2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC)

Selective Analogy of Mechanisms and Tools in Kubernetes Lifecycle for Disaster Recovery

1st Sameer
Developer Associate, IE Tools BLR
SAP Labs India Pvt. Ltd.
Bangalore, India
sameer@sap.com

2nd Suman De
Product Manager, S4 Product Insights
SAP Labs India Pvt. Ltd.
Bangalore, India
suman.de@sap.com

3rd Prashant Singh R
Cloud Architect, Tools – Security
SAP Labs India Pvt. Ltd.
Bangalore, India
prashant.singh.r@sap.com

2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC) | 978-1-6654-9111-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMNWC56175.2022.10031926

***Abstract*—Disaster recovery is an important aspect for an organization to prevent downtime in the event of failures due to human errors, hardware failure, as well as natural disasters, and in some cases to revert to a previous state of the application. This also provides a way to migrate your application if required. With the growth in the cloud, and Kubernetes being an industry standard for deploying an application, backing up a containerized application becomes a bit tricky. There are a lot of tools available to address this requirement and choosing the right tool is critical. Every organization has a different way of managing its deployment environments and therefore need to find the tools that best fit their requirement. This paper evaluates some of the widely used tools for Disaster Recovery to find which tools fit best in a specific environment and potentially provide a way to figure out which solution is best depending on certain standard requirements. The solutions will be assessed based on both features and cost. As a result, we can implement the best possible solution in our production environments.**

Keywords—Backup, Cloud, Disaster Recovery, SAP BTP, Google Cloud, Kubernetes, Velero, Kasten

I. Introduction

There are many possible disasters that can affect availability in a cloud-based environment. The most harmful and least controllable will be natural disasters like earthquakes and floods that can cause physical loss of data centers. The next type of disaster will be physical disasters like site failures for multiple reasons like power outages, water outages, human-induced errors, and facility break-ins. [8] The third and the most common are technological disasters like hardware failures, security breaches (like system hacks, malware, DoS attacks, phishing attacks, etc), data corruption, third-party vendor failures, internet outages, and so on. [14] There is a need for an understanding of Kubernetes Lifecycle and objects to be considered for the scenario: Services, Deployments, Ingresses, PVs, PVCs, Secrets, Config Maps, Horizontal Pod Scalers, Frontend Config, Backend Config, and Managed Certificates. [3]

The K10, as shown in Fig. 1, information management framework, designed specifically for Kubernetes, offers business maintenance teams an easy-to-use, scalable, and secure system for backup/restore, disaster recovery, and Kubernetes service agility. K10's software architecture and strong interfaces with relational and NoSQL databases, Kubernetes deployments, as well as all platforms offer organizations the option to choose their infrastructure while maintaining operational simplicity [5]. K10 is regulation and adaptable, with capabilities such as full-spectrum consistency, database integrations, automated application discovery, multi-cloud mobility, and a robust web-based user experience. Because of K10's comprehensive network support, you may select platforms (public/private/hybrid cloud/on-premises) [16] and Kubernetes distributions (cloud vendor-managed or self-managed) to serve three primary use cases:

- Backup and Restore
- Disaster Recovery
- Application Mobility

The other tool, Velero is available as open-source code, with support from the community available via their GitHub campaign page. Velero (formerly Heptio Ark) offers functionality enabling backing up and restoring Kubernetes network components including persistent volumes. Velero could be operated on-premises or on the web. Velero allows you to:

- Make backups of your cluster and restore them if necessary.
- Transfer clustered assets to another group.
- Duplicate your operational cluster for development and testing.

Velero is made up of:

- A server that is part of your cluster.
- A command-line client that is only available locally.

Data backup is a key feature for organizations to prevent downtime in the event of failures caused by human error, hardware failure, or natural catastrophes, and in certain circumstances, to restore the program to a prior state. [12] This also allows you to relocate your application if necessary. With the rise of cloud computing and Kubernetes as an industry standard for application deployment, backing up a containerized application has gotten more difficult. [7] There are several solutions available to meet this criterion, and selecting the proper tool is crucial. Every business manages its deployment environments differently, so it must choose the technologies that best suit its needs. [6] This study reviews some of the commonly used Disaster Recovery tools to determine which tools suit best in a certain context and maybe provide a mechanism to determine which solution is best based on certain standard requirements.

