KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2019

Plan to Fail:

A *good* Captain Doesn’t Sail Without Life Rafts

Carlisia Campos, Steven Wong
VMware

Abstract

Hidden slide during presentation – included for those finding deck online later

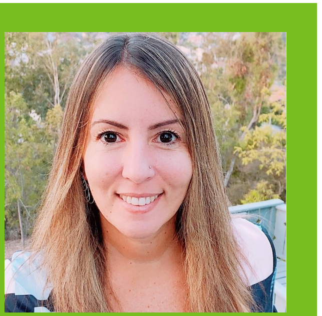
Historically, formal disaster recovery (DR) plans were only feasible for large enterprises. They could afford to allocate time, resources and the cost of duplicating a datacenter infrastructure.

With the popularity of public cloud and cloud native technologies, the cost and complexity of DR planning has been significantly reduced. This means every company, large and small, can engage in business continuity planning. Why is this important? These are some of the reasons:

Machines and software fail
- People make mistakes
- Hackers prey on the vulnerable
- Weather, fire, terrorism, more...
- You lose customers when there are outages and data loss
- Legal standards often require data retention

This talk will focus on:
- Items that need to be backup and why - some might surprise you
- Why you need selective restore capability
- Existing tooling to simplify and automate a DR strategy

Presenters



Carlisia Campos

San Diego

Senior member of Technical Staff, VMware

Carlisia is a maintainer of the open source project Velero, a cloud native disaster recovery and data migration tool for Kubernetes workloads.

GitHub: @carlisia

Steven Wong

Los Angeles

Open Source Community Relations Engineer, VMware

Active in Kubernetes community since 2015 – storage, IoT+Edge, running K8s on VMware infrastructure. Former engineer and architect of Avamar and other backup products.

GitHub: @cantbewong

Agenda

Why you need a recovery plan

Elements of a DR plan

Items you need to backup and why

Existing tools to implement backups and recovery

Demo

Why do you need a recovery plan?

Machines and software fail

People make mistakes

Hackers prey on the vulnerable

Weather, fire, terrorism, crime, earthquake, more…

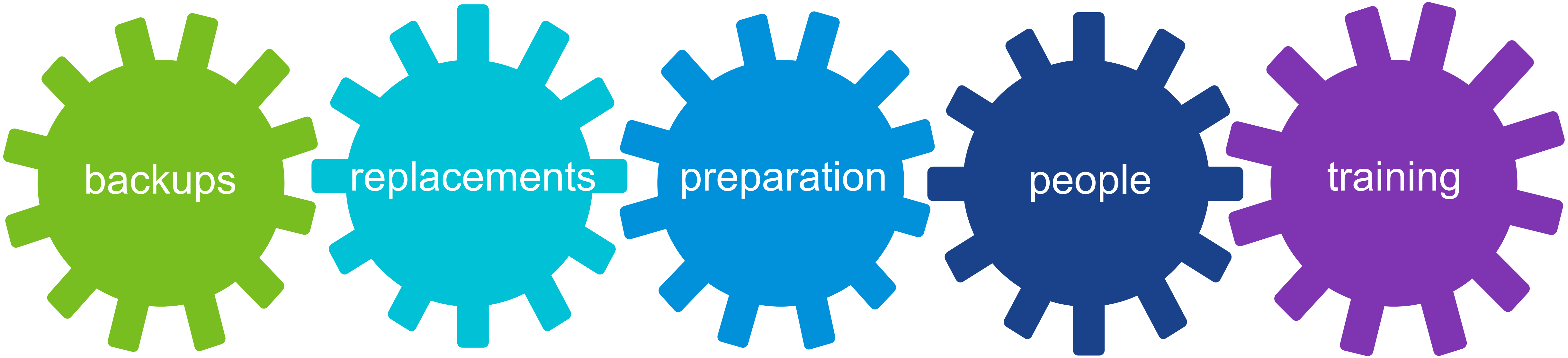
Customers have alternatives - *Customer retention is costly, but customer re-acquisition is devastatingly expensive*

Legal standards often require data retention and impose a duty of reasonable care to protect against physical and financial harm

Photo by chuttersnap on Unsplash

Elements of a Disaster Recovery Plan

All these work together



possibly replicated to alternate sites

hardware or available cloud capacity

A *planned* process

people who can carry out the process

recurring training, recovery tests

Not enough to just make a plan – record it as a living document

Runbooks documenting recovery procedures should be tested and retained offline –

pre-installed on tablets or even printed

What to protect

Critical components in a Kubernetes cluster

All *native* Kubernetes objects are stored on etcd. Periodically backing up the etcd cluster data is important to recover Kubernetes clusters under disaster scenarios, such as losing all master nodes.

etcd is also sometimes used to hold state for network plugins, CRDs, and other essential components.

BUT... Some critical state is held outside etcd.

