T13:43 “org.springframework.cloud.netflix.eureka.EurekaDiscoveryClient”从命名看，实现的是对Eureka发现服务的封装。EurekaDiscoveryClient依赖了EurekaClient接口，EurekaClient继承了LookupService接口，它们都是Netflix开源包中的内容，主要定义了针对Eureka的发现服务的抽象方法，而真正发现服务的是com.netflix.discovery.**DiscoveryClient**类

T14:08 关键部件

T14:12 写专利，微服务中的版本管理，两个版本是否兼容，升中版本，大版本

T14:55 客户端可以缓存从服务注册表处获得的网络地址。

一个服务实例通过POST请求来注册自己的网络位置，每隔30秒通过一个PUT请求重新注册。

注册表中的一个条目会因为一个HTTP DELETE请求或实例注册超时而被删除。

客户端通过一个HTTP GET请求来检索注册的服务实例。

DNS TEXT records用来存储Eureka集群配置，实际上是从可用区到Eureka服务器网络地址的列表的映射。当一个Eureka服务器启动时，会向DNS发送请求，检索Eureka集群的配置，定位节点，并为自己分配一个未占用的弹性IP地址。

T15:16 其他服务注册的例子：

●Etcd：一个高可用，分布式，一致的key-value存储，用来共享配置和服务发现。

●Consul：一个发现和配置服务的工具。客户端可以利用它提供的API，注册和发现服务。Consul可以监控监测来实现服务的高可用

●Apache Zookeeper：一个常用的，为分布式应用设计的高可用协调服务，最开始Zookeeper是Hadoop的子项目。

T15:44 微服务服务之间相互独立，所以不同的服务可以使用不同的语言来开发，或者根据业务的需求使用不同类型的数据库。

T15:56 微服务面临的挑战：

1. 运维开销
2. DevOps要求：开发团队需要保证一个Tomcat集群可用，保证一个数据库可用，这就意味着团队需要高品质的DevOps和自动化技术，全栈式人才很少。
3. 隐式接口：当一个服务更改接口格式时，可能涉及到此接口的所有服务都需要做调整。
4. 重复劳动：同一个功能在不同服务中可能多次重复编写
5. 分布式系统的复杂性：考虑网络延迟、容错、消息序列化、不可靠的网络、异步、版本控制、负载等
6. 跨进程的十五，大量的异步处理、多个微服务之间的整体测试都需要有一整套的解决方案，而现在，这些技术还未成熟。