连接说明防火墙开启或网络存在问题，需要协调相关负责人员排查。

线上环境排查

使用 SAE 管理和部署的应用都是使用的正式环境，正式环境存在严格的服务鉴权和数据隔离。由于存在鉴权，正式环境不同主账号之间的服务无法直接互相调用，同时开发环境也无法调用访问正式环境的服务。

1. 服务提供者排查。

- a) `id="codeph_hor_1ki_a0p">cat /home/admin/{taobao-tomcat 目录}/bin/setenv.sh` 此文件中 `-Daddress.server.domain={地址服务器域名}` 对应地址服务器域名。
- b) ping {地址服务器域名} 查看返回的 IP，是否正常。如果不能 ping 通说明网络存在问题，请排查网络。
- c) 清空 `/home/admin/logs/` 下所有日志，清空 `/home/admin/configclient/` 目录，清空 `/home/admin/{taobao-tomcat 目录}/logs/` 目录。
- d) 启动服务提供者应用，如果之前 Tomcat 已经启动，请重启。
- e) 首先查看 `/home/admin/{taobao-tomcat 目录}/logs/catalina.out`，此文件中无异常信息，最后显示启动耗时 ms。如果有异常，请解决异常。
- f) 查看 `/home/admin/{taobao-tomcat 目录}/logs/localhost-{日期}.log`，此文件中无异常信息。。如果有异常，请解决异常。
- g) 查看 `/home/admin/configclient/logs/configclient.log` 或 `/home/admin/logs/configclient/configclient.log`（不同版本稍有区别），其中对应服务 **[Register-ok]**

[Publish-ok]，确定服务名、版本、分组都是预期的信息。如果**[Publish or unregister error]**，请排查。

查看 catalina.out 日志中 edas.hsf.xxxx 格式的版本。

- A. 低于 edas.hsf.2114.1.0，必须创建对应服务分组，否则会鉴权失败。登录 EDAS 控制台，在左侧导航栏选择**服务市场 > 服务分组**查看应用的服务分组是否已经创建。
- B. 大于或等于 edas.hsf.2114.1.0，已经提供多租户方式隔离，可以不需要创建服务分组。对应服务会注册2次，一次是基于租户注册（总是成功），一次是基于分组注册（可能失败，但不影响服务调用）。

```
2018-07-19 10:28:44.716|ERROR|[] [] [%s] [Publish or unregister error] spas-
authentication-failed! dataId=com.alibaba.edas.testcase.api.TestCase:1.0.0
group:test error:java.lang.Error: A receivedRevision:2 tenant:DEFAULT_TENANT
2018-07-19 10:28:44.717|INFO|[] [] [Register-ok] Publisher (HSFProvider-com.
alibaba.edas.testcase.api.TestCase:1.0.0.2 for com.alibaba.edas.testcase.api.
TestCase:1.0.0)Tenant:0846c173-decf-4b47-xxxxxxx in group test in env default
2018-07-19 10:28:44.717|INFO|[] [] [Publish-ok] dataId=com.alibaba.edas.
testcase.api.TestCase:1.0.0, clientId=HSFProvider-com.alibaba.edas.testcase.api.
TestCase:1.0.0.2, datumId=ecu:ed5b9d2b-a276-4ad7-b7b9-14e432ff2356:192.
168.0.1,tenant=0846c173-decf-4b47-xxxxxxx, rev=2, env=default
```

上述 ERROR 级别日志中：使用tenant:DEFAULT_TENANT鉴权失败；使用tenant=0846c173-decf-4b47-xxxxxxx发布成功。只需保证两套鉴权成功一次即可。

- C. 如果**[Register-ok][Publish-ok]**，说明服务提供者正常注册到服务注册中心。

2. 服务消费者排查。

- a) cat /home/admin/{taobao-tomcat 目录}/bin/setenv.sh 此文件中 -Daddress.server.domain={地址服务器域名} 对应地址服务器域名。
- b) ping {地址服务器域名} 查看返回的 IP，是否正常。如果不能 ping 通说明网络存在问题，请排查网络。
- c) 清空 /home/admin/logs/下所有日志，清空 /home/admin/configclient/目录，清空 /home/admin/{taobao-tomcat 目录}/logs/目录
- d) 启动服务消费者应用，如果之前 Tomcat 已经启动，请重启。
- e) 首先查看 /home/admin/{taobao-tomcat 目录}/logs/catalina.out，此文件中无异常信息，最后显示启动耗时 ms。如果有异常，请解决异常。
- f) 查看 /home/admin/{taobao-tomcat 目录}/logs/localhost-{日期}.log，此文件中无异常信息。如果有异常，请解决异常。
- g) 查看 /home/admin/configclient/logs/configclient.log 或 /home/admin/logs/configclient/configclient.log（不同版本稍有区别，查看日志中对应的服务订阅情况，搜索

对应服务，查看[Data-received]是否收到服务提供者的具体信息。如果未收到服务提供者数据，需要确定服务提供者服务已经注册成功。

h) 从订阅者机器上 Telnet 服务提供者 IP 端口，是否可以正常联通。如果不能正常连接说明防火墙开启或网络存在问题，需要协调相关负责人员排查。

i) 相关日志排查

A. 查看 /home/admin/configclient/snapshot/DEFAULT_ENV/ 可以查到消费端服务订阅到的实际服务列表数据信息如下：

```
[root@iZ2ze26awga24ijh93152dZ com.alibaba.edas.carshop.itemcenter.
ItemService:1.0.0]# cat HSF-0846c173-decf-4b47-8aa0-xxxxxx.dat
[
  "192.168.0.1:12200?_p\u003dhessian2\u0026_ENV\
u003dDEFAULT\u0026v\u003d2.0\u0026_TIMEOUT\u003d3000\u0026_ih2
\u003dy\u0026_TID\u003d0846c173-decf-4b47-8aa0-04b5a5610096\u
u0026_SERIALIZE\u003dhessian\u0026_auth\u003dy"
]
```

