Kubernetes Community Days

为云原生关键工作负载保驾护航
——Velero备份容灾最佳实践

刘洋 （liuyang@jibudata.com）
龚永杰（gongyongjie@jibudata.com）

关于我们

Kubernetes Community Days

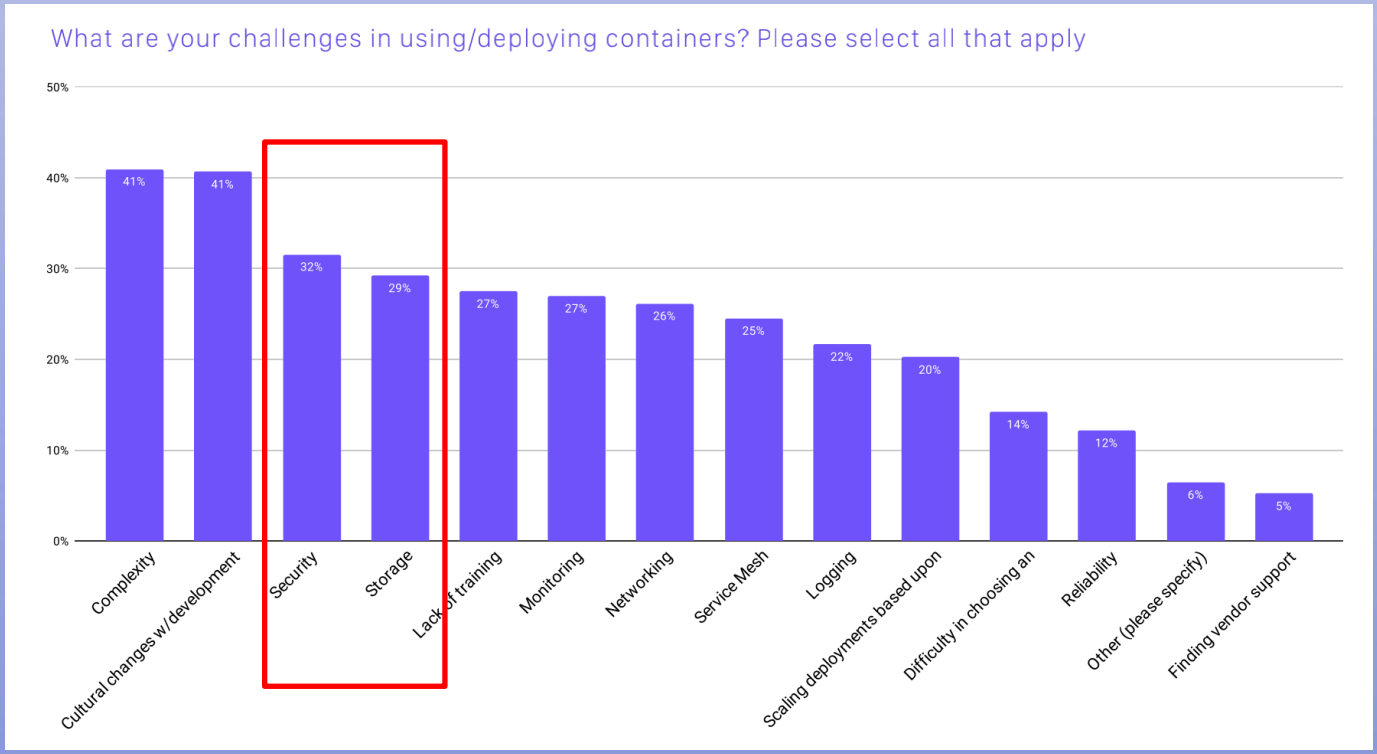
刘洋
CEO
骥步科技