Critical items outside etcd

protection and recovery plan needed:

- Persistent volumes
- Certificate and key pairs, Certificate Authority
- ServiceAccount signing
- LDAP or other authentication details
- State associated with any CRDs and CNI plugins not using etcd
- Network resources (configuration allowing recreation of DNS records, IP and subnet assignments, switch, firewall, routing, load balancing, proxies, etc.)
- Cloud provider specific account and configuration data
- Credentials for underlying infrastructure (access keys, tokens, passwords, etc.)

Photo by Miguel Orós on Unsplash

Container Images, Helm charts, other binaries and installables

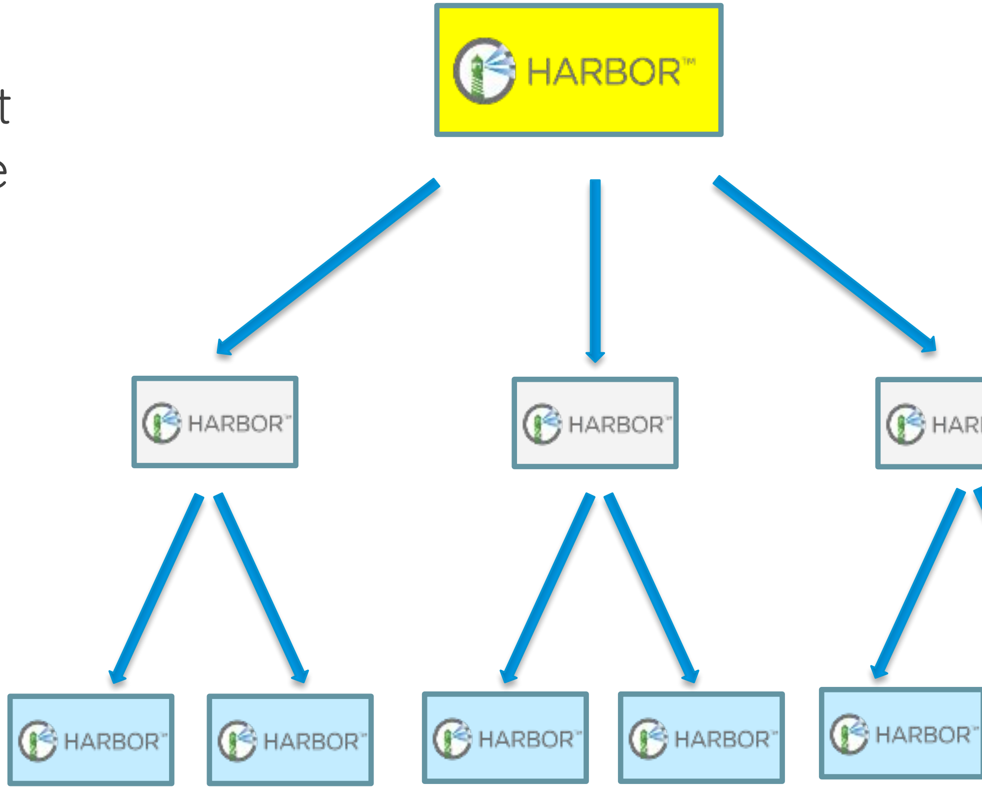
Kubernetes workloads are based on container images.

You want the source and content of these images to be trustworthy. This will almost always mean that you will host a local container image repository.

If a repository is lost, repulling images from the public Internet can be time consuming and present security issues. If you use image signing, signatures would need to be reapplied.

You may wish to consider a recovery solution for your registries.

A registry solution supporting redundancy and image replication can be a good building block for a recovery plan.



Redundancy alone is not enough

Helps reduce some types of outages… but

Bad updates, getting hacked, software bugs, or human error can simply replicate problems across redundant copies.

Photo by Andre Mouton on Unsplash

Better than a “mirror”:
Periodic backups of critical components to resilient storage

Kubernetes open source backup options

project	atomic or selective	Project contributors	Persistent volume support
etcd native	atomic	498	no
Velero	selective	92	yes
ReShifter	atomic	6	no
kaptaind	atomic	1	no

Differences in back/restore based on K8s namespaces, label selectors, etc

Use cases for Kubernetes backup solutions

Disaster Recovery

Restore clusters or applications after failures

Data Archival

Restore data that becomes lost or corrupt

Retire old data from expensive primary storage while retaining for compliance or future analytics

Migration

Migrate Kubernetes clusters or applications

Backup + Restore Demo

Photo by Dietmar Becker on Unsplash

Thank You

Q&A

Contacts

This deck: link tbd

- Kubernetes Velero Slack channel: https://kubernetes.slack.com/messages/velero
- Velero open source project Community:
  - Join: https://groups.google.com/forum/#!forum/projectvelero
  - Zoom meetings every 1st and 3rd Tuesday and recorded to YouTube channel’
  - GitHub link

Carlisia Campos

@carlisia

Steven Wong

@cantbewong