B. /home/admin/logs/hsf/hsf.log 可以查看服务调用报错信息。

C. /home/admin/logs/hsf/hsf-remoting.log 记录消费端与服务端心跳检查日志，若出现如下日志：

```
01 2018-06-20 12:35:00.797 ERROR [HSF-Worker-2-thread-1:hsf.remoting] [] [] [
HSF-0085] [remoting] fail to connect: /192.168.1.1:12200 in timeout: 4000
```

该日志为消费端与提供端无法建立 TCP 长连接。

A. 检查对应的机器 IP 上的服务正常启动，对应的端口已经处于监听状态（例如 12200）。

B. 2. 第一步没有问题，可以尝试在消费端 telnet 服务端 IP Port，检查端口是否未开放。

7.2.2 错误编码：HSF-0002

报错信息

消费端报错：HSFTimeoutException

解决方案

- 机器的网络是否健康。ping 一下服务端的 IP，查看是否连通。
- 服务端处理耗时>3s，在服务端的 hsf.log 里查找业务执行超时的日志，定位具体的类及方法：
 - 服务端出现序列化错误。请检查代码：流类型、File、超大的对象等都会导致序列化错误，请不要传递。
 - 代码性能较低，请进行代码优化。
 - 服务端逻辑复杂，必须处理时间>3s，则可以修改超时时间（参见《开发者指南》）。

- 偶然出现超时，两端出现 GC。检查服务端和客户端 GC 日志，耗时很长的 GC，会导致超时。网络搜索 java gc 优化。
- 客户端负荷很高，导致客户端请求发送失败，造成超时。增加客户端机器。

7.2.3 错误编码：HSF-0003

报错信息

消费端报错：**java.io.FileNotFoundException: /home/admin/logs/hsf.log**（系统找不到指定的路径）。

解决方案：由于默认的 HSF 日志路径找不到或没权限，通过在启动时加 `DHSF.LOG.PATH=xxx` 来修改默认的路径。

7.2.4 错误编码：HSF-0005

应用运行时上报 HSF-0005 错误编码时您可以查看本文进行处理。

报错信息

启动时报错：

**java.lang.IllegalArgumentException: 未配置需要发布为服务的 Object，服务名为：
com.taobao.hsf.jar.test.HelloWorldService:1.0.zhouli**

解决方案

- provider 的 bean 缺少 target 属性，请检查配置文件。
- target 指定服务的实现类不存在，请检查配置文件。

7.2.5 错误编码：HSF-0006

报错信息

unsupported serialization [{serializeType}]

解决方案

HSF 1.x 的序列化配置类型，不在支持的范围内。请检查对端的 HSF 版本和序列化协议的配置。

7.2.6 错误编码：HSF-0007

报错信息

启动时报错：**java.lang.IllegalArgumentException: unsurpport serialize type。**

解决方案

provider 的 bean 错误配置了 `serializeType` 属性或者 `preferSerializeType` 属性，请检查配置文件。建议使用 `hessian`，`hessian2`。

7.2.7 错误编码：HSF-0008

报错信息

java.lang.IllegalArgumentException: ProviderBean 中指定的服务类型不是接口 [com.taobao.hsf.jar.test.HelloWorldServiceImpl].

解决方案

provider 的 bean 配置的 `serviceInterface` 不是接口，需要配置成接口名，请检查配置文件。

7.2.8 错误编码：HSF-0009

报错信息

java.lang.IllegalArgumentException: 真实的服务对象 [com.taobao.hsf.jar.test.HelloWorldServiceImpl@10f0a3e8] 没有实现指定接口 [com.taobao.hsf.jar.test.HelloWorldService]。

解决方案

provider 的 target 指定的 bean 没有实现接口，请确定实现类实现了对应接口。

7.2.9 错误编码：HSF-0012

报错信息

invalid route while WriteMode=unit, route : \${route}

解决方案

出现这个错误，代表 route 所指定参数位置不正确，单元化属性 route 表示的是 `userId` 在参数列表中的下标，只能取 `[0, 参数长度)`。

7.2.10 错误编码：HSF-0013

报错信息

invalid route while WriteMode=unit, route : \${route}, route parameter should be: long

解决方案

检查配置的单元化属性 route 所对应的参数类型，类型只能是 `long` 或 `java.lang.Long`

7.2.11 错误编码：HSF-0014

报错信息

**java.lang.IllegalArgumentException: ProviderBean 中指定的接口类不存在
[com.taobao.hsf.jar.test.HelloWorldService1]**

解决方案

provider 的 serviceInterface 属性配置错误，指定的接口不存在。

7.2.12 错误编码：HSF-0016

报错信息

启动 HSF SERVER 失败

解决方案

- 检查12200端口是否有冲突，一般 Server Bind 失败会造成启动失败。
- 多网卡且存在外网 IP 机器，通过-Dhsf.server.ip 来指定本地 IP。

7.2.13 错误编码：HSF-0017

报错信息

**启动时报错java.lang.RuntimeException: [ThreadPool Manager] Thread pool allocated
failed for service [com.taobao.hsf.jar.test.HelloWorldService:1.0.zhouli]: balance [600]
require [800]**

解决方案

分配的线程池不够用，HSF 默认的最大线程池是 600，可以通过配置 jvm 参数 -Dhsf.server.max.poolsize=xxx 来修改默认的全局最大线程池大小。

7.2.14 错误编码：HSF-0018

报错信息

[RuntimeInfo] Runtime information published failed.