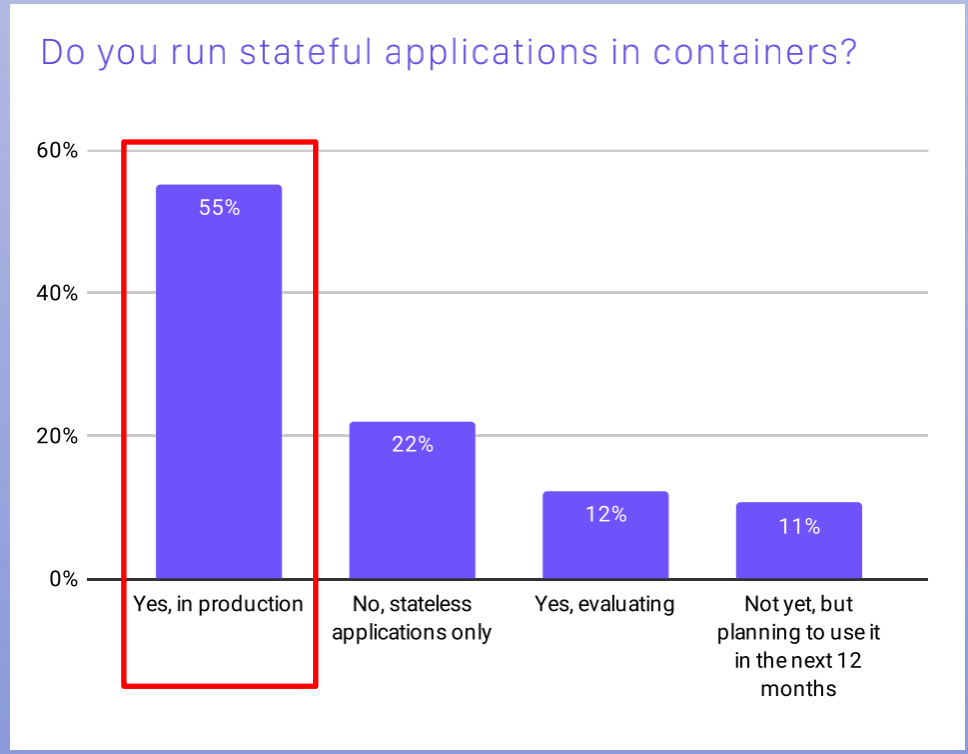
龚永杰
总架构师
骥步科技

云原生数据安全日益重要

Kubernetes Community Days

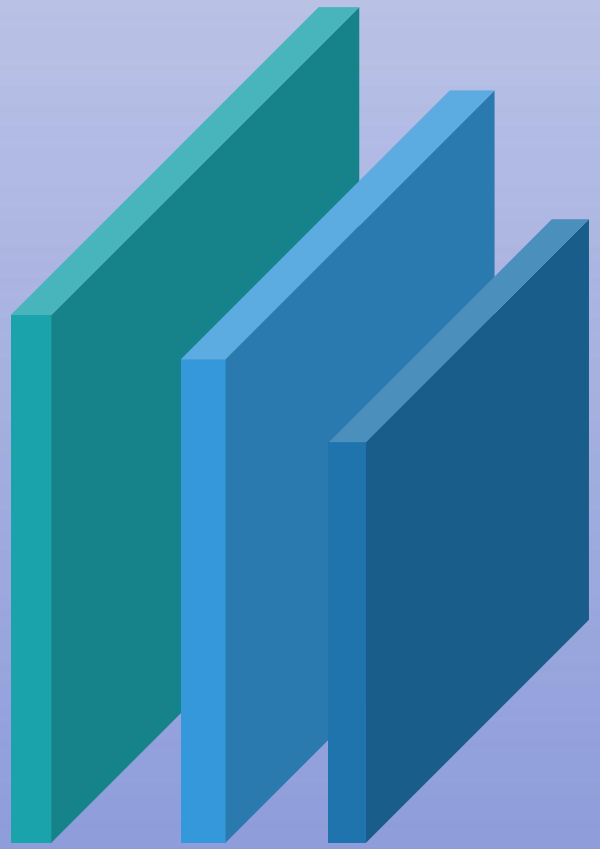
- 等保2.0、数据安全法等国家政策法规进一步加强了数据保护的要求，明确了企业责任
- 云上安全事件的频度和影响日益加深
- 传统的灾备方案无法很好的工作于容器环境，企业需要全新的云原生数据保护解决方案





