# 多集群互备与异地还原

## 背景

鉴于目前的生产备份形势，在可见的现在，可预见的未来，数据量会有较大的存量和应对较大的增量。

比如，仅招投标生产环境的minio 已有 118G的数据，且小文件居多，平常的压缩备份手段很难再使用

除此之外，数据的种类也较多，含minio redis mysql mongo kafka 等，种类繁杂，备份手段不统一。

且数据会进一步增长，比如实验室项目，仅生产目前已知将要上传的数据200多G，并会继续上传

## 应对策略

大规模容器编排场景下，数据存储需要依赖外部的专用文件存储系统，或者存储服务。目前使用的为longhorn。

Kubenetes本身在1.12之后引入了 VolumeSnapshot （卷快照）的概念，但在之后由于存在In-tree 和 Out-of-tree 的争论，VolumeSnapshot的实用性一直在演进中，1.17之前一直是alpha , 1.17之后才进入beta版本，1.20之后才开始GA。

鉴于市面上的大多数生产K8S 版本，都小于1.20 （Kubernetes 1.20的CRI （容器运行时接口）“抛弃” 了docker） ，但目前，开发人员使用，以及绝大多数的已有环境都在运行 docker，这一巨大变动，可能约束了我们并不能大动作的变动kubenetes版本

而且，VolumeSnapshot服务并非kubernete“自带”服务，其本身需要存储厂商较好的支持，才可以实现 CSI （容器存储接口） 级别的 snapshot

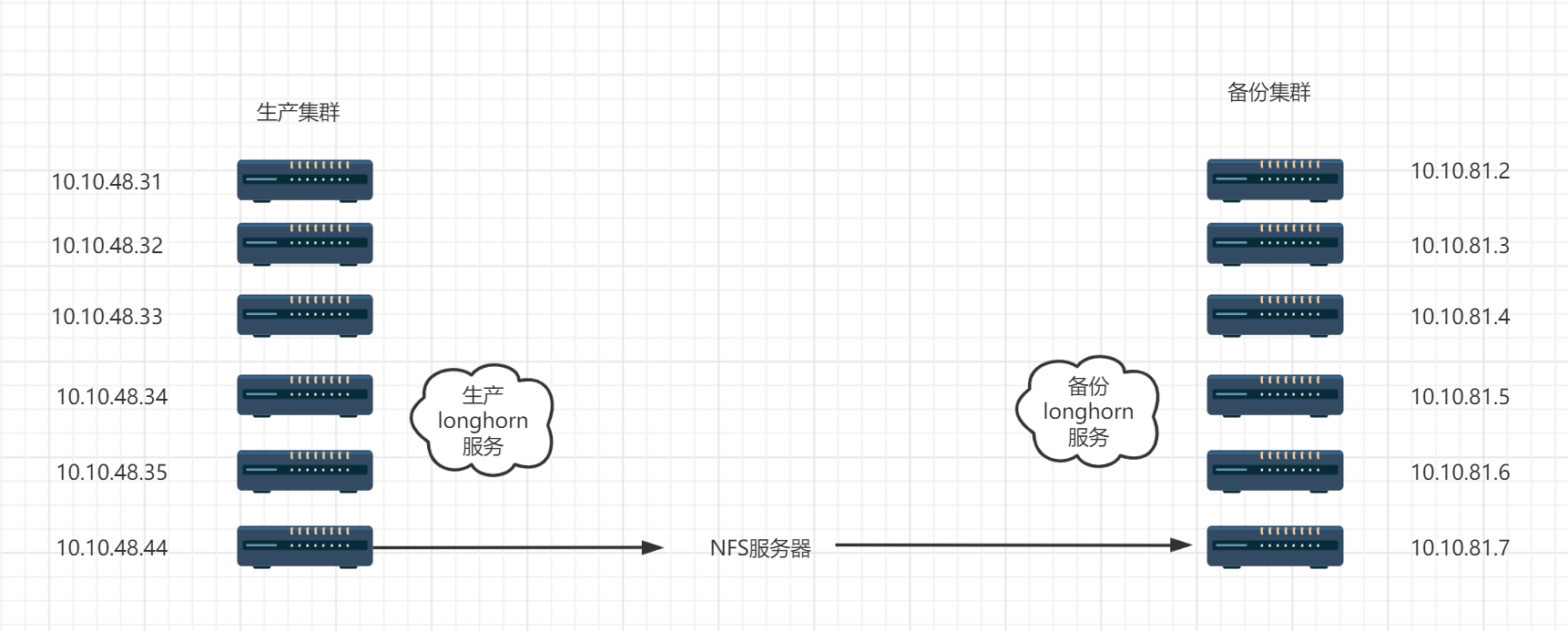
在实践中发现，longhorn 在UI 界面的 snapshot 和 Backup 中，snapshot并非CSI 层面的 VolumeSnapshot 概念，并不能作为 VolumeSnapshot直接做备份和使用。而是该存储内部的一种快照实现。而且longhorn本身做为一种CSI，也不支持 CSI snapshot，测试中均为实现。

