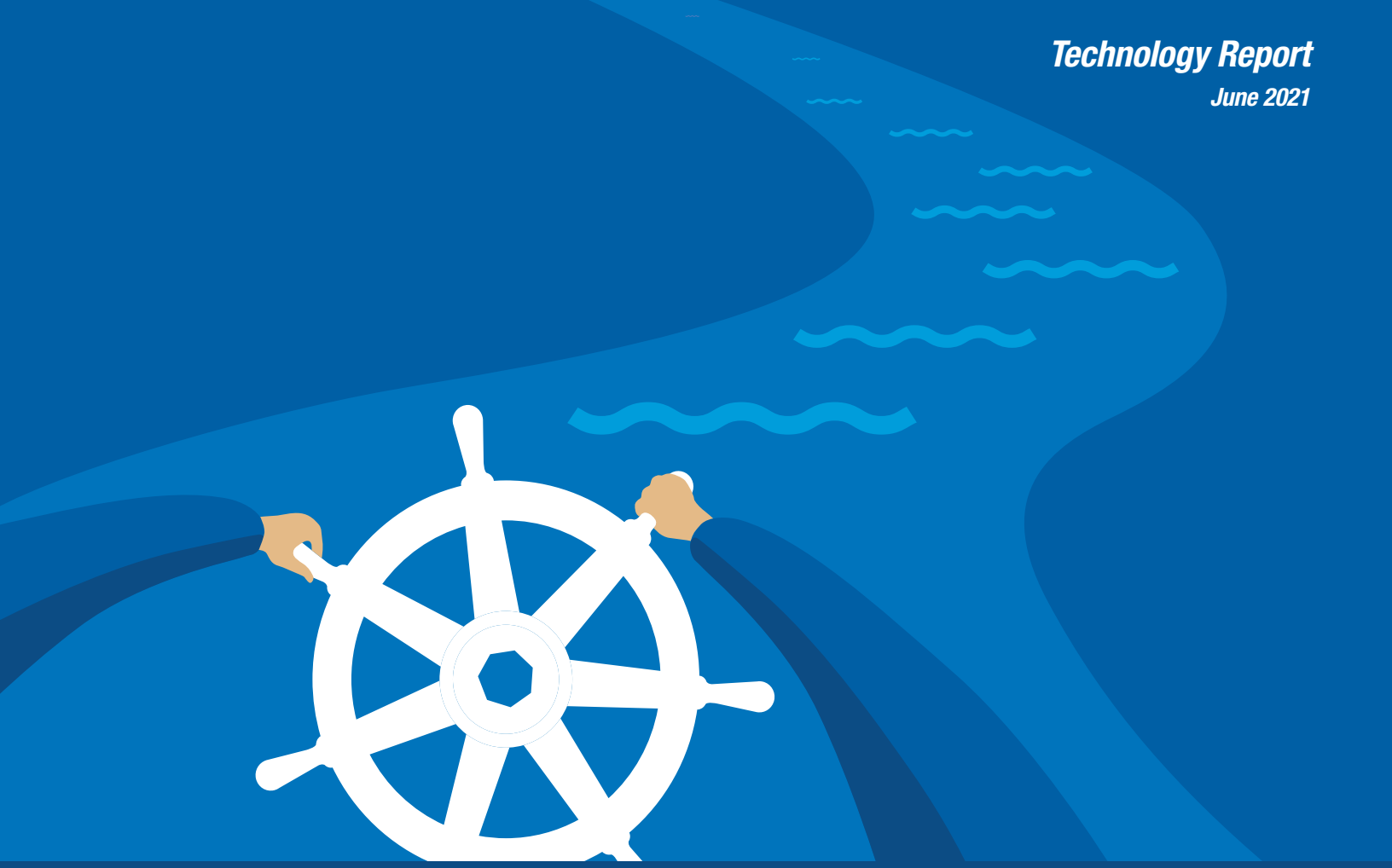
Technology Report
June 2021

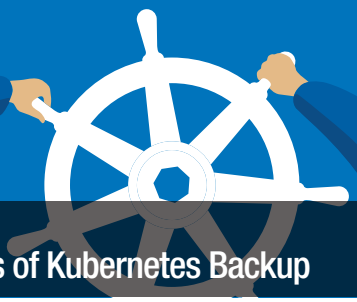


DCIG

Quantifying and Responding to the Specific Challenges of Kubernetes Backup

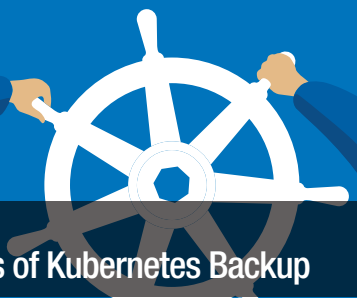
By Jerome M Wendt

HYCU



Quantifying and Responding to the Specific Challenges of Kubernetes Backup

Table of Contents



Quantifying and Responding to the Specific Challenges of Kubernetes Backup

Executive Summary

Almost every enterprise in the next few years expects to introduce and adopt Kubernetes in their production environment. These same enterprises also recognize moving Kubernetes into production requires it meet the same standards as their other production applications. Among these requirements, they must have a means to backup and recover the containerized applications they host in Kubernetes.

On the upside, commonalities do exist between backing up containerized applications and VMs. All applications are virtual, APIs exist to perform backups, and backup data still consists of ones and zeros at its core. This makes it feasible to back up and store backups of containerized applications and VMs in the same backup vaults.

However, backing up containerized applications in Kubernetes presents five distinct challenges. These include:

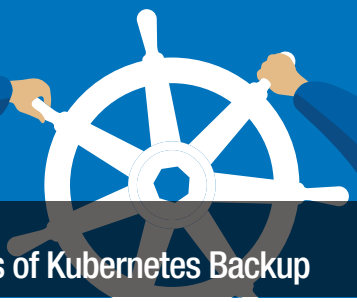
1. Protecting the temporal data and storage of containerized applications
2. Managing the unpredictable nature of containerized applications
3. Meeting the dynamics of rapidly changing Kubernetes environments
4. Providing the full breadth of backup capabilities needed for Kubernetes
5. Creating reliable restores from an ever-changing production Kubernetes environment

To meet these five challenges of Kubernetes backups, any backup solution must capitalize on specific Kubernetes features. Chief among them, it must utilize the metadata associated with each containerized application. In this way, it may quantify if it must back up the containerized application and, if so, how to best back it up.