云原生数据保护的技术需求

Kubernetes Community Days



- **以应用为中心**
  - 数据管理和保护的目标和粒度是应用而不是机器
  - 屏蔽底层基础设施的异构和复杂性
  - 实现通用的应用数据一致性
- **与Kubernetes生态集成**
  - 以云原生的方式部署和使用
  - 与DevOps和微服务最佳实践深度结合
  - 自动化
- **生而多云**
  - 支持多种异构的基础设施
  - 提供跨多云的迁移和灾备能力
  - 开放数据标准，无厂商锁定

云原生数据保护常见场景

Kubernetes Community Days

生产安全

- ✓ 备份与恢复
- ✓ 远程容灾

降本增效

- ✓ 跨集群迁移

业务价值

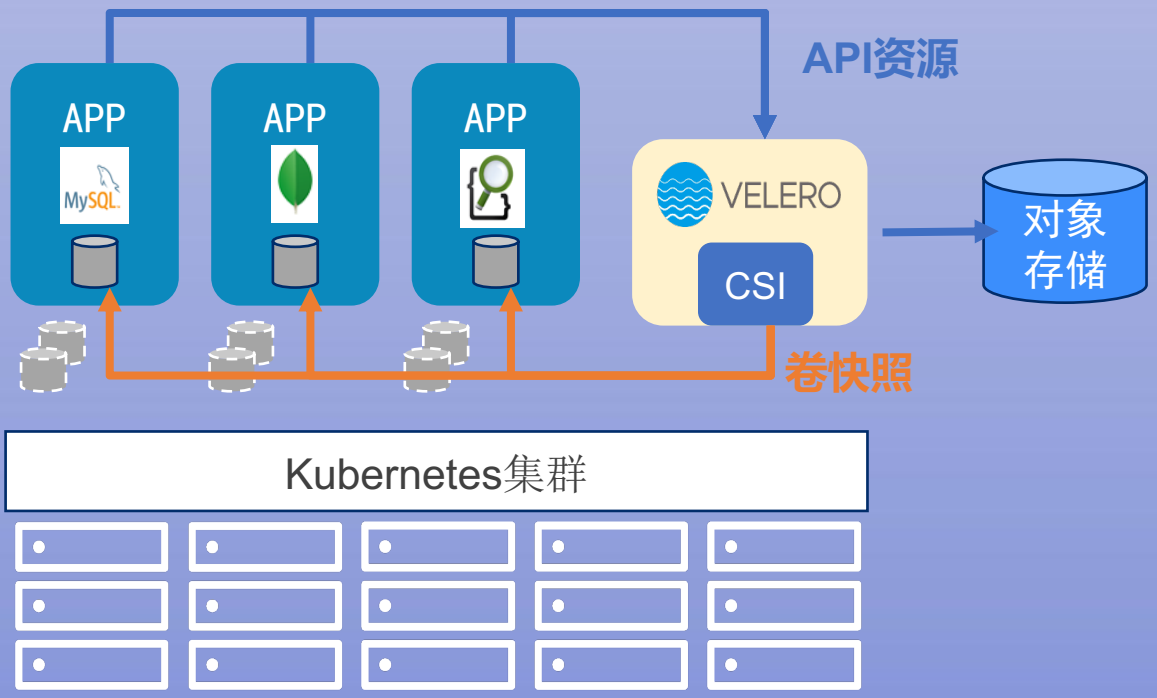
- ✓ 快速搭建开发测试环境
- ✓ 数据重用

使用Velero备份多种应用

Kubernetes Community Days

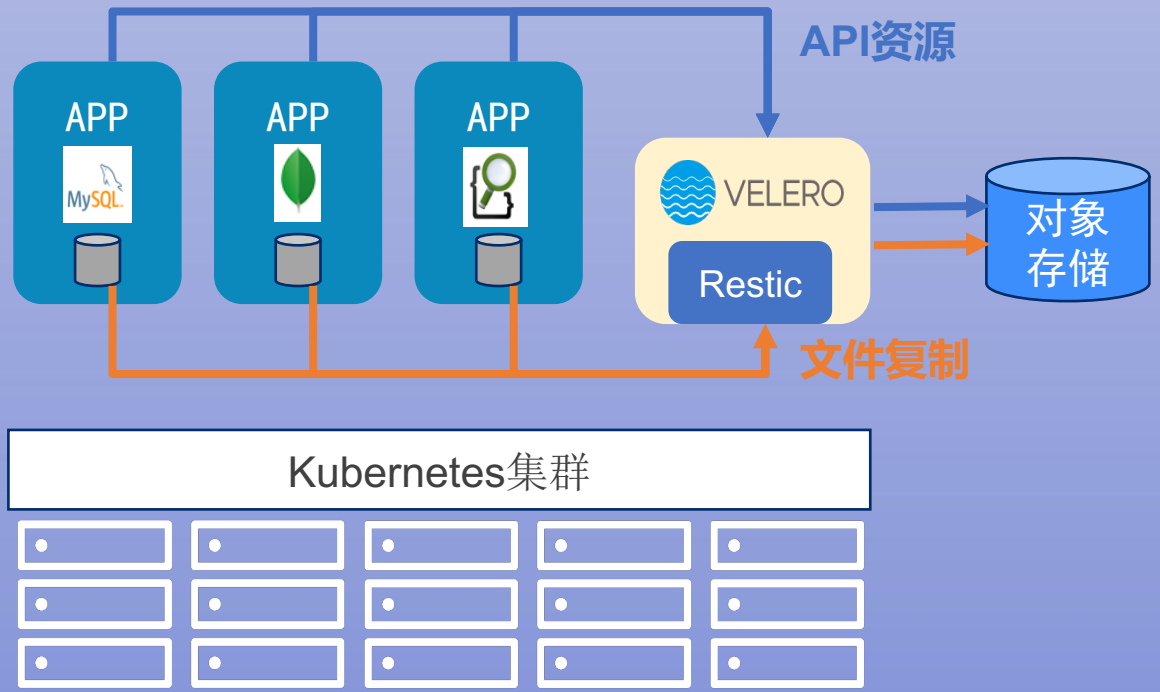
快照方式

- 存储须支持CSI快照接口
- 对象存储只存储API对象



文件复制方式

- 支持没有快照功能的存储
- 对象存储同时存储API对象和PVC数据

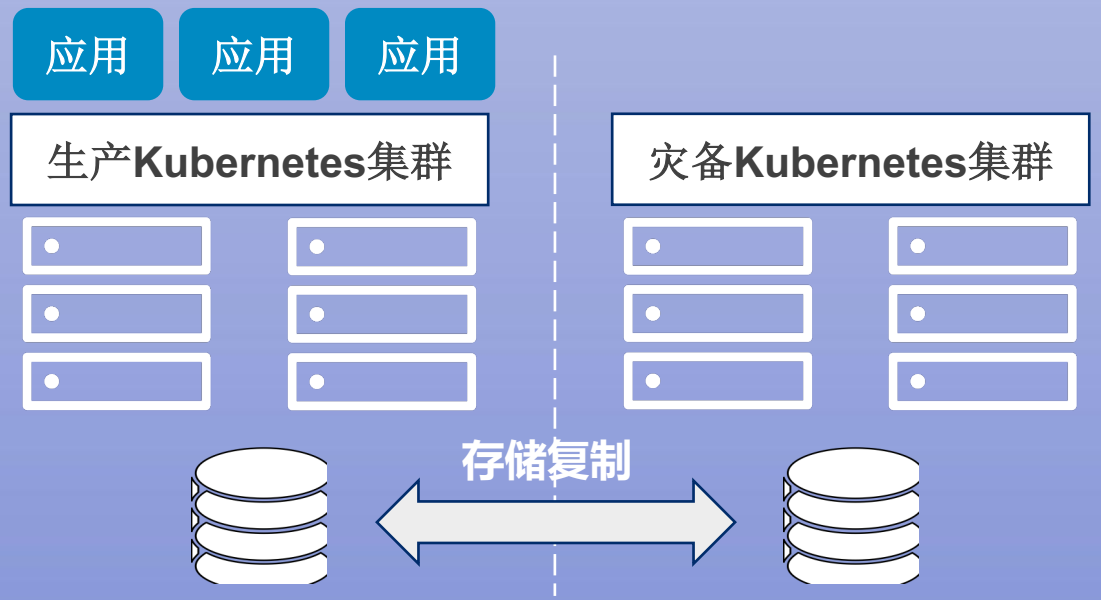


