使用如下命令编译和打包镜像：

mvn clean package; docker build -t harborcluster-dev.casccloud.cn:31104/tsf\_100000000/mesh-provider:v1.0 .  
mvn clean package; docker build -t harborcluster-dev.casccloud.cn:31104/tsf\_100000000/mesh-consumer:v1.0 .

注意事项：

* Consumer工程的RestTemplate的目标地址为mesh-provider:18081。注意：mesh-provider名字不是由Provider工程的$spring.application.name定义的，而是spec.yaml中$spec.services.name定义的。
* 虽然Consumer工程定义了@LoadBalanced Template，但是这里和普通Template对象功能一样，因为LoadBalance是由Sidecar处理的。所以，所以实际应用中，可以去掉。
* Consumer工程同时演示了如何使用FeignClien访问服务。
* Consumer工程演示了如何传递用于调用链追踪的headers。注意，如果Provider工程未传递相应的headers，这是因为Provider本身即是调用链的最末端。
* TSF mesh应用的调用链是由sidecar处理的，注意这里并未使用TSF Spring Cloud应用类似的调用链日志方式收集，虽然TSF agent sidecar容器依然存在。
* Provider和Consumer工程使用JDK11编译。其中的Dockerfile及entrypoint.sh可以作为其他项目的模板。

参见如下演示项目输出的调用链截图。这里是登录到provider agent容器，执行如下命令，也可以通过NodePort方式访问mesh-consumer服务。

curl mesh-consumer:18083/echo-rest/yy01

curl mesh-consumer:18083/echo-lb/yy02

curl mesh-consumer:18083/echo-feign/yy03

图片包含 截图, 监控, 电脑, 屏幕

描述已自动生成

图片包含 游戏机, 截图

描述已自动生成

图片包含 游戏机, 截图

描述已自动生成

图片包含 游戏机, 截图

描述已自动生成