解决方案

如果 hsf.log 中出现 [RuntimeInfo] Runtime information published failed。一般是由于 redis 服务器 down 掉，对正常的调用无影响，但在服务治理会查不到相关服务。出现这个问题，可以反馈给HSF相关负责人。

7.2.15 错误编码：HSF-0019

报错信息

白名单规则配置出错

- **[WhiteListRuleParser Component] Invalid rule: \${ruleContent}**
- **[WhiteListRuleParser Component] Parsing failed, raw rule: \${ruleContent}, parser: \${parser}.**

TPS 限流规则配置出错

- **[TPSRule Component] Invalid rule: \${ruleContent}**
- **[TPSRule Component] Parsing failed, raw rule: \${ruleContent}, parser: \${parser}.**

解决方案

规则如果解析正确，会在 `hsf-config.log` 中打印 **[TPSRule Component] Parsing OK: \${rule}**，请检查配置的规则格式是否正确

7.2.16 错误编码：HSF-0020

报错信息

WARN taobao.hsf - HSF 服务：com.taobao.hsf.jar.test.HelloWorldService:1.0.zhouli 重复初始化！

解决方案

HSF 在一个进程中，由服务名和版本号唯一区分一个服务。不支持同一个进程发布或订阅同个服务名、同个版本号，但不同组别的服务，请检查配置文件。例如配置文件中同时存在：

com.taobao.hsf.jar.test.HelloWorldService 1.0 groupA

com.taobao.hsf.jar.test.HelloWorldService 1.0 groupB

7.2.17 错误编码：HSF-0021

报错信息

启动时报错：

java.lang.IllegalArgumentException: ProviderBean 中指定的接口类不存在 [com.taobao.hsf.jar.test.HelloWorldService1]。

java.lang.IllegalArgumentException: ConsumerBean 中指定的接口类不存在[com.taobao.hsf.jar.test.HelloWorldService1]。

解决方案

HSFSpringProviderBean 的 serviceInterface 属性配置错误，指定的接口确实不存在（HSF-0014），或者 HSFSpringConsumerBean 中字段 interfaceName 指定的接口不存在（HSF-0021）。

7.2.18 错误编码：HSF-0022

报错信息

[Grouping Rule] xxx error message.

解决方案

检查归组规则的配置，有效规则必须满足：

- 分组名称不为空
- 服务名称列表不为空
- IP 规则列表不为空
- 服务名称列表中的正则表达式合法

7.2.19 错误编码：HSF-0023

报错问题

权重规则配置出错。

报错原因

权重规则解析失败、应用规则失败等错误。

解决方案

检查权重规则的配置项，在 hsf-config.log 中可以查看到 [Weighting Component] Weighting rule for service:[\${uniqueName}] parsed OK: \${weightingRule} 代表配置成功。

7.2.20 错误编码：HSF-0024

HSF 支持用户使用 Groovy 脚本配置服务的路由规则。

路由规则的配置方式，请参见 [Routing Rule Wiki](#)，在完成路由规则的配置后，您可能会发现路由规则没生效，这些没生效一般就是指调用并没有被路由到预期的机器上，然而这些没生效有可能是其他因素引起的，例如路由规则没写对、路由规则里的 IP 没有提供服务等。

这里，就来介绍下如何排查路由规则没有生效，共包含以下几步：

1. 客户端是否为本地调用、泛化调用

如果客户端通过本地调用、泛化调用的方式消费服务的话，那么是不会使用路由规则逻辑的，自然就出现了路由规则不生效这种表象。

- 本地调用：指的是一个进程即是一个服务的发布者，又是这个服务的消费者，这个时候 HSF 默认优先走本地调用，即进程内的 Java 调用，而非 RPC 调用。这种情况，是不走路由规则的逻辑的。
- 泛化调用：指的是不依赖服务的二方包，直接通过 GenericService 通过服务描述进行消费的场景，这个时候 HSF 由于没有服务的 Class，无法执行路由规则里调用业务类的逻辑，因此也不走路由规则的逻辑。

HSFOPS 上的服务测试功能走的是泛化调用，因此也不会生效路由规则。

2. 客户端是否收到路由规则

HSF 的路由规则存放在 Diamond 上，在客户端启动时，会从 Diamond 上拉取服务对应的路由规则。

在 hsf.log（一般的路径是 HSF 2.2: \${user.home}/logs/hsf/hsf-config.log 或 HSF 2.1: \${user.home}/logs/hsf/hsf.log）中搜索服务名，如果正确收到路由规则，会看到类似如下的日志：

```
01 2015-10-09 13:20:06.402 WARN [com.taobao.diamond.client.Worker.default:t.hsf] []  
[] [] [Metadata Component] Received rule for service [com.alibaba.dt.op.authclient.api.  
ResourceAPI:1.0.0.daily]: Groovy_v200907@package hqm.test.groovy  
public class RoutingRule{  
    Map<String, List<String>> routingRuleMap(){  
        return [  
            "G1":["100.69.161.201:*"]  
        ]  
    }  
  
    String interfaceRoutingRule(){  
        return "G1";  
    }  
}
```