基于Velero构建低成本云容灾

Kubernetes Community Days

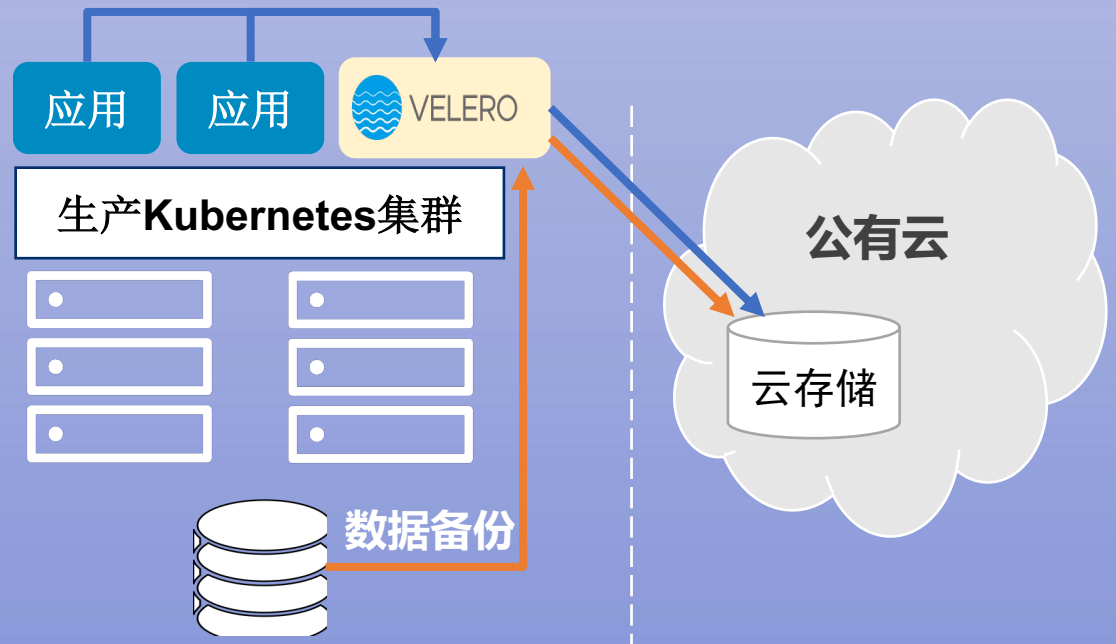
传统容灾方式

- 灾备站点需建设同等基础设施
- 基于存储复制，两端存储需同构
- RPO和RTO好



云容灾方式

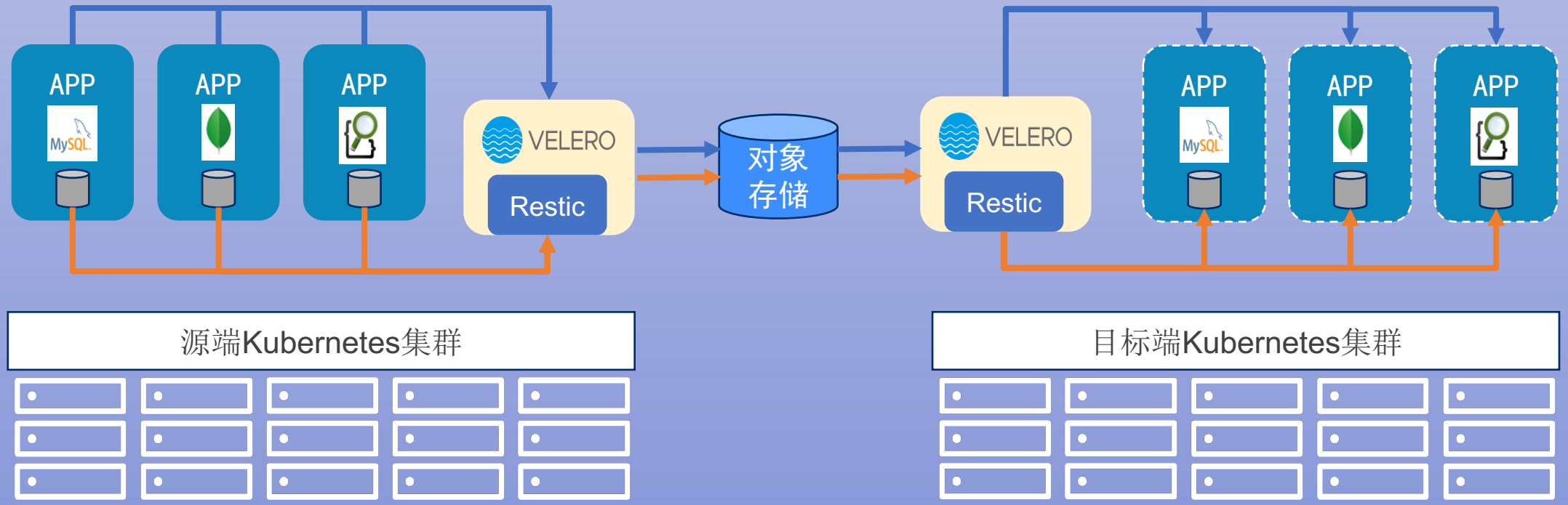
- 云端资源可按需拉起
- 两端基础设施可异构
- RPO和RTO有局限性



使用Velero进行跨集群迁移

Kubernetes Community Days

- 集群大版本升级
- 更换基础设施
- 负载均衡