It must then use these metadata labels for multiple purposes. Minimally, it must use them to uniquely identify which backup is which. It also should coordinate backups across multiple containers so it may synchronize reliable recoveries, if required.

Finally, enterprises will need a backup solution delivered on a cloud-native architecture to dynamically adapt to an ever-changing Kubernetes environment. Ideally, the backup solution will also be delivered as a service and manage both containerized application and VM backups.

HYCU addresses these specific challenges of Kubernetes backups. It meets continuing enterprise needs to protect their virtualized environments while positioning them to back up their new Kubernetes environment. This equips them to manage containerized application and VM backups in the same way with minimal or no operational disruption.



Quantifying and Responding to the Specific Challenges of Kubernetes Backup

The Future is Kubernetes

One of the largest transformations in computing's history appears on the cusp of occurring. In this iteration, organizations focus less on the exact hardware, software, networking, virtualization, or cloud technologies they use. Rather, they may deploy containerized applications on any cloud platform in any environment.

This vision stands poised to become a reality. Deploying containerized applications managed by Kubernetes hosted on cloud-native platforms puts this ideal within reach. By embracing these next-generation technologies organizations may:

- Accelerate and shorten application development and deployment
- Align application placement on cloud platforms with their business, cost, performance, and resource needs
- Better forecast future costs and resource requirements
- Host applications on any cloud platform at any time
- Scale IT infrastructure up or down on demand
- Use fewer IT resources to develop and host applications
- Use and re-use available IT resources more efficiently and effectively

These benefits prompt many enterprises to embrace these technologies as the future of IT. Among them, enterprises specifically anticipate using Kubernetes to coordinate and orchestrate containerized workloads and services across multiple cloud environments.