如果没有收到路由规则，请检查：

1. 路由规则的命名是否与服务名对应，具体请参见 [Routing Rule Wiki](#)。
2. 路由规则配置在 diamond 上的环境和客户端所在的环境是否一致。

3. 路由规则是否正确解析

在 hsf.log 中搜索服务名，如果 HSF 的路由规则解析正确，会看到类似如下的日志：

```
01 2015-10-09 13:20:06.761 INFO [com.taobao.diamond.client.Worker.default:t.hsf] []  
[] [] Parse route rule succeeded, RouteRule:com.taobao.hsf.route.service.RouteRule@  
4441ec5a{  
    keyedRules={G1=[100.69.161.201:*]},
```

```
interfaceRule=G1,
methodRule={},
argsRule={}
}
```

如果 HSF 的路由规则解析失败，会看到相关的失败信息，请对照日志和 [Routing Rule Wiki](#) 检查您的路由规则。

4. 路由规则的内容是否正确

路由规则收到了、也解析对了，如果路由规则执行的效果还与预期不同，请依次检查以下内容：

1. 路由的目标机器是否真的提供了该服务

请在对应环境的 HSF 服务治理平台上查询服务，看路由的目标 IP 是否都在 Providers 列表中。

- 在老版本的 HSF 中，如果路由规则指定的 IP 没有提供服务，会报地址找不到错误
- 在新版本的 HSF 中（2.1.0.7 开始），如果路由规则指定的 IP 没提供服务，会提示开启空保护，那么此时路由规则就不再生效，HSF 会从 ConfigServer 存放的 Provider 地址列表中为客户端选取一个可用 IP 进行调用。在使用方法路由时，如果碰到算出的地址为空，这种情况下，在 hsf.log 中搜索服务名，会看到类似如下的日志。其中空保护开启的标志为倒数第二行的 EmptyProtection triggered [true]:

```
01 2015-10-09 13:20:06.761 WARN [HSF-AddressAndRule-2-thread-1:t.hsf] [] [] [
Address Component] isEmptyProtection: true
01 2015-10-09 13:20:06.761 WARN [HSF-AddressAndRule-1-thread-1:t.hsf] [] [] [
Address Component] isEmptyProtection: true
01 2015-10-09 13:20:06.761 WARN [HSF-AddressAndRule-2-thread-1:t.hsf] [] [] [
Address Component] newAllAvailableAddresses is empty for service : com.alibaba.
dt.op.authclient.api.ResourceAPI:1.0.0.daily
01 2015-10-09 13:20:06.761 INFO [HSF-AddressAndRule-2-thread-1:t.hsf] [] [] [
AddressBucket-com.alibaba.dt.op.authclient.api.ResourceAPI:1.0.0.daily] Refresh:
all amount [0], available amount [0], local preferred switch [off].Unit=UNIT
01 2015-10-09 13:20:06.761 INFO [HSF-AddressAndRule-1-thread-1:t.hsf] [] [] [
Address Component] route result com.alibaba.dt.op.authclient.api.ResourceAPI:1.0
.0.daily, addresses remain[1], EmptyProtection triggered [true]
01 2015-10-09 13:20:06.762 INFO [HSF-AddressAndRule-1-thread-1:t.hsf] [] [] [
AddressBucket-com.alibaba.dt.op.authclient.api.ResourceAPI:1.0.0.daily] Refresh:
all amount [1], available amount [1], local preferred switch [off].
```

2. 路由规则是否正确

仔细检查您的接口路由、方法路由、参数路由，是否真的写对了。

3. 地址计算优先级

如果开启了同机房规则，那么同机房的地址优先于路由规则的地址，路由规则的地址在同机房规则结果的地址集合里再计算。

路由规则分为接口级 > 方法级 > 参数级，若某一级的路由规则不存在，即其返回的 key 为 null，或在 routingRuleMap 中找不到相应值，则认为这一级对路由地址不做限制，地址取上一级的全部地址。

总体的计算过程如下：

- 接口级路由规则（正则式）最终地址列表会和同机房规则结果地址列表作交集
- 方法级路由规则（正则式）最终地址列表会和接口级地址列表作交集
- 参数级路由规则（正则式）最终地址列表会和方法级地址列表作交集

5. 路由规则是否真的被调用

如果按照以上内容排查后，都没有问题，请再确认路由规则指定的服务/方法是否真的被调用了？如果相应的方法没有被调用，则在目标机器上肯定也搜不到调用日志。

可以通过 `System.out.println()` 在路由规则中打印内容来确认，同时注意返回的内容必须为一个闭包。例如：

```
ovy_v200907@package hqm.test.groovy
public class RoutingRule {
    Map<String, List<String>> routingRuleMap() {
        return [
            "BSeller_address_filter_Key":["10.177.75.54:*"],
            "ASeller_address_filter_Key":["10.97.94.33:*"]
        ]
    }

    Object argsRoutingRule(String methodName, String[] paramTypeStrs) {
        if (methodName.startsWith("checkUrlPermission")) {
            System.out.println("Match the checkUrlPermission method.");
            return {
                Object[] args ->
                if(args[1] % 2 == 1 ) {
                    System.out.println("Route to BSeller_address_filter_Key");
                    return "BSeller_address_filter_Key";
                } else {
                    System.out.println("Route to ASeller_address_filter_Key");
                    return "ASeller_address_filter_Key";
                }
            }
        }
    }
}
```

7.2.21 错误编码：HSF-0027

报错信息

[HSF-Provider] HSF thread pool is full.