Velero最佳实践分享

Kubernetes Community Days

01 1. Velero IO path

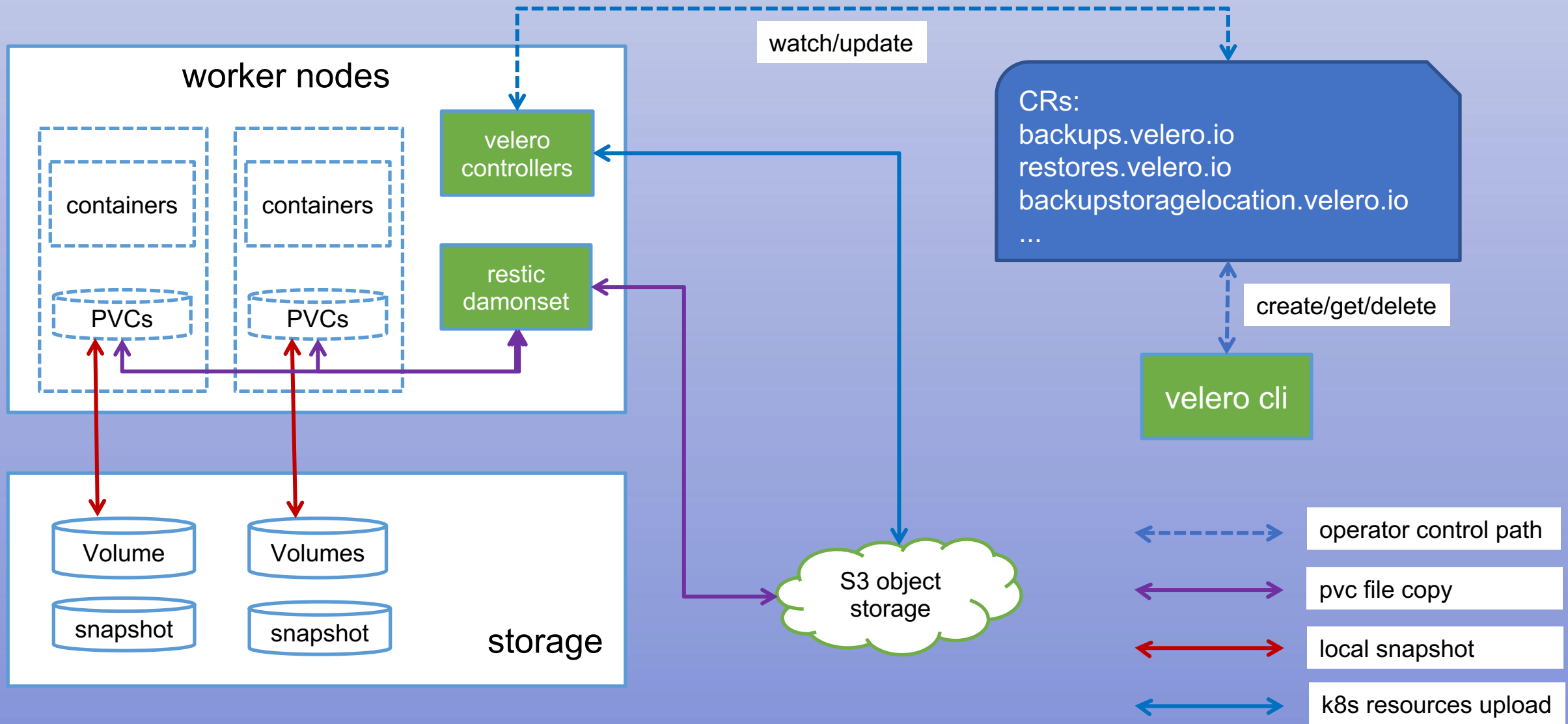
02 2. 快照 V.S Restic

03 3. Velero的最佳实践

04 4. 常见问题和解决方法

05 5. Velero中文社区

Velero IO path



快照 V.S Restic

Kubernetes Community Days

考量维度	基于存储的本地快照	基于Restic 的文件复制
应用性能影响	低，通过CSI接口调用存储系统的快照实现	取决于数据量大小，占用额外CPU, Network和Storage 资源从节点上复制数据上传至对象存储
数据可用性	依赖于存储系统的本地高可用性	对象存储跟生产环境隔离，具有独立可用性，多数对象存储方案支持跨站点高可用
数据一致性	支持 Crash Consistency，配合hook机制可实现应用数据一致性	无保证，需要配合hook机制来实现应用数据一致性
执行速度	快，秒级	慢，速度取决于本地IO能力，网络带宽和对象存储性能
环境限制	底层存储需要支持CSI 快照接口，本身具备快照能力，比如Ceph或者传统中高端块存储设备	对底层存储无依赖，支持多数PVC（但hostpath 和projected volumes除外）

