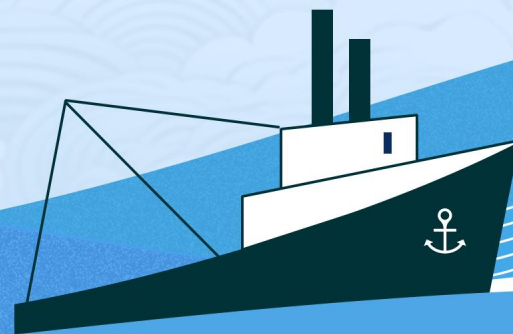


Kubernetes跨站点容灾 技术方案与最佳实践

骥步科技CEO 刘洋

骥步科技总架构师 龚永杰



Content 目录

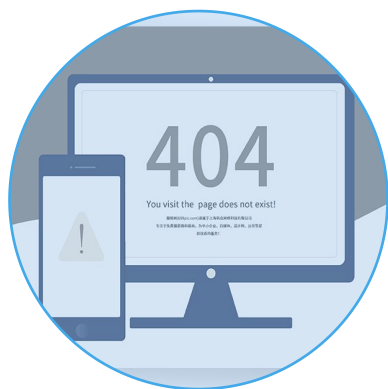
- 01** Kubernetes跨站点容灾方案的挑战与设计思路
- 02** Kubernetes跨站点容灾方案的常见技术方案
- 03** Kubernetes跨站点容灾方案的最佳实践和案例分享

Part 01

Kubernetes跨站点容灾方案的挑战与设计思路

影响Kubernetes集群可用性的因素

- Kubernetes Failure Stories: <https://k8s.af/>



软件故障



人为失误



机房事故

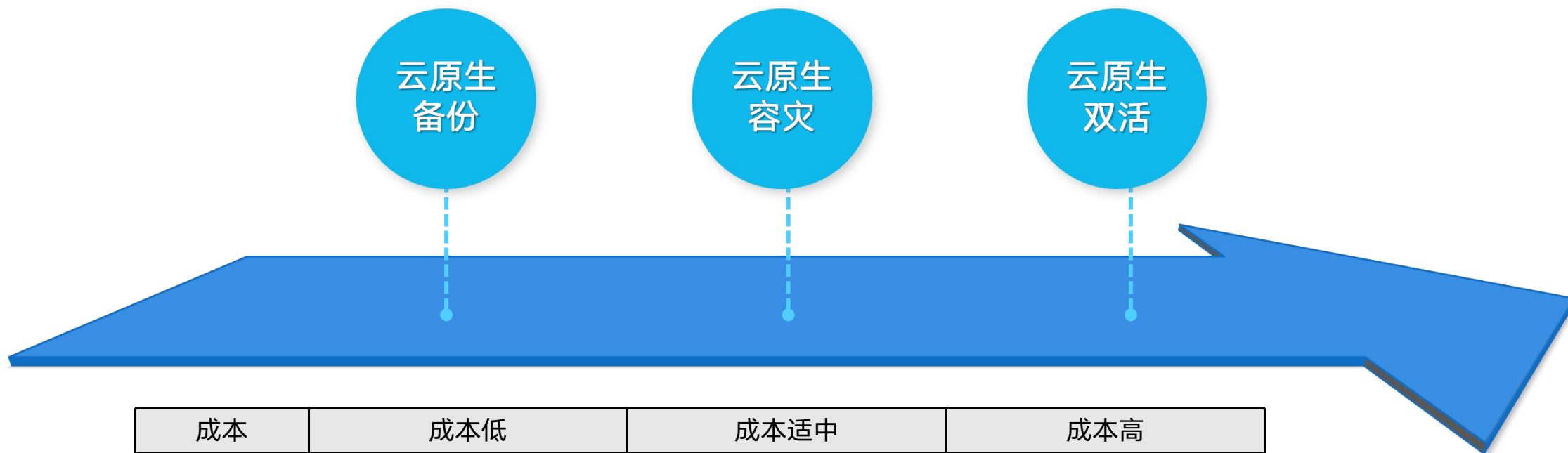


自然灾害



网络中断

Kubernetes集群业务连续性建设路线图



成本	成本低	成本适中	成本高
RPO	小时级	0, 秒级-分钟级	0, 或接近0
RTO	小时级	秒级-分钟级	0, 或接近0
方案复杂度	低, 通用方案	中, 通用方案	高, 针对应用设计