（一直卡在创建中）





## 架构示意



## 实现目的

**对于任意的存储块，PV/PVC，可以指定任意的备份策略到远端地址，并可以在远端进行数据的还原**

鉴于此，使用longhorn系统自带的backup 系统完成备份系统的搭建

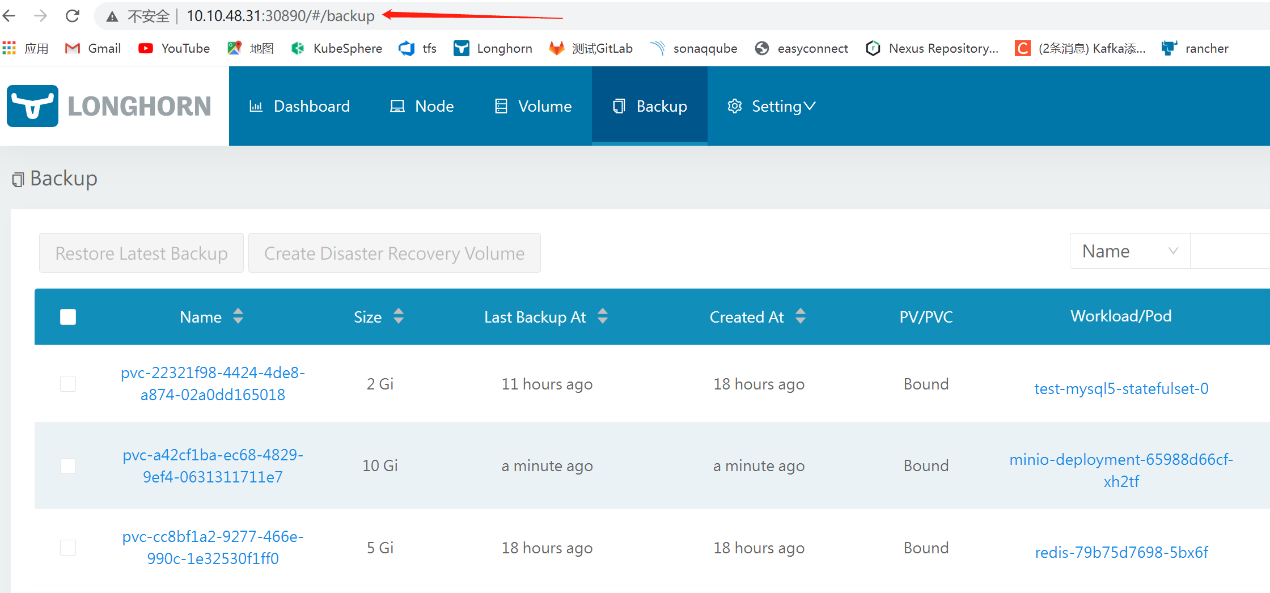