978-1-6654-9111-2/22/$31.00 ©2022 IEEE

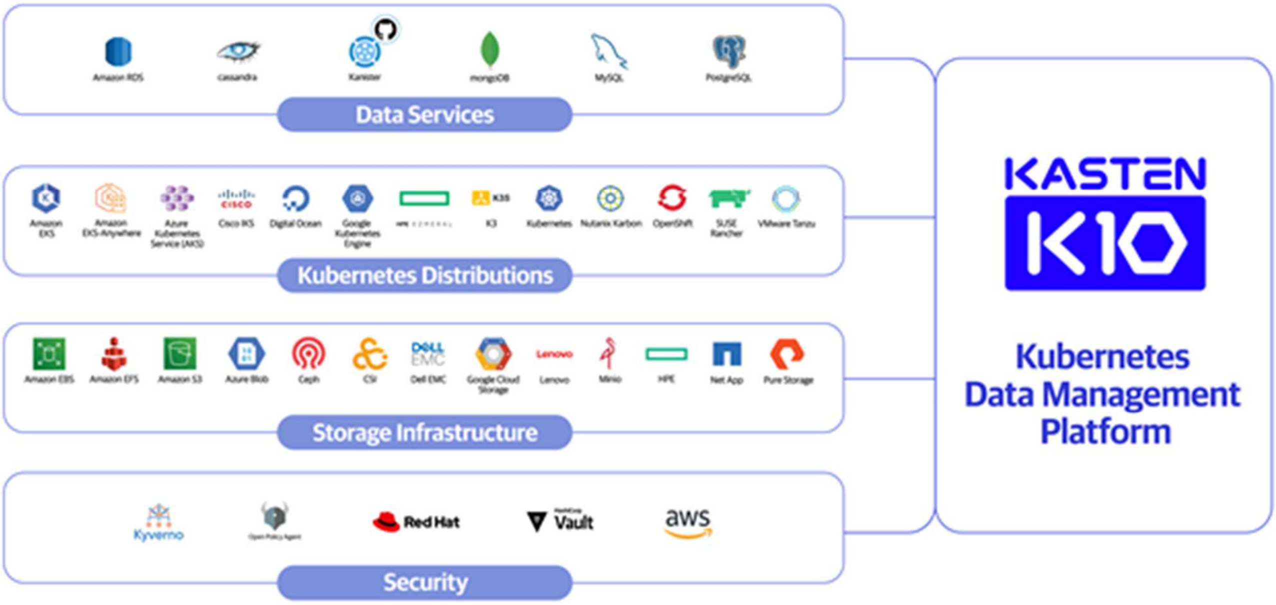


Fig. 1. K10 Ecosystem with various services, distribution, infrastructures, and security measures.

The solutions mentioned in subsequent sections will be evaluated based on both their features and their cost. As a consequence, we can deploy the best solution into our production environments. This paper covers a section for recent literature followed by a comparative study of tools and their favourability and their nuances.

II. Literature Survey

IT cloud applications and microservices supplied through cloud vendors have substantial issues in maintaining services and ensuring their continuation following a calamity. [9] Discovering means to guarantee that the process of data backup and recovery is successful for delivering maximum information uptime, flexibility, and dependability within an affordable price would be the major problem with data retrieval (DR) inside the web. Several online recovery systems were built for a single-cloud architecture; nevertheless, creating a single copy of data may not be adequate since data destruction throughout a calamity may result in irrecoverable loss. Additional approaches also include numerous reconstructions on several faraway cloud providers (Multi-Cloud). Some methods advised achieving a greater standard of dependability by creating at least 2 duplicates of the information and maintaining all copies in a centralized area or dispersing it across multiple remote sites. The disadvantages of this strategy are high expenses, significant physical storage usage, and enhanced internet traffic (particularly in the case of data-intensive cloud-based applications). They explore the difficulties created by DR in both single-cloud and multi-cloud scenarios within that study. They additionally look at past research on cloud-based DR to highlight the challenges that cloud-based DR experts believe are particularly essential. [1]

