



《手写OS操作系统》小班二期招生，全程直播授课，大牛带你掌握硬核技术！

点此查看

从所有教程的词条中查询...

首页 > 慕课教程 > Go工程师体系课全新版 > 1. 什么是服务注册和发现

- 全部开发者教程
2. git的简单规范

3. go代码规范

4. go项目目录规范

5. 微服务该采用multi-repo还是mono-repo?

6. 微服务的目录结构 (mono-repo)

7. go代码的检测工具

8. go中常见的错误

第22周 设计模式和单元测试

1. go最常用的设计模式 - 函数选项

2. 单例模式和懒加载

3. 测试金字塔

第23周 protoc插件开发、cobra命令行

1. protoc调试源码

2. protoc自定义gin插件

第24周 log日志包设计



bobby · 更新于 2022-11-08

◀ 上一节 9. redis的安装 2. consul的安装... 下一节 ▶

1. 什么是服务注册和发现

假如这个产品已经在线上运行，有一天运营想搞一场促销活动，那么我们相对应的【用户服务】可能就要新开启三个微服务实例来支撑这场促销活动。而与此同时，作为苦逼程序员的你就只有手动去 API gateway 中添加新增的这三个微服务实例的 ip 与port，一个真正在线的微服务系统可能有成百上千微服务，难道也要一个一个去手动添加吗？有没有让系统自动去实现这些操作的方法呢？答案当然是有的。当我们新添加一个微服务实例的时候，微服务就会将自己的 ip 与 port 发送到注册中心，在注册中心里面记录起来。当 API gateway 需要访问某些微服务的时候，就会去注册中心取到相应的 ip 与 port。从而实现自动化操作。

2. 技术选型

Consul 与其他常见服务发现框架对比

名称	优点	缺点	接口	一致性算法
zookeeper	1.功能强大，不仅仅只是服务发现 2.提供 watcher 机制能实时获取服务提供者的状态 3.dubbo 等框架支持	1.没有健康检查 2.需在服务中集成 sdk，复杂度高 3.不支持多数据中心	sdk	Paxos
consul	1.简单易用，不需要集成 sdk 2.自带健康检查 3.支持多数据中心 4.提供 web 管理界面	1.不能实时获取服务信息的变化通知	http/dns	Raft
etcd	1.简单易用，不需要集成 sdk 2.可配置性强	1.没有健康检查 2.需配合第三方工具一起完成服务发现 3.不支持多数据中心	http	Raft

9. redis的安装 ◀ 上一节 下一节 ▶ 2. consul的安装和配置

✎ 我要提出意见反馈

