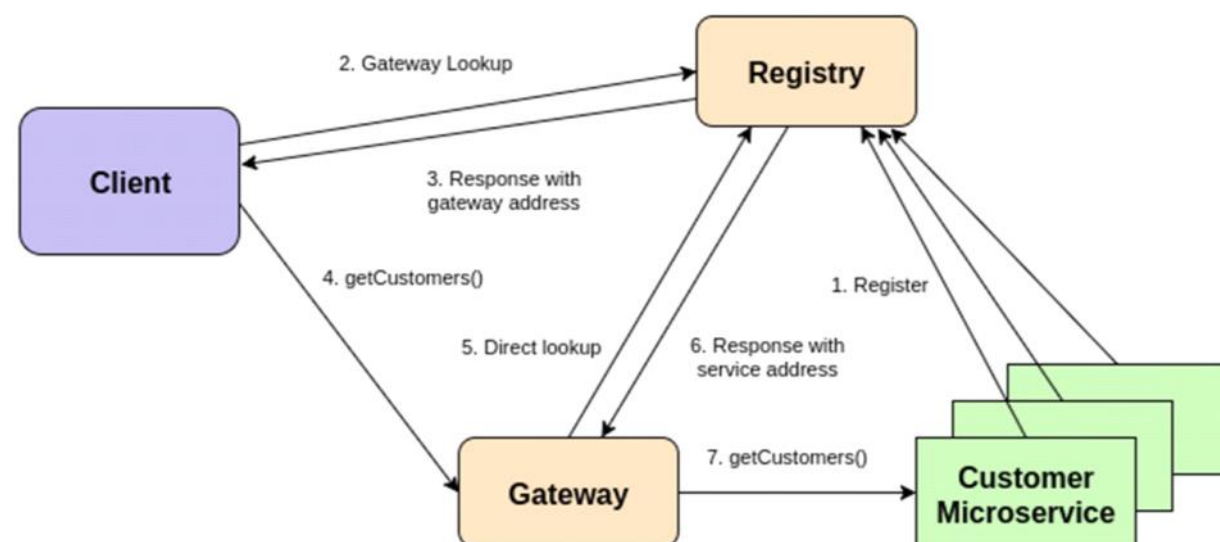
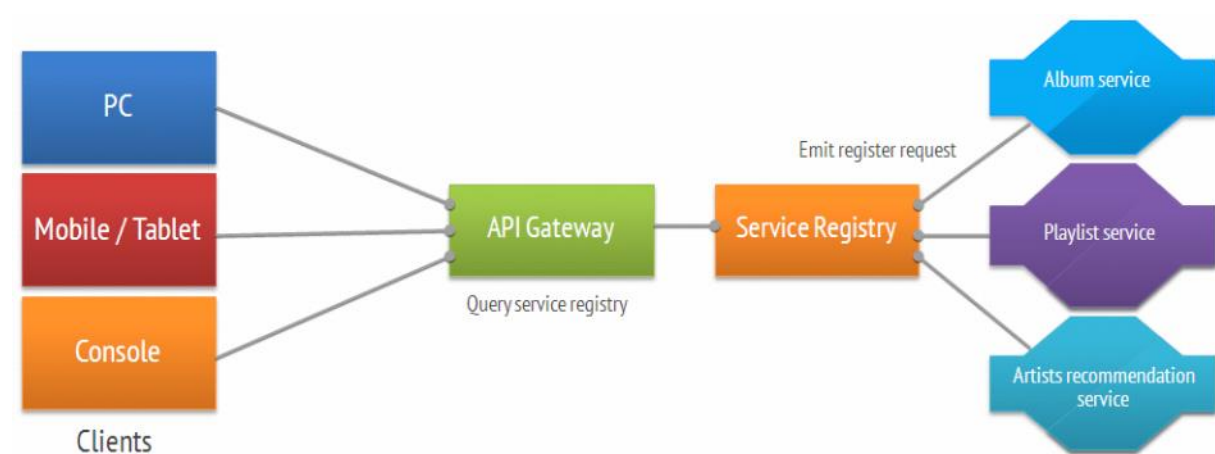


第20期-微服务的注册与发现

2020年3月12日 21:00

服务要被使用，就需要对外提供服务的位置信息，这个位置信息通常是一个IP地址+端口。在服务只有单个实例并且地址不会动态变化的情况下，服务的位置在使用端可以通过配置文件甚至代码等方式固定死。但在位置信息会动态发生变化的情况下，服务实例就需要将这个地址注册到一个注册中心。