In the subsequent work, we explore the work by S.Prakash, S.Mody, A.Wahab, S.Swaminathan, and R.Paramount in their paper. In a nutshell, a contingency plan refers to the method, strategies, and protocols involved in establishing the restoration or continuance of key innovation infrastructures for an organization following an environmental or man-made calamity. It is an essential component of Firm Continuity Management (BCM), and that lacking, the commercial grinds to a halt. SME's might be able to lessen their reliance on expensive IT infrastructure by implementing disaster recovery on the cloud. This article discusses the necessity of cloud computing and the services it provides. [13] It seeks to provide the necessity for DR in enterprises, as well as how DR inside the cloud might be the greatest solution across all cloud services, and that it would be extremely useful to SMEs. [2]

Next, we look at a talk by D.Dixit and V.Kamra about implementing a data protection strategy. Managing statutory obligations, human mistakes, malware, and cluster updates necessitates information and state protection. As a result, volume snapshots have been one of the collection team's highest demanded capabilities, and they are now included within the CSI spec. Therefore, snapshots themselves are frequently insufficient. Additional considerations include application consistency, local storage, archiving, closure, and licensing. Both presenters throughout this presentation identified the issues of adopting a comprehensive privacy security strategy in a container-based system, open source software ready to aid with this, and a framework to evaluate what your environment requires. [10] This session is intended for DevOps professionals who want to increase the dependability of existing enterprise applications. [4]

III. Comparative Study

This paper evaluates some of the widely used tools for Disaster Recovery to find which tools fit best in a specific environment and potentially provide a way to figure out which solution is best depending on certain standard requirements. The solutions will be assessed based on both features and cost. [11] As a result, we can implement the best possible solution into our production environments. This paper observed the following pointers:

- High-level overview of how Velero can be used to do backup and recovery of a Kubernetes application

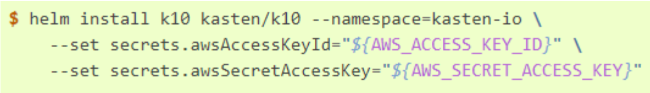
- Given all the above information and some of our environment-specific complexities we decided to try out 2 of the available option i.e., velero and Kasten k10. For our testing, we used a stateful app with resources like stateful sets deployed in Google Kubernetes Engine which is managed Kubernetes service by Google Cloud Platform.

A. Velero Implementation

- The idea was to use velero to backup Kubernetes objects along with all the persistent volume using the GCP provider plugin and velero native persistent volume snapshots instead of using restic snapshots.
- Velero Installation:
- The installation requires the following:
  - Velero client on the installed on the local machine
  - GCP bucket
  - GCP service account with permissions to make changes to the GCP bucket
  - Service Account Key for the above service account
  - Velero server installed on the Kubernetes cluster with service account key, bucket name, and compatible GCP provider plugin
- Apart from the above configuration also enables restic integration using --a use-restic flag with velero installed command
- Important Points:
  - The GCP plugin installation requires a custom GCP role to be created. Be careful while creating this new role as it can be harmful if something goes wrong.
  - After installing velero with all the required configuration, we took a backup of all the resources in the default namespace hoping it will backup all the Kubernetes along with all the persistent volumes data. After the initial test run, we found that persistent volumes are not getting backed up for some reason. Upon further investigation, we found out that the persistent volumes used in the application are provisioned using CSI drivers which require an additional plugin for backups. The first thought was to install velero with the CSI plugin for persistent volume backups. While going through the process for plugin use, we found that it required additional Kubernetes objects to be created in the cluster like VolumeSnapshotClass and VolumeSnapshot which seemed like a cumbersome process. After some more research, we found that restic is a much better option as it does not depend on the provisioner for taking backups and instead takes file system native backup. This provides two benefits:
    - We don’t need to change our velero configuration in case we move to a different persistent disk provisioner.
    - It takes file system native backup which paves a way for data migration as we can move our data from one provisioner's disk to other provisioners allowing us to move from one cloud provider to another with ease.
    - After changing the velero configuration to use restic, we again took a backup of the default namespace. This time we saw that backup details show the persistent volumes being backed up using restic.
- Testing Scenario:
  - After taking a backup of the default namespace using velero and adding some dummy data in one of the persistent volumes
  - We uninstalled one of the helm charts from the applications
- Without restic:
  - After running the restore, the helm installation came up as it was before
  - However, the content added directly to the pod manually did not get restored
- With restic:
  - After running the restore, the helm installation came up as it was before
  - The content of persistent volumes also got restored