解决方案

- 服务器端某个服务处理速度过慢，不能及时处理客户端的请求，导致服务端业务执行的线程池达到最大值。HSF 默认会 dump 文件：`/home/admin/logs/hsf/HSF_JStack.log`（默认路径），查看此文件的 HSFBizProcessor-xxx 线程堆栈信息，分析性能瓶颈。
- HSF 默认的初始化线程数是 50，最大线程数是 720，可以通过配置 JVM 参数 `-Dhsf.server.min.poolsize` 和 `-Dhsf.server.max.poolsize` 进行修改。

7.2.22 错误编码：HSF-0030

报错信息

[HSF-Provider] 未找到需要调用的方法。

解决方案

- 服务端未提供此方法，登录 EDAS 控制台，在左侧菜单选择**应用管理**，单击服务提供者应用名称，跳转到应用详情页面，单击左侧菜单**服务列表**查看对应的服务是否已经发布成功。
- 新旧版本并存，调用到错误版本的服务。同上查看。
- 服务方和消费方接口不一致，例如：服务方的参数是 `java.lang.Double`，而消费方调用时使用 `double`。
- 服务端\客户端加载了不一致的接口类，请比较服务端和客户端接口 JAR 包的 MD5 是否一致。

7.2.23 错误编码：HSF-0029

报错信息

There is no TOP transformer for Service:[\${serviceUniqueName}].

解决方案

TOP 调用在服务端进行转换时，没有找到对应的 transformer。故在服务端初始化时，请通过 `com.taobao.hsf.ServiceInvokeTransform.Helper#register` 注册一个 transformer。

TOP 调用方式

TOP 调用方式 TOP 方式是为了避免网关应用依赖后端服务的 API 包。

客户端

客户端使用 `com.taobao.hsf.app.spring.util.SuperHSFSpringConsumerBeanTop`，该类无需服务端的接口包。

通过 `com.taobao.hsf.app.spring.util.SuperHSFSpringConsumerBeanTop#invoke` 发起调用。

```
/**
 * TOP调用该方法来访问远程HSF服务。
 * <p>
 * 这里需要注意的是parameterTypes参数，该参数表示要调用的方法的参数类型。必须是完整的
 * 的JAVA类名，而且如果参数是原子类型，这里的类型就是原子类型的名称。
 * </p>
 *
 * @return 对端业务响应或业务异常
 *
 * @throws HSFException
 *         如果本侧或对端HSF层出现错误，则抛出HSFException
 * @throws Throwable
 *         如果服务端业务出现异常，则抛出Exception
 */
public Object invoke(String methodName, String[] parameterTypes, Object[] args) throws
HSFException, Throwable {
    return consumerBeanTop.invoke(methodName, parameterTypes, args);
}
```

这里的 `args` 不是真正的服务端业务类型。

服务端

服务端初始化时，需要通过 `com.taobao.hsf.ServiceInvokeTransform.Helper#register` 注册一个 transformer。

```
/**
 * 用于把服务的调用参数转换成服务可以接受的形式。
 *
 * @param methodName
 *         要调用的服务方法名
 * @param args
 *         转换前的调用参数
 * @param parameterTypes
 *         要调用的服务方法的真实参数类型
 */
abstract public Object[] transformRequest(String methodName, Object[] args, String[]
parameterTypes)
    throws Exception;

/**
 * 转换请求参数类型
 *
 * @param methodName
 * @param args
 * @param parameterTypes
 * @return
 * @throws Exception
 */
abstract public String[] transformRequestTypes(String methodName, Object[] args, String
[] parameterTypes)
```

```
throws Exception;
```

该 transformer 服务在服务端将客户端传递的参数转化为真实的业务类型，HSF 使用转化后的参数进行业务调用，然后使用 transformer 将业务结果进行转化后返回给 top 客户端。

7.2.24 错误编码：HSF-0031

报错信息

[HSF-Provider] 执行 HSF 服务 xxx 的方法 xxx 耗时 xxxms，接近超时时间。

解决方案

服务端会在超时时间-实际耗时 <100ms 时，打印此日志，默认超时时间为 3s。

- 如果超时时间很短，例如小于 100ms，每次调用都会打印这个日志，请忽略这个日志。
- 如果超时时间已经很长，打印此日志，则说明业务执行慢，需要分析业务执行的性能瓶颈。

7.2.25 错误编码：HSF-0032

报错信息

please check log on server side that unknown server error happens.

解决方案

服务端在处理请求时，发生了未捕获的异常，请查看服务端的 {user.home}/logs/hsf.log 日志解决。

7.2.26 错误编码：HSF-0033

报错信息

Serilization error during serialize response.

解决方案

- 服务端在序列化返回数据时发生异常，请查看服务端的 hsf.log 日志解决。
- 如果包含 must implement java.io.Serializable 字样，请在 DO (Domain Object) 上实现一下可序列化的接口。

7.2.27 错误编码：HSF-0034

报错信息

根据用户配置的单元化参数 route 获取 userId 时发生了异常。

解决方案

检查报错服务的 route 参数与对应方法中 userId 的对应关系。

7.2.28 错误编码：HSF-0035

报错信息

RPCProtocolTemplateComponent invalid address.

