微服务

发展历程

- 响应式微服务
  - 及时响应
  - 可恢复性
  - 弹性
  - 消息驱动
- 服务网格与云原生
  - 将服务间的网络通信层及其控制策略下沉到基础设施，就形成了所谓的“服务网格 技术。通过微服务、容器化、持续交付、 Devops 等技术，组成了所谓的云原生” 体系
- 数据库网格
- 单元化架构
  - 以单元为组织架构，以单元化部署为调度单位 。
    每个单元，是一个五脏俱全的缩小版整站，它是全能的，因为部署了所有应用；但它不是全量的，因为只能操作
    一部分数据。能够单元化的系统，很容易在多机房中部署，因为可以轻易地把几个单元部署在一个机房，而把另
    外几个部署在其他机房。通过在业务入口处设置一个流量调配器，可以调整业务流量在单元之间的比例。

何时使用

- 系统复杂度高了之后考虑微服务。微服务应用在复杂度低的情况下，生产力反而比单体架构低。当在复杂度高的地方，情况恰恰相反。所以业务有一定的复杂度后方考虑。
- 大规模复杂业务系统的架构升级与中台建设

应用一般步骤

- 调研
- 分析
- 规划
- 组织

准备阶段

- 改进
- 治理
- 部署
- 拆分

实施阶段

最佳实践

- 遗留系统改造
  - 功能剥离、数据解耦
  - 自然演进、逐步拆分
  - 小步快跑、快速迭代
  - 灰度发布、谨慎试错
  - 提质量线、还技术债
- 恰当粒度拆分
  - 高内聚低耦合
  - 不同阶段拆分要点不同
- 扩展立方体
  - 水平复制：复制系统
  - 功能解耦：拆分业务
  - 数据分区：切分数据
- 自动化管理
  - 自动化测试
  - 自动化部署
  - 自动化运维

降低服务拆分带来的复杂性
提升测试、部署、运维效率

- 分布式事务
  - 幂等/ /去重/ 补偿。慎用分布式事务
- 完善监控体系
  - 业务监控
  - 系统监控
  - 容量规划
  - 报警预警
  - 运维流程
  - 故障处理

稳定性建设