B. Kasten K10 Implementation

- The idea here is to use kasten, Fig. 2, to backup all of the Kubernetes objects and all the persistent volumes using a VolumeSnapshotClass for CSI driver provisioned persistent volumes.
- Having a VolumeSnapshotClass for the CSI driver is a mandatory requirement.
- Kasten Installation:
  - The installation requires the following:
  - GCP service account with permissions to make changes to the GCP bucket
  - Service Account Key for the above service account
  - Consider environment variables required for installation

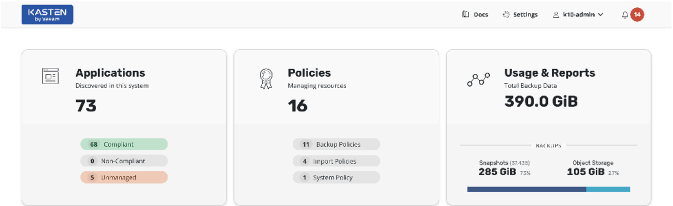


```
$ helm install k10 kasten/k10 --namespace=kasten-io \
    --set secrets.awsAccessKeyId="${AWS_ACCESS_KEY_ID}" \
    --set secrets.awsSecretAccessKey="${AWS_SECRET_ACCESS_KEY}"
```

Fig. 2. Environment Variables to be considered for AWS installation

Table I. Comparative study between Available Tools

	Cohesity	Kasten K10	Portworx (PX-Backup)	Velero	TrilioVault	KubeDR	Metallic
Support for Namespace Backups	Allows backup for entire namespaces	Allows namespace specific backups	Allows namespace specific backups	Allows namespace specific backups	Allows namespace specific backups	Does not allow namespace specific backups	Allows namespace specific backups
Support for Application Backups	Does not support backup for individual applications	Allows application specific backups with the help of labels	Allows application specific backups with the help of labels	Allows application specific backups with the help of labels	Allows application specific backups with the help of labels and helm	Does not support backup for individual applications	Allows application specific backups with the help of labels
Synchronous / Asynchronous Backups	Supports asynchronous backups only	Supports asynchronous backups only	Supports asynchronous backups only	Supports asynchronous backups only	Supports asynchronous backups only	Supports asynchronous backups only	Supports asynchronous backups only
Stateless / Stateful Application Support	Supports both stateful and stateless application backup	Supports both stateful and stateless application backup	Supports both stateful and stateless application backup	Supports both stateful and stateless application backup	Supports both stateful and stateless application backup	Supports both stateful and stateless application backup	Supports both stateful and stateless application backup
Encryption	Provides data encryption at rest.	Provides encryption for data at rest and in motion	Provides encryption for data at rest and in motion	Supports encryption for data at rest but anyone with access to bucket can decrypt	Provides data encryption at rest.	Provides data encryption at rest.	Provides encryption for data at rest and in motion
Supported Cloud Platforms	AWS, GCP, Microsoft Azure	AWS, GCP, Microsoft Azure	AWS, GCP, Microsoft Azure	AWS, GCP, Microsoft Azure	AWS, GCP, Microsoft Azure	AWS, GCP, Microsoft Azure	AWS, Microsoft Azure
RBAC Support	Supports RBAC	Supports RBAC	Supports RBAC	Does not support multi-user RBAC	Supports RBAC	Does not support RBAC	Supports RBAC
Backup Technology Support	Supports backup to file stores and S3	Supports backup to block stores, file stores and S3	Supports backup to S3 and block stores	Supports backup to S3 and block stores	Supports backup to S3 and file stores	Supports backup to S3 only	Supports backup to S3 and block stores
Incremental / Scheduled Backups	Supports both	Supports both	Supports both	Supports both	Supports both	Supports scheduled backups only	Supports incremental backups only
Miscellaneous	Cannot achieve zero RPO* objective No inbuilt support for NoSQL backups. Requires additional integrations	Cannot achieve zero RPO* objective Marketplace Integration with all major cloud providers Supports object level selection Multi-Cluster Support Supports Cluster Migration Provides templates for common use cases and allows custom templates	Cannot achieve zero RPO* objective Supports pod level selection	Can integrate with NooBaa to allow multi-cloud backups Cannot achieve zero RPO* objective Supports object level selection Open Source Supports Cluster Migration with restic integration	Cannot achieve zero RPO* objective	Only supports backup for entire cluster without any granular support Supports granular recovery Cannot achieve zero RPO*	Cannot achieve zero RPO* objective