Kubernetes容灾方案的挑战

1

以应用为中心的容灾取代以机器为中心的容灾

2

容灾流程需要协调多种资源的同步和切换

3

需要能检测和处理各种复杂的异常场景

4

需要支持容灾站点的异构基础设施和K8s版本



容器容灾方案设计原则

01

以应用为中心

支持完整的应用资源定义，可以进行应用、应用组和集群级别的端到端切换和恢复

03

自动化

自动捕获应用发布、版本更新和配置变更，自动在站点间同步应用资源 and 数据

02

流程化

自动管理切换流程中的要素依赖和顺序，确保容灾切换的及时性和准确性

04

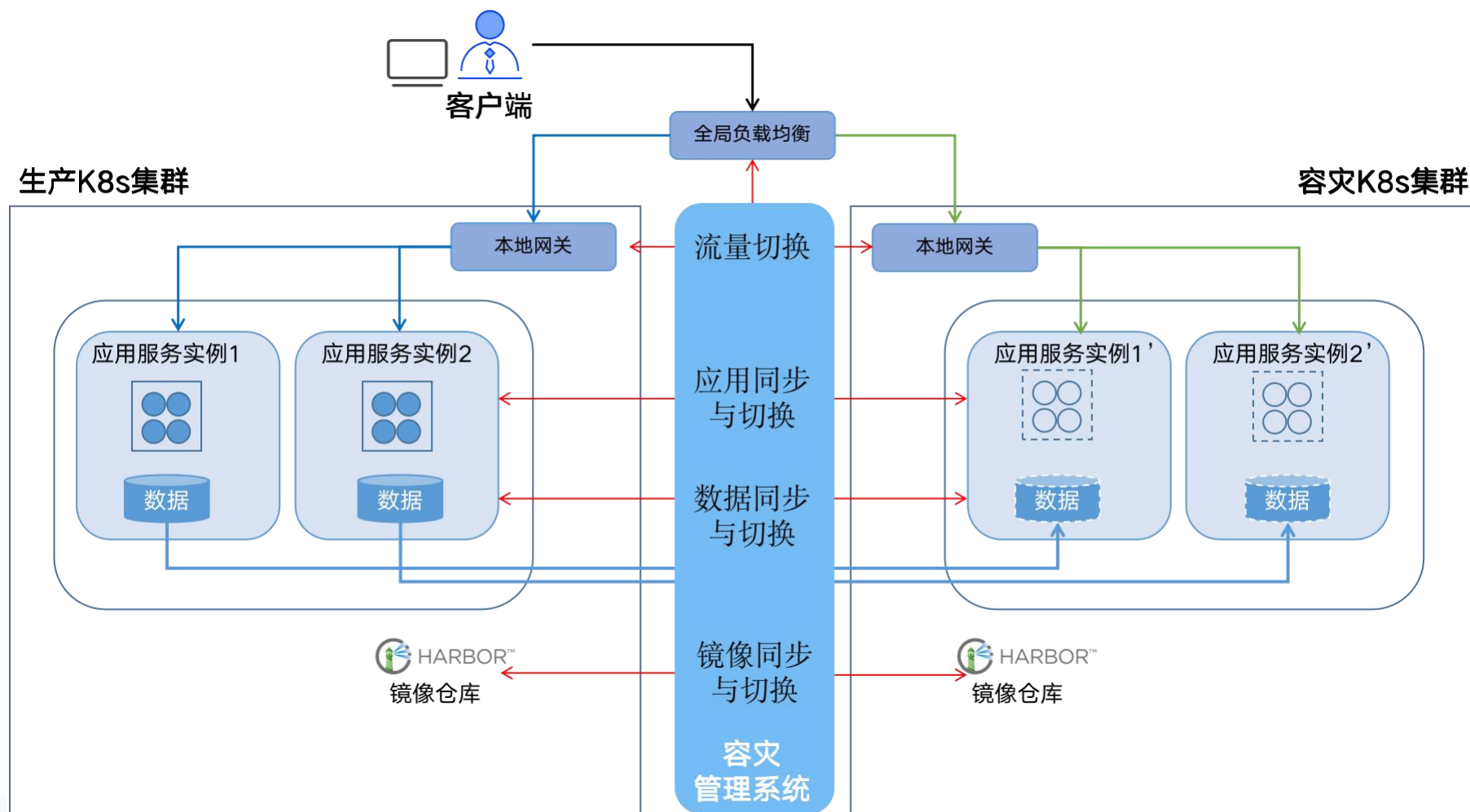
标准化

尽可能基于K8s的标准接口和社区的最佳实践设计方案，增强适配性

Part 02

Kubernetes跨站点容灾方案的常见技术方案

资源和数据同步设计



应用同步方案

- ✓ 多集群发布
- ✓ Velero

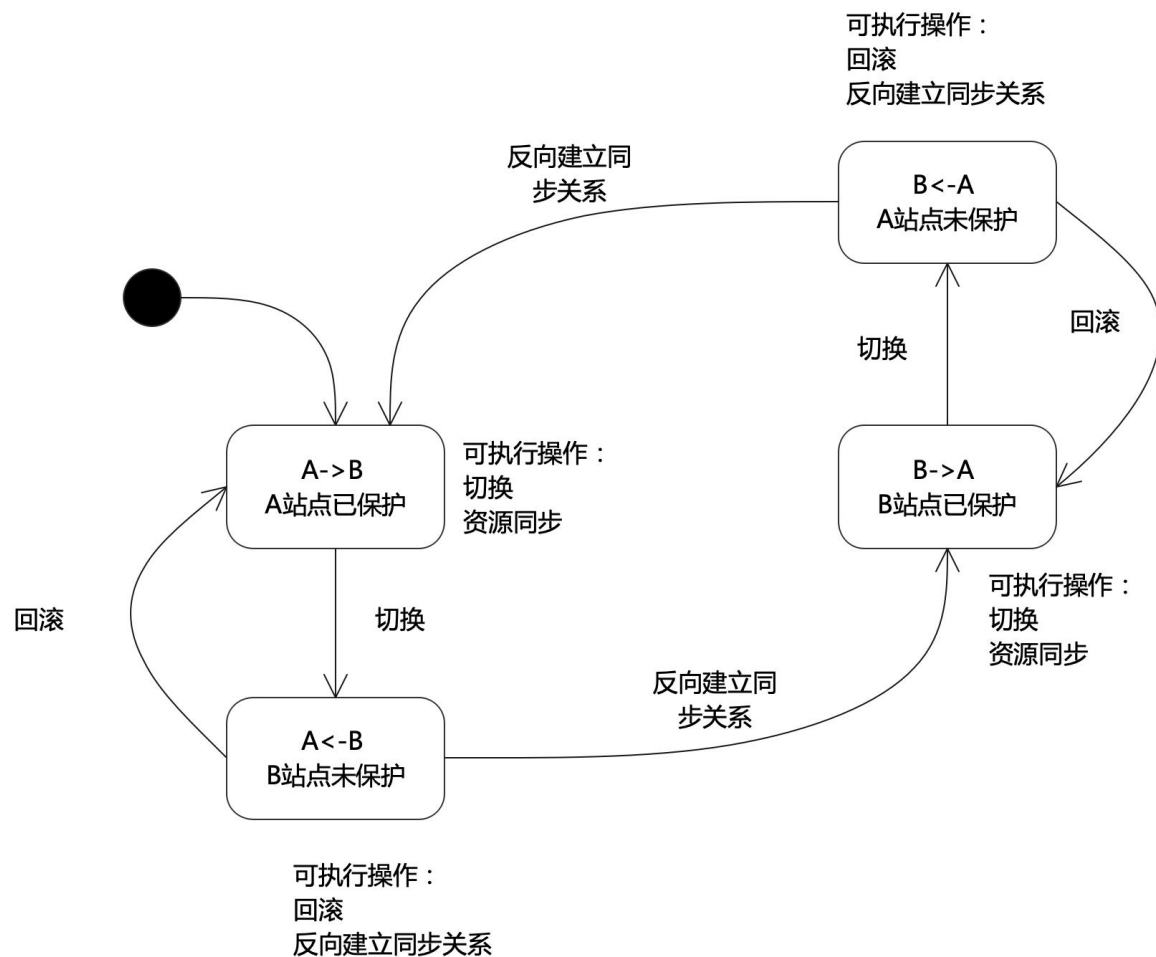