解决方案

当前机器与对应的地址无法建立 TCP 连接，查看对应远端地址及端口是否可以连通。

7.2.29 错误编码：HSF-0038

报错信息

服务器在多网卡的场景下，HSF 服务端绑定在了错误的 IP 上。

解决方案

JVM 启动参数添加 `-Dhsf.server.ip=xxx.xxx.xx.xx` 指定到期望的 IP 地址。

7.2.30 错误编码：HSF-0051

报错信息

仅用于向用户提醒，不影响应用的正常运行。提醒内容包括配置错误（将使用默认值）、方法过期、HSF dump 内存失败等。

解决方案

忽略，或参考异常详情的提示进行处理，如更新配置、替换过期的方法等。

7.2.31 错误编码：HSF-0077

报错信息

HSF 向 Redis 汇报元数据失败。

解决方案

首先需要明确的是，该异常不影响 HSF 服务的发布、订阅和调用。一般是由于 Redis 服务器不稳定或连接时网络环境抖动造成。可忽略，或通过重启应用重试。

7.2.32 错误编码：HSF-0089

报错信息

解析订阅的 Diamond 配置时，出现了异常。

排查步骤

该错误码表明 HSF 订阅的某些 diamond 配置，在解析时出现了异常。例如，路由规则格式不正确等。

diamond 配置的解析一般不会影响应用正常启动和服务调用。

具体的信息请参考 `${user.home}/logs/hsf/hsf.log` 和 `${user.home}/logs/hsf/hsf-config.log` 中对应的异常信息，判断是什么规则出错的，对这个规则做修正即可。

7.3 Ali-Tomcat 问题

7.3.1 Ali-Tomcat 启动失败

Ali-Tomcat 会因为各种错误导致启动失败，需要查看 `catalina.out` 及 `localhost.log` 这两个日志文件进行定位。如果使用 tomcat4e 插件，则可以在 Eclipse 的控制台看到具体的问题描述。

7.3.2 com.taobao.diamond.client.impl.DiamondEnvRepo 初始化失败

错误信息

Could not initialize class com.taobao.diamond.client.impl.DiamondEnvRepo

解决方案

地址服务器中 Diamond 数据为空，需要确信地址服务器配置正确且稳定，访问 `http://jmenv.tbsite.net:8080/diamond-server/diamond`，能正常返回。



说明：

访问 <http://jmenv.tbsite.net:8080/diamond-server/diamond>

- Windows: C:\Windows\System32\drivers\etc\hosts
- Linux: /etc/hosts

7.3.3 未知实例异常

错误信息

Caused by: java.net.UnknownHostException: iZ25ax7xuf5Z

解决方案

iZ25ax7xuf5Z 是当前实例名，确定 `/etc/hosts` 存在本机 IP 及当前实例名的配置。如果没有请添加，例如：192.168.1.10 iZ25ax7xuf5Z

7.3.4 为什么我在代码中配置的 Tomcat 端口不生效？

当您的应用部署到 EDAS 后，如果遇到以下类似问题，均可参考此文档解决。

- 为什么我的 SpringBoot 应用，在 application.properties 文件中配置的 tomcat 端口不生效？
- 为什么我在控制台修改了 tomcat 端口后，在机器上查看时，发现监听在此端口的并不是 tomcat 进程？

其实这些是正常现象，详情如下：

由于 EDAS 实现流量灰度功能的需要，在 EDAS 中，所有使用了流量灰度的应用。服务请求都是先经过 Tengine，然后再由 Tengine 转发到 Tomcat 的。

为了使得 Tengine 能够顺利找到 Tomcat 的端口，所以在使用了 Tengine 的情况下，EDAS 将 Tomcat 端口固定成了 65000。

形如这种配置

```
server.port=28082
```

端口其实还是 65000，因为我们在启动参数中默认添加了 `--server.port=65000`

所以观察 tomcat 是否启动成功时，需要注意，tomcat 是监听在 65000 端口的。

修改应用端口，其实修改的是 Tengine 的端口

同样的，修改应用端口，其实修改的是 Tengine 的端口，形如以下的控制台配置。

应用设置

JVM

Tomcat

负载均衡

健康检查

基本信息

点击高级配置，可以对server.xml全文进行配置，应用启用高级配置后，应用分组将使用应用配置。

配置项	配置内容（若无自定义值输入，系统将沿用父级配置。）
应用端口：	8080
Tomcat Context：	☐ 程序包名字 ☒ 根目录 ☐ 自定义
最大线程数 ②	400
Tomcat 编码 ②：	ISO-8859-1 v ☒ useBodyEncodingForURI
高级设置 v	

配置Tomcat

取消

将 Tengine 的端口改成了 8080，tomcat 的端口仍为 65000。

7.4 Pandora 问题

7.4.1 如何设置用户日志不被覆盖

问题现象

使用 EDAS 后，Log4j 日志无法输出。

可能原因

Log4j 日志被覆盖会导致日志无法输出。

解决方案

1. Log4j 不要使用默认名称，会被自动屏蔽，将配置文件改为其他名称，例如放在 WEB-INF/log4j-myAppName.xml。
2. 配置 Spring 的 Log4jListener，在 web.xml 中增加如下配置。

```
<listener>
  <listener-class>org.springframework.web.util.Log4jConfigListener</listener-class>
</listener>
<context-param>
  <param-name>log4jConfigLocation</param-name>
  <param-value>WEB-INF/log4j-myAppName.xml</param-value>
</context-param>
```

myAppName 是应用自定义的名字，可以根据情况进行定义。

3. 启动应用，Pandora 会首先加载内置的 Log4j.jar，同时加载 Pandora 内部配置的 Log4j 配置文件，此时不会加载业务应用的配置。
4. Pandora 加载完成后，在应用的 web.xml 加载时会执行 Log4jConfigListener 加载业务自定义的 Log4j 配置，保证业务的 Log4j 配置生效。