在后端存放备份文件系统的选型中，有minio 和 NFS，考虑使用NFS

流程为：

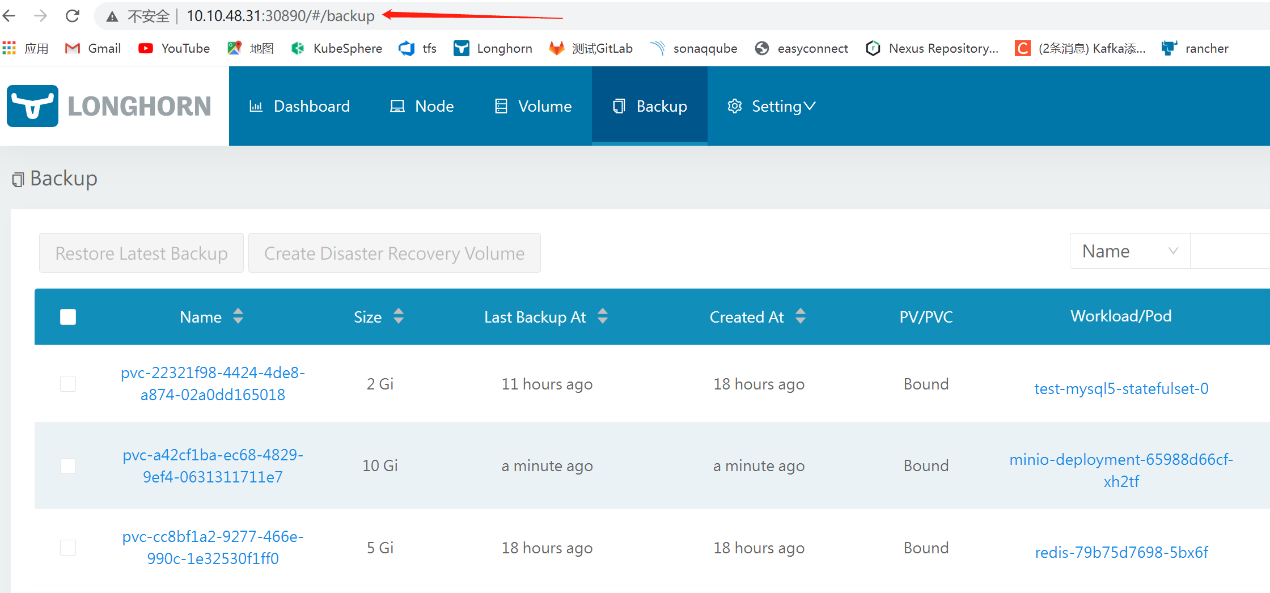
生产集群 10.10.48.0/24 🡪 10.10.48.44 NFS 🡪 10.10.81.0/24 备份集群

