《11\_ZooKeeper、Eureka、Consul、Nacos的选型对比》

zookeeper、eureka、consul、nacos

zookeeper和eureka，consul用的没那么多，nacos现在用的越来越多，以后也会是一个大的趋势，但是现在可能还没那么的普及

zookeeper的基础知识以及基本的架构原理，互联网Java工程师面试突击第三季，里面是有讲过zk的一些架构原理，如果大家不了解的话，可以去看一下

zookeeper的原理，leader+follower，leader写，同步到follower，follower可以读，保证顺序一致性，就是基本尽量保证到数据一致的，主动推送，典型的CP，leader崩溃的时候，为了保证数据一致性，尽量不要读到不一致的数据，此时要重新选举leader以及做数据同步，此时集群会短暂的不可用，CP

服务注册中心选型对比的时候，其他的分布式系统选型的时候，CP，AP

P简单来说就是任何分布式系统一般都会满足，他就是分区容错性；CP，C，一致性，尽可能的去保证你读取到的数据是一致的，牺牲掉一个A，可用性，一旦leader崩溃，zk可能会有一个短时间内，几十s有可能，集群不可用，此时需要重新选举一个leader，然后再做数据同步，保证数据一致性之后再开放让你来读取

consistency，availability

关于eureka的一些架构原理，21天互联网Java工程师面试训练营（分布式篇），儒猿技术窝，重点讲解了eureka的一些架构原理

eureka的原理，peer-to-peer，大家都能写也都能读，每个节点都要同步给其他节点，但是是异步复制的，所以随时读任何一个节点，可能读到的数据都不一样，任何一个节点宕机，其他节点正常工作，可用性超高，但是数据一致性不行，AP

Consul也是基于raft算法的CP模型

Nacos也是基于raft算法的CP模型，同时也支持配置成类似eureka的AP

其实CP或者AP也都行，CP就是偶尔可能短时间不可用，AP就是可能数据不一致，两个都有问题，但是在生产环境下，无论CP还是AP其实都用的很多

其实说白了，zk作为注册中心是早期dubbo时代的标配；后续spring cloud进入国内市场，大家就都用eureka了，但是spring cloud也推荐了consul，所以consul也有不少人在用，zk、eureka、consul，其实都有人用

但是未来还是建议大家用nacos，因为nacos的功能最为完善，包括了雪崩保护、自动注销实例、监听支持、多数据中心、跨注册中心同步、spring cloud集成、dubbo集成、k8s集成，这些都支持，其他的几个技术基本都支持部分罢了