7.4.2 Pandora 启动失败，QoS 端口绑定异常

问题现象

错误信息：Cannot start pandora qos due to qos port bind exception

解决方案

QoS 端口被占用，处理方式同轻量配置中心端口冲突解决办法，具体请参见[执行 startup.bat 和 startup.sh 不能正常启动](#)

7.4.3 Pandora 版本问题

当您的应用程序启动或运行时，如果您发现异常堆栈中出现以下内容，且对应的类名都是以 com.taobao、com.alibaba、com.aliware 开头，表示您使用的 Pandora 版本不是最新版，部分功能不支持。

1. java.lang.LinkageError
2. java.lang.ClassNotFoundException
3. java.lang.NoSuchMethodError
4. java.lang.NoClassDefFoundError
5. org.springframework.beans.NotWritablePropertyException: Invalid property 'xxxxxx' of bean class [com.alibaba.xxxxxx]:

6. RuntimeException("Can not load this fake sdk class")

出现如上问题，您需要升级 Pandora。

在EDAS 环境中升级

在应用详情页单击**容器版本**，选择最新版本，单击**升级到该版本**。

容器版本			
版本号	描述	发布日期	操作
3.4.1	支持 DubboX 的通信	2018-03-15	升级到该版本
3.4.0	EDAS 与 ACM SDK 兼容版本	2018-03-07	✓
3.3.9	EDAS 与 ACM SDK 兼容版本	2018-01-17	降级到该版本
3.3.6	修复 EagleEye 调用链中展示 Unknown 所导致应用拓扑大屏无法展示的问题	2017-12-20	降级到该版本
3.3.5	HSF V2.2 支持 ACM	2017-11-30	申请使用此版本
3.3.4	测试版本，请尽快升级	2017-11-16	此版本不可用

在 Ali-Tomcat + Pandora 中升级

如果您在 Ali-Tomcat + Pandora 容器中开发应用，升级 Pandora 的步骤如下。

1. 下载 [Pandora 容器](#)。
2. 删除原有的 Pandora 版本文件。

```
rm -rf /home/admin/tomcat/deploy/taobao-hsf.sar
```

Pandora 需要安装在 Ali-Tomcat 的安装路径下。上面的命令示例中 Ali-Tomcat 的安装路径为 **/home/admin/tomcat**。如果您的 Ali-Tomcat 安装在其它路径，请使用实际路径替换。

3. 将新下载的 Pandora 压缩包移动到 Ali-Tomcat 的 deploy 文件夹下。

```
mv /home/admin/downloads/taobao-hsf.tgz /home/admin/tomcat/deploy
```

- **/home/admin/downloads/taobao-hsf.tgz** 为 Pandora 本地下载路径。
- **/home/admin/tomcat** 为 Ali-Tomcat 安装路径。

如果 Pandora 下载或 Ali-Tomcat 安装在其它路径，请使用实际路径替换。

4. 解压 Pandora 压缩包。

```
tar -zxvf /home/admin/tomcat/deploy/taobao-hsf.tgz
```

在 PandoraBoot 中升级

如果您通过 PandoraBoot 来使用 Pandora，升级 Pandora 只需直接在工程打包时添加强制更新的参数 **-u** 即可。

列如在使用 Maven 构建时，直接在工程路径下执行 `mvn clean package -U` 即可完成 Pandora 的版本更新。

本地开发环境如何查看 Pandora 的版本

应用正常启动后，在开发环境的 console 中可以看到 Pandora 信息，其中 **SAR Version** 为 Pandora 版本信息。

```
*****
**
**                                Pandora Container                                **
**
** Pandora Host:      10.81.32.236                                           **
** Pandora Version:   2.1.4                                                 **
** SAR Version:       edas.sar.V3.4.1                                       **
** Package Time:      2018-03-14 20:11:35                                   **
**
** Plug-in Modules: 18                                                       **
**
** metrics ..... 1.6.4-SNAPSHOT                                             **
** edas-assist ..... 1.9                                                     **
** pandora-qos-service ..... edas215                                         **
** pandolet ..... 1.0.0                                                       **
** spas-sdk-client ..... 1.2.7                                               **
** eagleeye-core ..... 1.6.4-SNAPSHOT                                         **
** tddl-driver ..... 1.0.5-SNAPSHOT                                           **
** vipserver-client ..... 4.6.8-SNAPSHOT                                       **
** diamond-client ..... 3.8.7                                                 **
** configcenter-client ..... 1.0.3                                           **
** spas-sdk-service ..... 1.2.7                                               **
** config-client ..... 1.9.4                                                  **
** unitrouter ..... 1.0.11                                                    **
** monitor ..... 1.2.3-SNAPSHOT                                               **
** sentinel-plugin ..... 2.12.11                                              **
** ons-client ..... 1.7.1-EagleEye-SNAPSHOT                                   **
** hsf ..... 2.2.5.0-Dubbo                                                    **
** pandora-framework ..... 2.0.8                                              **
**
** [WARNING] All these plug-in modules will override maven pom.xml dependencies. **
** More: http://gitlab.alibaba-inc.com/middleware-container/pandora/wikis/home **
**
*****
```