Velero的最佳实践 - 1

Kubernetes Community Days

在生产环境中，结合高频度本地快照备份和低频度restic备份到对象存储

- 发挥本地快照的优势
  满足crash consistency一致性要求，速度快，对应用性能影响小

- 提高数据可靠性
  同时在本地和远端S3对象存储中都保存可恢复数据

- 恢复灵活性
  通过对象存储中的备份数据，可将应用恢复至其他集群

注意: 当前velero需要创建单独的备份任务分别做快照和Restic备份

Velero的最佳实践 - 2

Kubernetes Community Days

从应用角度选择合适的备份粒度和备份策略

- 定义针对不同应用的备份策略：备份频率，保留时长

- 定义较小粒度应用模版，方便灵活指定需要备份的K8S资源

- 避免在一个任务中包含大量PVC 数据使得备份任务执行过长

注意: 当前velero 备份执行为串行方式，多个备份任务会排队执行
优化方案：部署2个velero实例分别执行短时间备份任务和长时间备份任务

Velero的最佳实践 - 3

Kubernetes Community Days

多集群环境中共享同一对象存储时要防止冲突

场景：集中管理多集群备份，方便应用和数据在多集群中迁移

- 在每个K8S集群中创建BSL指向同一个S3 bucket时，BSL中需要配置prefix参数，区别不同集群，比如使用cluster name作为prefix

- 在迁移场景中，两边集群需要共享同一个BSL，目标集群需要修改访问模式为只读

注意：Velero在备份和定期清理过期数据的过程中会lock S3，上述配置可避免因为lock造成的日常备份任务失败

常见问题和解决方法 - 1

Kubernetes Community Days

1. 删除长时间未完成的备份或者恢复任务，会导致velero阻塞无法处理后续任务

阻塞原因：

1) 任务在未指定超时时长条件下，默认为4小时
2) 当前任务除非失败或者完成，否则状态会保存在velero内存中
3) velero controller 当前采用串行模式执行任务

任务长时间无响应可能的原因：

1) 数据量过大
2) 底层存储性能问题造成过长时间没有完成数据拷贝
3) velero plugin start timeout

目前 velero暂不支持 abort 执行过程中的任务
解决方法：删除正在执行的任务CR，重启velero pod，排查原因，创建新的备份或者恢复任务重新执行

常见问题和解决方法 - 2

Kubernetes Community Days

2. 排查应用恢复失败的问题

恢复到源集群典型失败原因：

- 资源冲突，比如 ingress 或者 node port 已被使用，无法创建资源
- 资源不足，当前集群没有额外的CPU/Memory资源

恢复/迁移到其他集群典型失败原因：

- 依赖的环境资源没有准备好

- 不同k8s API 版本兼容性问题
=> 安装velero的时候打开 --features=EnableAPIGroupVersions

- 没有对应的 storage class
=> https://velero.io/docs/v1.6/restore-reference/#changing-pvpvc-storage-classes

其他排查方法: 查看velero pod log，搜索关键字 "error restoring”

常见问题和解决方法 - 3

Kubernetes Community Days

3. 使用restic备份PVC的相关问题

1) restic无法备份未被挂载的PVC

分析: Velero 将整个'/var/lib/kubelet/pods' 目录通过hostpath 挂载到restic daemonset，之后访问特定PVC目录来拷贝数据； 因此无法访问未被挂载的PVC 数据。

解决方法：将 PVC 挂载到一个“sleep” pod 上再进行备份

2) 仅备份一个pod中的部分PVCs

场景：对于临时数据的PVC，比如暂存应用日志，无需进行数据备份

解决方法：在对应pod上添加 annotation: backup.velero.io/backup-volumes-excludes=volume1,volume2, ...

常见问题和解决方法 - 4

Kubernetes Community Days

4. 应用的custom resource status 无法恢复

分析：从Velero 相关issue的讨论，当前设计理念认为 CR status 应该从CR spec重新构建出来，status 恢复为非原子操作，存在并发操作的问题，因此在恢复时会将CR status清除

导致的问题：
恢复后会触发某些job类型CR的重新执行；某些CR状态丢失造成应用状态不一致

当前解决方法：
修改应用operator逻辑，利用annotation同步保存status状态；在恢复后，operator先尝试从annotation中恢复 status

Velero中文社区 - velero.cn

Kubernetes Community Days

Kubernetes
Community Days

谢谢！
Thank you !

KubeCon China 2021:

protect your database workloads in kubernetes - by Jibu Tech