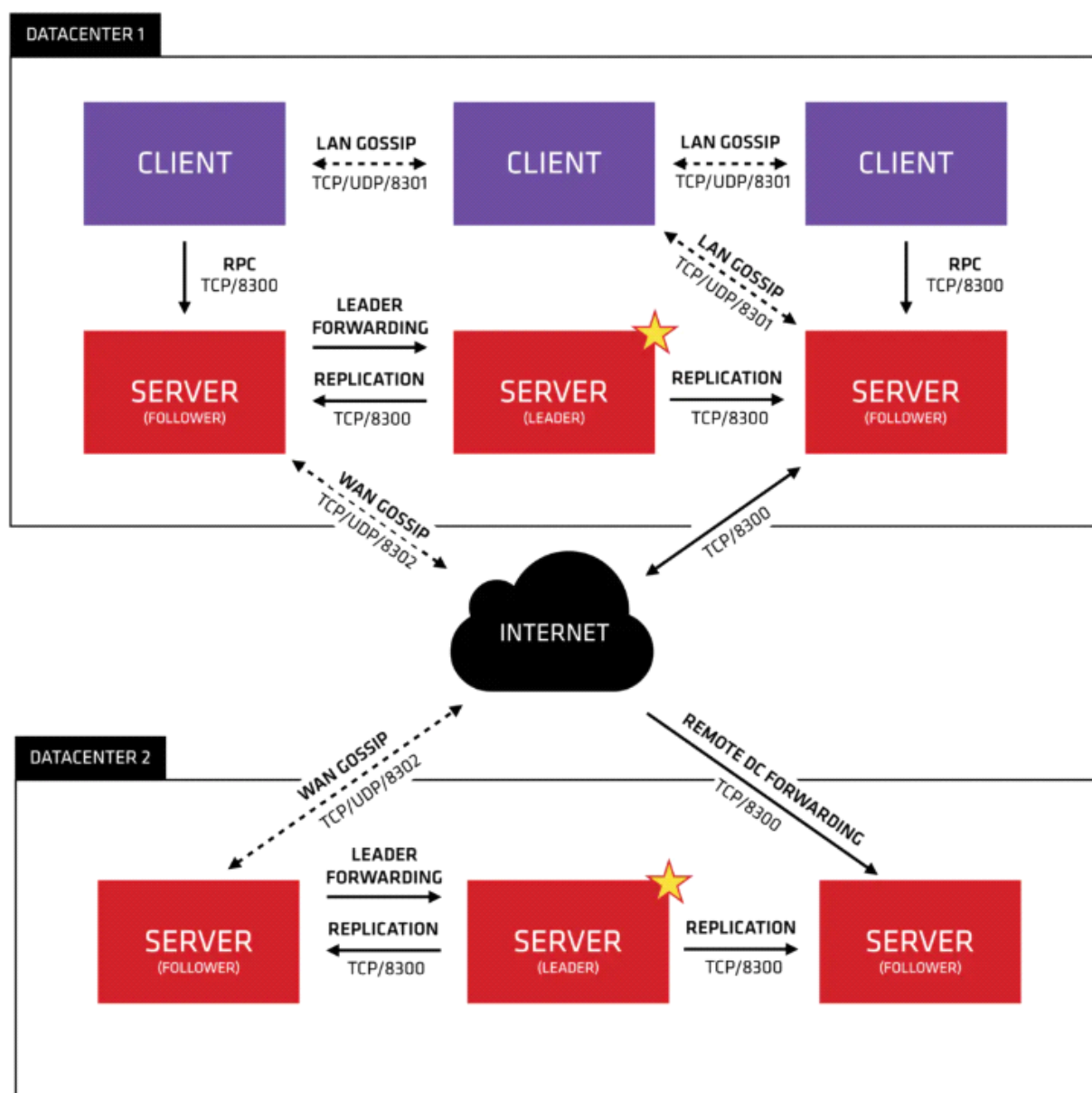


主流的服务注册与发现框架

ZooKeeper、etcd、Eureka、Consul 、Nacos

学习 Consul 框架

开源，提供服务注册发现、健康检查、负载均衡和全局分布式的键值存储。



常用命令

集群状态查看: `consul operator raft list-peers`

查看 members 状态: `consul members`

查看当前节点信息: `consul info`

服务元数据地址

<http://127.0.0.1:8500/v1/catalog/service/servicea>

参考资料

[微服务可寻址性和服务注册表](#)

[.NET Core 微服务之基于Consul实现服务治理](#)

[.NET Core + Consul 服务注册与发现](#)

[ASP.NET Core gRPC 使用 Consul 服务注册发现](#)

[.NET Core + Ocelot + IdentityServer4 + Consul 基础架构实现](#)

[轻松构建微服务之服务注册和发现](#)

集成第三方框架

[NuGet package that provides Consul support in Ocelot](#)

[Load Balancing with NGINX and Consul Template](#)

[Getting Started With Consul and Kubernetes](#)

[Configuring and Running Envoy](#)

