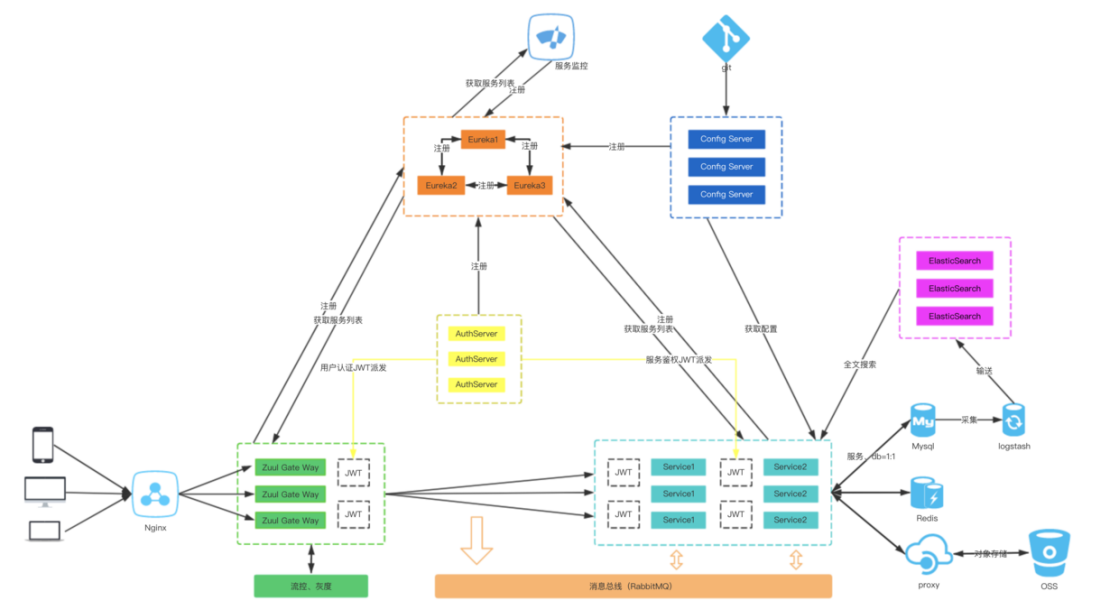


# 落地架构概览



# 部署指南

## 基础环境

- mysql
- redis
- RabbitMQ

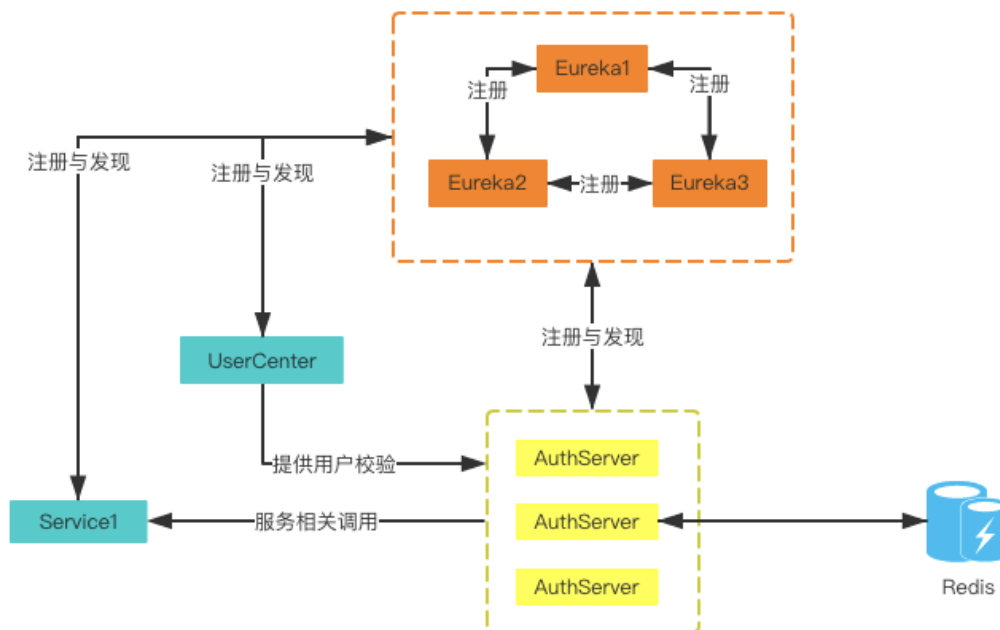
## 注册中心集群部署

采取Eureka本身互为节点的方式，进行集群部署。

```
eureka:
  instance:
    statusPageUrlPath: /info
```

```
healthCheckUrlPath: /health
# docker 部署开启
prefer-ip-address: true
ip-address: 127.0.0.1
client:
  serviceUrl:
    defaultZone:
http://${EUREKA_HOST1:localhost}:${EUREKA_PORT1:8761}/eureka/,http:
//${EUREKA_HOST2:localhost}:${EUREKA_PORT2:8761}/eureka/
```

## 鉴权中心集群部署



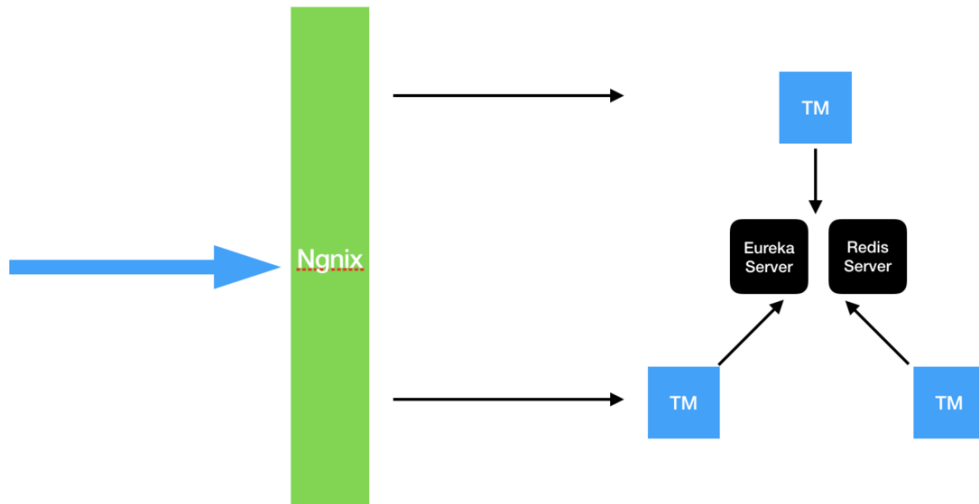
鉴权中心的集群部署本身是基于Eureka的来实现的，而调用方也是通过客户端ribbon负载的方式来管控。其中关于密钥与公钥的派发一致性，则是通过redis来进行控制的。

## 用户中心（业务服务类似）集群

同鉴权中心，一样是基于Eureka的客户端来实现的。

# 事务中心集群部署

原理图:



1. 配置Redis集群
2. 配置TM服务 部署多分tm，然后修改各个配置文件的 `eureka.client.serviceUrl.defaultZone`，指向各服务的tm地址中间用", "分割。

例如：

<http://192.168.1.101:8761/eureka/>,<http://192.168.1.102:8761/eureka/>,<http://192.168.1.103:8761/eureka/>

3. 配置nginx负载均衡

负载均衡TM服务。

4. 修改各事务模块的tx.properties配置文件url地址参数为nginx负载均衡的地址。

## 前端部署

将前端代码编译后生成的 `dist` 文件夹，放置部署某个服务器上，利用nginx进行指向

## 部署nginx配置参考

```
location / {  
    # 指向我们打包后上传的前端文件  
    root /opt/nginx/dist;  
    index index.html;  
}  
location /jwt/ {  
    # 转发请求到后端服务网关  
    proxy_pass http://127.0.0.1:8765/jwt/;  
}  
# 通过路径前缀将请求转发到后端服务器  
location /api/ {  
    proxy_pass http://127.0.0.1:8765/api/;  
}
```