数据卷同步方案

- ✓ restic
- ✓ rsync/rclone
- ✓ 存储复制

镜像同步方案

- ✓ Harbor replication
- ✓ 手工pull/push

站点切换状态机设计



• 规范切换场景和流程

- ✓ 计划内切换
- ✓ 故障切换
- ✓ 容灾演练

• 规范切换动作

- ✓ Failover/切换
- ✓ Reverse/反向同步
- ✓ Restore/回滚

Part 03

Kubernetes跨站点容灾方案的最佳实践和案例分享

容灾方案RTO优化最佳实践

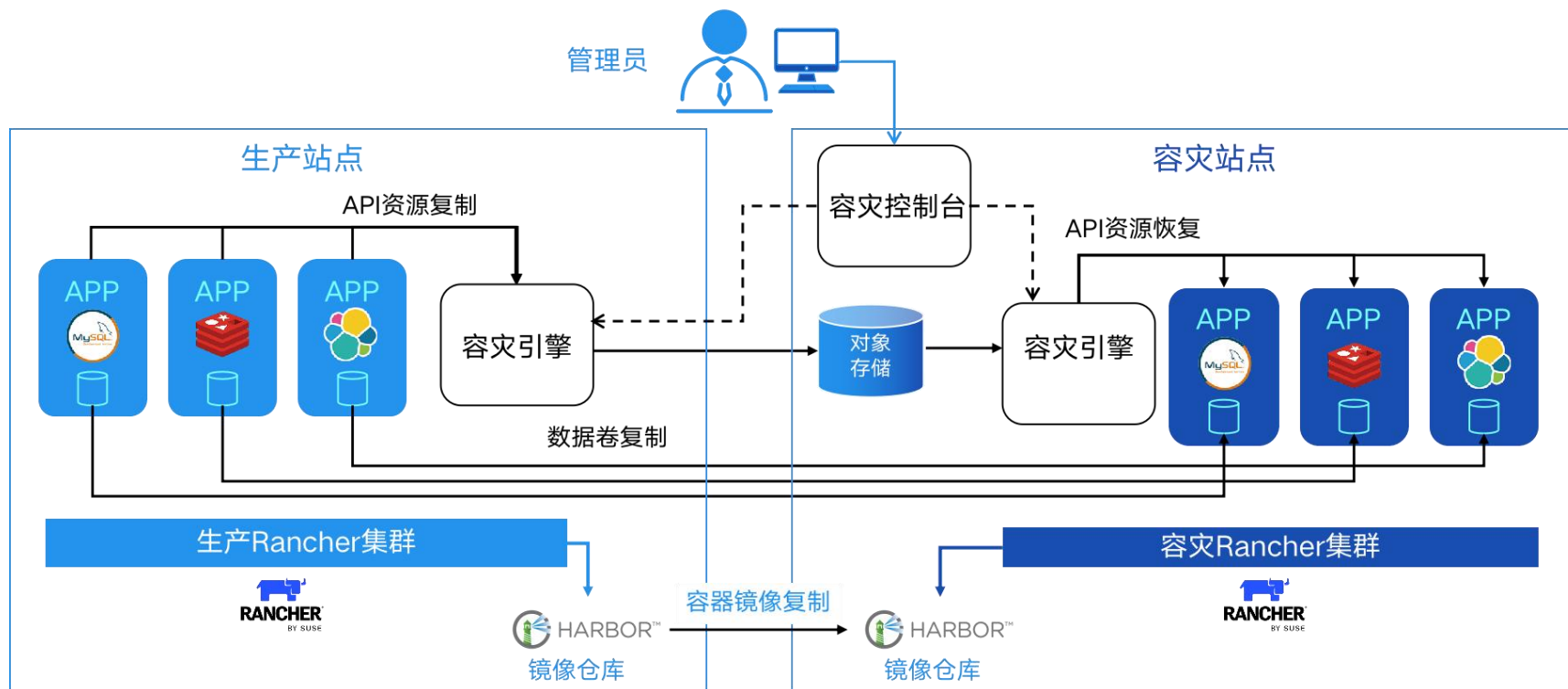
- 容灾站点应用准备
 - ✓ 实现应用部署但不拉起
- 镜像预拉取
 - ✓ 在目标集群节点上预先进行关键镜像拉取
- 数据卷准备优化
 - ✓ 优先推荐数据卷复制的方式，保持数据卷数据ready状态
 - ✓ 如果采用备份恢复的方式，需要实现数据卷在容灾站点的持续恢复

RPO优化

- 选择最佳的数据同步策略

数据同步方案	适用场景	方案限制	方案RPO
共享/双活存储	容灾站点容器应用和主站点容器应用共享同样的存储设备，或存储本身支持在两个站点间的双活访问，常见方案如共享NFS，分布式文件系统等。	双站点共享同一存储设备存在单点故障风险。双活存储一般成本较高，对存储网络要求高。	0
存储同步/异步 数据卷复制	容灾站点和主站点部署了同构的存储产品，可以直接利用存储本身同步/异步的数据卷复制能力实现PVC同步。	需要两站点采用同样的存储设备/型号，一般需要专有的存储网络保障数据同步性能。	0到分钟级
文件复制引擎 (rsync /rclone 等)	主从站点的存储是异构配置，无法直接进行数据复制，用户对RPO有一定容忍度。	大量小文件复制效率较低，RPO难以保证。rsync要求直接网络连接。	分钟级
备份恢复	通过持续备份主集群的数据卷，并持续恢复到从集群进行数据同步。对存储本身无要求，主从集群之间可以无直接网络连接。	数据复制性能不够好。	分钟级到小时级