备份测试结果展示：

主集群备份文件展示



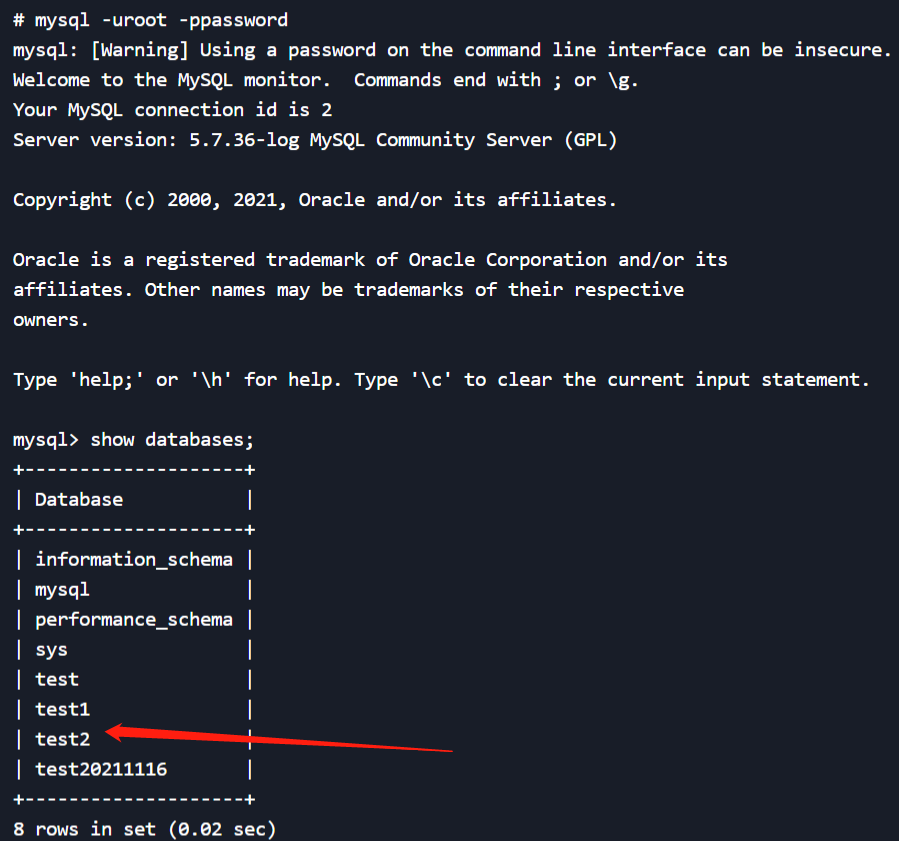
备用集群备份文件展示



可以发现，两个集群拥有相同的备份，这样还可以实现在备份环境中实现数据恢复

利用备份环境还原数据例子如下：

数据得到保存且可以还原



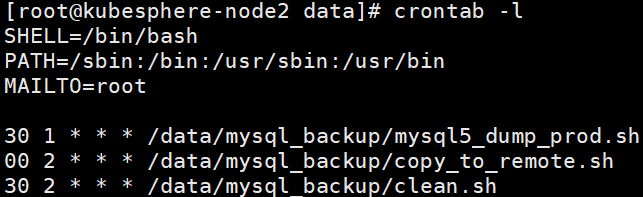
## 备份原则

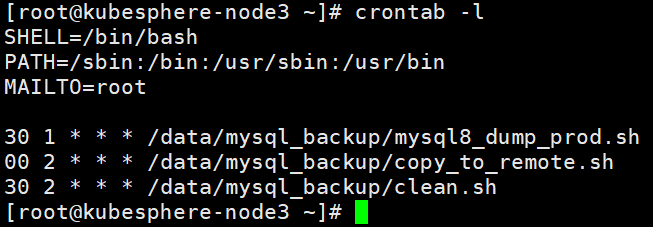
* 备份必要数据。目前只备份生产数据
* 逻辑备份为主。对于可以进行逻辑备份数据，如MySQL库数据，优先考虑如MySQLdump，Xtrabackup等专业备份工具，对于其他数据，如minio存储，redis, mongo等采取底层备份
* 异机，异集群存放数据。
* 异机，异集群可恢复数据。
* 备份粒度为天，结合生产环境开启bin\_log, 粒度内数据安全性，如果需要恢复数据，以bin\_log为主。
* 暂定备份2周

## 目前逻辑备份情况

生产情况的数据库，分为MySQL 5 和 MySQL 8 两个版本

目前采用mysqldump工具，分布在10.10.48.35 node2（mysqldump 5版本）和10.10.48.44 node3 (mysql 8版本）,结合crontab,做每天的备份





以8为例，其中

mysql8\_dump\_prod.sh 代表备份生产数据库，如果有新加生产环境，按脚本内的内容添加即可。

copy\_to\_remote.sh 会把每日的备份数据，备份到远程机器上，远端机器IP为 10.10.58.81

clean.sh 会定期清理14天以外的备份文件，保留14

总体逻辑备份目前保持上述结构，具体一些修改，可以修改上述具体文件