稳定性建设

发布工程

灾备系统

混沌工程

故障自愈

基于“主备”的冗余设计

■存在切换不成功、切换后状态不一致等通病

■要主节点故障导致主备切换后，一般需要尽早跟进问题并修复节点

基于“负载均衡”的设计

基于平台的设计

■云平台，包括k8s系统上的自动恢复和扩缩容功能本身就是故障自愈的实现

基于业务架构的设计

■许多分布式自身就能支持自愈机制，且大部分都涉及到了CAP原则

■例如Zookeeper、Etcd等

灾备建设

同机房灾备

■通常在同一机房跨机架基于两个以上的服务做主备集群

■运维工程师需要了解服务器的真实物理连接和部署拓扑

■容灾能力：同一机房内涉及到的单机或单机柜

同城双活

■将业务部署至同一城市的两个机房中

■距离要足够远（不会被同一故障影响），又不太远（方便管理及建立起高速专用线路），50公里以内能确保网络延迟低于3ms

■两个机房间的通信链路需要高可用和高容量，需要双线，且应该基于不同的ISP链路

异地数据灾备

■数据成为业务的关键资源时，进行异地灾备，能给影响到全市范围的灾害事件后储备恢复业务的基础

两地三中心

■两个城市，三个数据中心的灾备方式，可以在城市级的灾害中，快速恢复上线业务，甚至是确保线上业务不离线

■需要从数据底层开始设计，通常的设计是数据库多写，然后互相进行数据同步

■需要对业务模块有清晰的认识，尽量降低应用请求的跨机房操作

■需要定制专门的数据同步工具，要支持数据路由和多路复制等功能

分布式多活

■理论上出现故障，系统均能通过切换流量和数据的方式，确保业务继续运行

■系统自带数据同步和数据一致性逻辑，通常会基于Raft、Paxos等协议实现，支持全自动的灾备切换