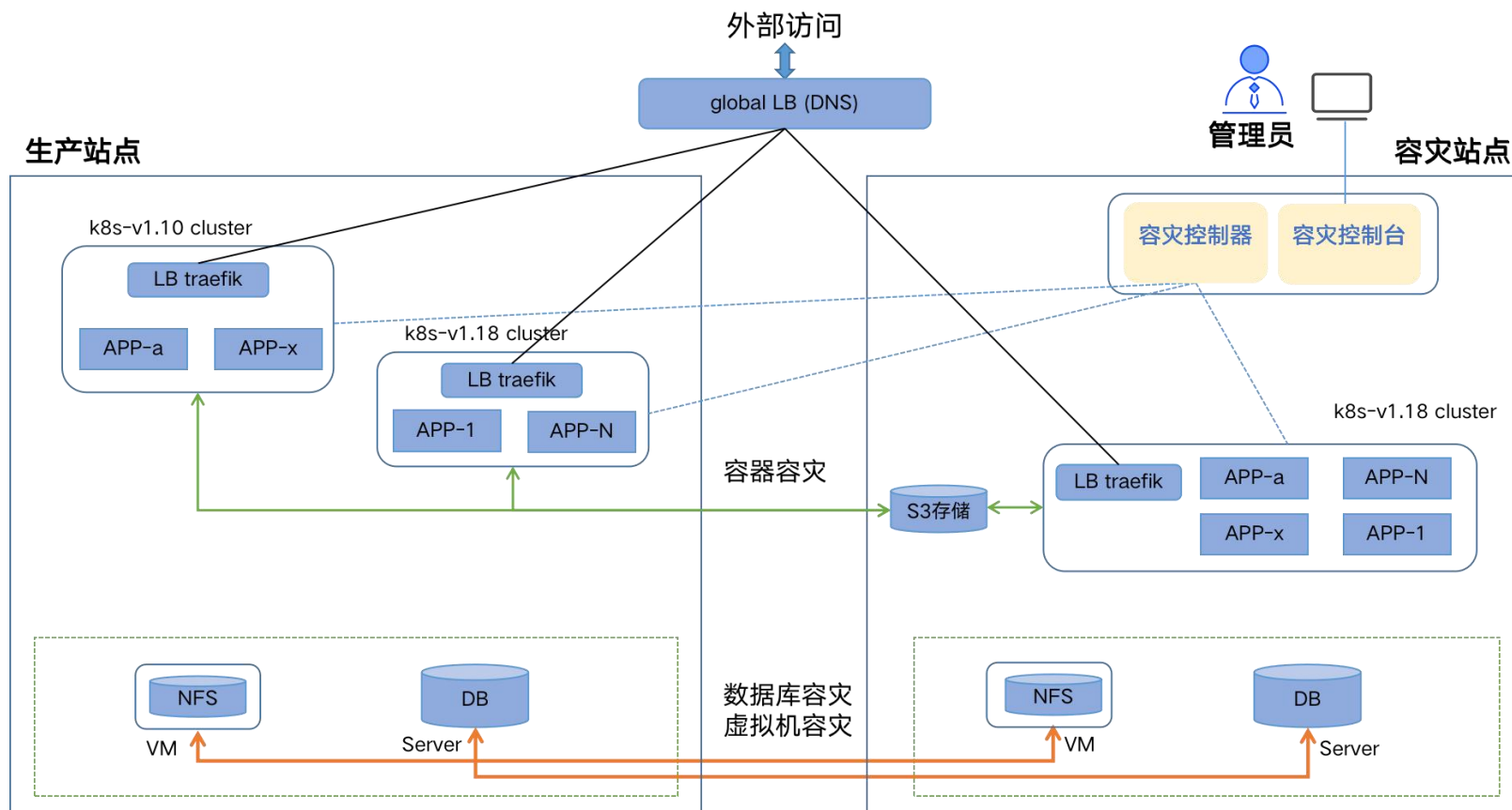
案例分享：某保险公司容器集群同城高可用方案



方案设计亮点

在不改变用户现有的应用发布流程情况下，实现用户容器应用和中间件的跨站点同步及端到端容灾切换能力，利用两个站点间商业化存储的异步复制能力，满足客户分钟级RPO和RTO要求。

案例分享：某生信公司容器集群同城容灾方案



方案设计亮点

为用户提供不同K8s版本生产集群的容灾切换管理能力，方便用户实施后续的滚动升级。与外部数据库和虚拟机容灾方案配合，实现分钟级RTO的容灾要求。

Thanks.