Applications
Discovered in this system
73
68 Compliant
0 Non-Compliant
5 Unmanaged

Policies
Managing resources
16
11 Backup Policies
4 Import Policies
1 System Policy

Usage & Reports
Total Backup Data
390.0 GiB
BACKUPS
Snapshots (37,438) 285 GiB 73%
Object Storage 105 GiB 27%

Fig. 3. A typical Kasten Dashboard with Reports

IV. Advantages and Discussions

Cumulating mechanisms and tools for identifying the fault and possible recovery mechanisms have evolved with different industry-based solutions providing various advantages for Kubernetes-based practitioners to ensure that deployed applications are best monitored and repaired. This novel study with implementation for Velero and Kasten provides several benefits from a business and academic standpoint, which are as follows:

- Provide a practical reference point for industry practitioners to select and implement the right disaster recovery mechanism
- Identify tools [15] that best serve needs based on the platform and type of technology stacks being considered for deployment
- Understand the implemented Velero and Kasten tools, Fig. 3, and how they offer advantages in following a specific architecture principle in cloud-native business applications
- Select the right tool from an operational standpoint
- Prepare a checklist for the components of the technology stack in question
- Zero down on the right security approach with the required versatility
- Helps prepare the right approach for the deployment model for Cloud Native applications and APIs
- Ensures the DevOps team that a standardized set of parameters are explored
- Academically, upgrade the knowledge base on industry-grade disaster recovery tools

Along similar lines, there are challenges laid bare for Enterprise Applications using Open-Source technologies that require a check for the right tool to choose. For example, Velero is the right choice for such cases and can be selected above the other tools explored in Table I. Similar parameters decide the viability and feasibility of using any of the mechanisms as highlighted in the table and can be used by DevOps practitioners.

V. Conclusion

In the event of problems brought on by human error, hardware failure, or natural disasters, disaster recovery is a crucial component for organizations to reduce downtime and, in certain situations, to restore the program to a prior state. There are several tools available to meet this need, so picking the appropriate one is crucial. To determine which tool works best in a particular setting and perhaps offer a mechanism to determine which solution is best based on specific standard requirements, this article reviews some of the commonly used tools for disaster recovery. Based on functionality, integration capabilities, platform supports, and other vital parameters in Kubernetes, the solutions are evaluated. Then the way to select an ideal mechanism or tool for production environments is shown. This paper considers Velero and Kasten K10 and provides the step-by-step implementation done by the authors to further asses the comparison mentioned in the text. This paper is a practical study of disaster recovery mechanisms and tools that help for the Kubernetes lifecycle and explores a niche topic in cloud development.

VI. Future Scope

The fundamental need for this research was to identify various mechanisms and tools for disaster recovery and implementation to evaluate their effectiveness for any industrial usage. There is a scope to create better tools considering the updation of modern business requirements of scaling and performance with lesser absorption of resources and space. This paper creates a baseline for such novel tools to be invented and used in cloud development.

Acknowledgment

This paper was possible thanks to the guidance of resources at SAP Labs India. The author is thankful to BITS Pilani for providing the platform to work on this industry-focused paper on data recovery tools. The authors would like to thank SAP techNxt for providing a medium for upcoming authors to take the regular work commitments and channel the same for research that uplifts academic value.