7.5 轻量级配置及注册中心问题

7.5.1 更新配置后，相关应用中的配置未生效

如果您在更新配置后，相关应用中的配置未生效，请按本文提供的方法尝试解决。

现象

在轻量级配置及注册中心更新了配置，但配置在应用中未生效。

可能的原因

- 轻量级配置及注册中心的监听查询中应用程序所在的机器 IP 配置错误。
- 轻量级配置及注册中心中监听添加异常。
- 配置推送异常。
- 轻量级配置及注册中心和应用中的 dataId/Group 配置不一致。

处理办法

1. 登录轻量级配置及注册中心控制台，在左侧导航栏选择**配置管理 > 配置列表**。
2. 在**配置列表**页面找到指定配置项，在操作列单击**更多**，在下拉菜单中选择**监听查询**。
3. 查看该配置项有哪些 IP 在监听。
 - 如果应用程序所在的机器 IP 没有在监听列表中，在应用程序中检查是否正确配置了 server 地址。
 - 如果配置不正确，修改配置。
 - 如果配置正确，查看日志来定位是否添加监听成功。
 - 使用 ACM 的应用的日志文件为 `${user.home}/logs/diamond-client/diamond-client.log`
 - 使用 Nacos Config 的应用日志文件为 `${user.home}/logs/nacos/config.log`
 - 在日志中查找搜索关键字 **custom-serverlist** 的日志行，确认程序运行时连接的 server。
 - 在日志中查找搜索关键字 **add-listener** 的日志行，确认程序中是否添加配置监听成功。
 - 如果程序所在的机器 IP 在监听列表中，查看以下日志来确定最新的配置是否成功推送到指定的客户端。
 - 对于使用 ACM 的应用，在日志文件 `${user.home}/logs/diamond-client/diamond-client.log` 中搜索包含关键字 **data-received** 的日志行，确定最后一次数据接收到的时间和内容。
 - 对于使用 Nacos Config 的应用，在日志文件 `**${user.home}/logs/nacos/config.log**` 中搜索包含关键字 **notify-ok** 或者 **notify-listener** 的日志行，确定最后一次数据接收到的时间和内容。

4. 检查应用程序中指定的 **dataId/Group** 是否和控制台设置的一致。

- 如果不一致，修改应用程序中的 **dataId/Group**。
- 如果一致，查看以下日志来确定最新的配置是否成功推送到指定的客户端。
 - 对于使用 ACM 的应用，在日志文件 `${user.home}/logs/diamond-client/diamond-client.log` 中搜索包含关键字 **data-received** 的日志行，确定最后一次数据接收到的时间和内容。
 - 对于使用 Nacos Config 的应用，在日志文件 `**${user.home}/logs/nacos/config.log**` 中搜索包含关键字 **notify-ok** 或者 **notify-listener** 的日志行，确定最后一次数据接收到的时间和内容。

7.5.2 服务调用不通

如果您遇到服务调用不通的问题，请按本文的方法尝试解决。

现象

在轻量级配置及注册中心中，服务消费者调用服务提供者失败。

可能的原因

- 服务在轻量级配置及注册中心注册失败。
- 服务的网络发生切换。
- 服务推送失败。

处理办法

1. 在轻量级配置及注册中心上查看服务是否注册成功。
 - 注册成功，进行步骤 2。
 - 注册失败，进行步骤 5。
2. 查看服务所在的实例是否发生网络切换，即服务提供方的 IP 变了。
 - 发生了网络切换，执行步骤 3。
 - 未发生网络切换，执行步骤 4。
3. 重新启动应用，使用最新的 IP 注册成功后，再次检查调用是否成功。
 - 调用成功，结束。
 - 仍然调用不通，执行步骤 4。

4. 查看轻量级注册及配置中心的推送日志（`${安装目录}/logs/config-server-push.log`），排查服务是否推送成功。

- 如果服务端推送成功，则在消费端的机器或者容器的日志文件中查看是否接收到最新的服务 IP 列表或者观察是否 ERROR 级别的日志。

基于不同方式开发的应用，日志文件的路径不同：

- 基于 HSF/Dubbo 开发的应用：`${user.home}/logs/configclient/config-client.log`
- 基于 ACM 开发的应用：`${user.home}/logs/vipsrv-logs/vipclient.log`
- 基于 Nacos 开发的应用：`${user.home}/logs/nacos/naming.log`

- 如果服务端推送不成功，请联系 EDAS 技术支持人员。

5. 在轻量版配置及注册中心上查看不到该服务的注册。这个时候在应用所在的机器或者容器内查看`${user.home}/logs/configclient/config-client.log`这个日志文件，看是否注册发生异常。

7.5.3 服务注册慢

如果您遇到服务注册慢的问题，请按本文的方法尝试解决。

现象

本地服务运行后，注册到轻量级配置及注册中心很慢（超过 1 s）。

可能的原因

- 轻量级配置及注册中心问题
- 开发框架处理注册逻辑问题

处理办法

在日志文件中查看服务注册到轻量级配置及注册中心的耗时。

基于不同方式开发的应用，日志文件的路径不同：

- 基于 HSF/Dubbo 开发的应用：`${user.home}/logs/configclient/config-client.log`
- 基于 ACM 开发的应用：`${user.home}/logs/vipsrv-logs/vipclient.log`
- 基于 Nacos 开发的应用：`${user.home}/logs/nacos/naming.log`

如果服务注册到轻量级配置及注册中心的耗时在 50 ms 以内，但服务发布仍然慢（大于 1 s），请联系 EDAS 技术支持人员协助确认开发框架层面在发布服务时做了哪些处理，导致服务发布慢。