Web

HAProxy

Gateway

Gateway

Discovery Cluster

Config Cluster

Some

Some

Feign

Feign

Git

Person

Person

Redis

DB

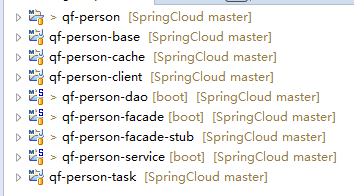
Turbine

Admin

Sleuth

基于Spring Cloud的架构设计图详解：

1. 采用Git存放各个环境所需要的配置文件。
2. 采用Spring Cloud Eureka来实现注册中心，此注册中心会部署到三台服务器上，作为集群供其他组件来注册和查找。
3. 采用Spring Config Server来实现配置中心，所有的配置中心的文件都从Git中获取，之后供其他组件来从此配置中心查询其所需要的配置，这里也会部署到至少两台服务器作为集群供其他组件来获取配置。
4. DB、Redis等常用的模块。
5. 对于底层的各个服务模块（Person）来说，每个模块至少部署到两台服务器上，之后供调用者采用负载均衡的方式来调用，对于单个的服务模块来说其代码结构为：



这是一个maven项目，其中包含base、façade-stub、façade、service、redis、dao、task等。

Redis

Service

Facade

Client

DAO

Façade-Sub

Base

Base 模块：主要做Util模块来用，里面放置一些本模块内部使用的公共方法与常量。

DAO模块：主要放置本模块中与数据库相关的操作都放置到此模块中。此处使用MyBatis来作为数据库层的处理方案。

Redis模块：主要是放置与Redis相关的操作都放到此模块中，此处使用处理Redis集群处理技术来进行处理。

Client模块：主要是处理与其他模块系统集成相关的操作都放到此模块中进行处理。

Service模块：业务处理逻辑都放置到这个模块进行处理。

Façade-Sub模块：所有供其他系统来调用的接口和调用所使用的request、response和constants都放置在这里。其他系统在使用的时候直接采用fastJson将参数和结果在类和JSON之间进行转换。

Façade模块：此模块中的类实现Façade-sub中的接口，在此模块中主要是做一些参数校验以及对Service模块进行调用并将结果返回。

（6）模块Some这一层也是一个服务，此服务可以直接调用DB、Redis等基础组件，也可以调用其他服务（如Person服务），调用Person服务通过Feign方式来调用。

（7）模块zuul这一层主要是做统一网关来用，即前端所有的请求与多个后端服务解耦，前端直接调用zuul这一层，之后再由zuul调用其他后端服务，同时还可以在这个模块中添加一些类似于servlet中filter的功能做一些统一处理的事情。

（8）HAProxy做一下Zuul层的loadbalance。

（9）最上层是Web或者APP层，直接调用下层服务。

（10）Turbine模块主要是负责实时监控整个断路器、服务调用状态等信息。

（11）Admin模块主要是负责监控整个系统中各个微服务的机器、本身的状态。

（12）Sleuth模块主要是负责调用链路追踪。

接下来技术方面还需要做的事情：

1. 找出更好的监控，能显示出每个服务、每台主机调用次数以及多少成功、多少失败。
2. 与ELK集成搭建日志系统。
3. 研究RabbitMQ，提供消息系统的解决方案。
4. 与Docker集成，研究k8s实现与其集成。
5. 与Jenkins集成，完成整套环境的部署。