References

[1] M. M. Alshammari, A. A. Alwan, A. Nordin, and I. F. Al-Shaikhli, "Disaster recovery in single-cloud and multi-cloud environments: Issues and challenges," 2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS), 2017, pp. 1-7, DOI: 10.1109/ICETAS.2017.8277868.

[2] S. Prakash, S. Mody, A. Wahab, S. Swaminathan, and R. Paramount, "Disaster recovery services in the cloud for SMEs," 2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012, pp. 139-144, DOI: 10.1109/ICCCTAM.2012.6488087.

[3] S. De, "A Study on Chaos Engineering for improving Cloud Software Quality and Reliability," 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), 2021, pp. 289-294, DOI: 10.1109/CENTCON52345.2021.9688292.

[4] Deepika Dixit, Vaibhav Kamra, “Day 2 with Stateful Applications - Implementing a Data Protection Strategy”, KubeCon + CloudNativeCon North America, Seattle, WA, USA, Dec 9-14, 2018

[5] Kasten K10 Documentation, Available: https://docs.kasten.io/latest/index.html, Last Accessed: 16.08.2022

[6] Suman De, “An efficient technique of resource scheduling in the cloud using graph coloring algorithm”, Global Transitions Proceedings, Volume 3, Issue 1, 2022, Pages 169-176, ISSN 2666-285X, https://doi.org/10.1016/j.gltp.2022.03.005.

[7] M. M. Khalel, M. Arul Pugazhendhi, and G. R. Raj, "Enhanced Load Balancing in Kubernetes Cluster By Minikube," 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), 2022, pp. 1-5, DOI: 10.1109/ICSTSN53084.2022.9761317.

[8] A. A. Tamimi, R. Dawood, and L. Sadaqa, "Disaster Recovery Techniques in Cloud Computing," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 2019, pp. 845-850, DOI: 10.1109/JEEIT.2019.8717450.

[9] L. Abdollahi Vayghan, M. A. Saied, M. Toeroe, and F. Khendek, "Microservice Based Architecture: Towards High-Availability for Stateful Applications with Kubernetes," 2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS), 2019, pp. 176-185, DOI: 10.1109/QRS.2019.00034.

[10] M. Panjwani and S. De, "Study of Cloud Security in Hyper-scalers," 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), 2020, pp. 29-34, DOI: 10.23919/INDIACom49435.2020.9083727.

[11] X. Yu, D. Wang, X. Sun, B. Zheng and Y. Du, "Design and Implementation of a Software Disaster Recovery Service for Cloud Computing-Based Aerospace Ground Systems," 2022 11th International Conference on Communications, Circuits and Systems (ICCCAS), 2022, pp. 220-225, doi: 10.1109/ICCCAS55266.2022.9825253.

[12] S. Gokulakrishnan and J. M. Gnanasekar, "Data Integrity and Recovery Management in Cloud Systems," 2020 Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 645-648, doi: 10.1109/ICISC47916.2020.9171066.

[13] T. Sato, F. He, E. Oki, T. Kurimoto and S. Urushidani, "Implementation and Testing of Failure Recovery Based on Backup Resource Sharing Model for Distributed Cloud Computing System," 2018 IEEE 7th International Conference on Cloud Networking (CloudNet), 2018, pp. 1-3, doi: 10.1109/CloudNet.2018.8549455.

[14] S. De, "A Novel Perspective to Threat Modelling using Design Thinking and Agile Principles," 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), 2020, pp. 31-35, doi: 10.1109/PDGC50313.2020.9315844.

[15] L. Zheng et al., "Disaster SitRep - A vertical search engine and information analysis tool in disaster management domain," 2012 IEEE 13th International Conference on Information Reuse & Integration (IRI), 2012, pp. 457-465, doi: 10.1109/IRI.2012.6303044.

[16] S. Hamadah and D. Aqel, "A Proposed Virtual Private Cloud-Based Disaster Recovery Strategy," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 2019, pp. 469-473, doi: 10.1109/JEEIT.2019.8717404.