A June 2020 survey by the Cloud Native Computing Foundation found 78 percent of its members already use Kubernetes in some form.[1] A separate survey found that 64 percent of enterprise IT professionals have already deployed Kubernetes on-premises.[2]

The Google Cloud provides perhaps the best insight into how pervasive Kubernetes could become in enterprises. Google already runs all its applications on containers within its cloud launching of billions of containers each week.[3] It uses Kubernetes to deploy, manage, operate, and orchestrate the deployment of containerized applications within its cloud.[4]

Making this vision a reality requires enterprises to deploy an infrastructure that protects these containerized applications and their data. As part of this emerging infrastructure, many enterprises need a backup solution in place before formally adopting Kubernetes. Ideally, they hope to use the same software to back up their both existing virtual environment and their new Kubernetes environment.

Commonalities in Backup Characteristics

Containers and virtual machines (VMs) share multiple traits that permit enterprises to consider using a single solution to protect both. Three characteristics that stand out include:

- ***Containers and VMs are both virtual.*** Containers and VMs both operate independently of the platform that hosts them. Appropriately designed backup software may treat and protect both containers and VMs as objects.
- ***APIs available.*** The underlying Kubernetes and hypervisor platforms that host containers and VMs offer APIs. The backup software interacts with these APIs to perform tasks such as taking snapshots of individual containers or VMs.
- ***Data is data.*** Whether enterprises host their data in containers or VMs, data is still data. Backup software stores and manages all backup data in essentially the same way regardless of its origin.

Despite these similarities and others, backup software must manage container backups and recoveries differently than VM backups.

The Five Challenges of Kubernetes Backup

Backup software must address five specific challenges that backing up and recovering containers in Kubernetes creates.

#1 – Ephemeral Storage

Deploying containerized applications in Kubernetes differs in a significant way from deploying applications on VMs. If an application or its hosting VM shuts down, the application's data and VM's storage persist. In this way, the next time the appli-

1. https://enterprisersproject.com/article/2020/6/kubernetes-statistics-2020. Referenced 5/19/2021.
2. https://containerjournal.com/topics/container-ecosystems/survey-sees-kubernetes-enterprise-adoption-gains/. Referenced 5/19/2021.
3. https://cloud.google.com/containers. Referenced 5/21/2021.
4. https://cloud.google.com/learn/what-is-kubernetes. Referenced 5/21/2021.

Quantifying and Responding to the Specific Challenges of Kubernetes Backup

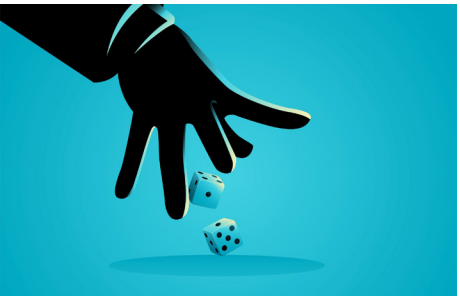
cation within the VM or the VM starts up, it can access its allocated data or storage.

Kubernetes handles the data and storage resources associated with containerized applications and their pods differently. Once an application or a pod shuts down, Kubernetes automatically recoups its allocated resources, to include its storage. Once reclaimed, the data associated with that application or pod effectively becomes "lost."

This creates a two-fold challenge from a backup perspective.

First, the backup software must become aware the containerized application exists.

Second, containerized applications may only exist for a brief time. Therefore, one cannot schedule daily backups and expect them to protect the data of these applications. Backups must occur during the time the containerized application exists.



#2 – Unpredictability

The flexibility that Kubernetes offers to start containers anywhere at any time on almost any cloud heightens its appeal to enterprises. However, this same flexibility also introduces levels of unpredictability and complexity that may far exceed environments with VMs.

A backup solution must perform multiple tasks to back up this environment. It must first address the challenges associated with containerized applications and their ephemeral storage. It must also determine if it needs to back up each containerized application since not all containers require backups.

Should it need to back up the containerized application, it must use the appropriate policy to back it up. The policy will determine how often the solution needs to back up the application. A backup may just occur once, such as just prior to the containerized application shutting down. Alternatively, backups may occur multiple times during the containerized application's life.

#3 – Backup Solution Scalability

Kubernetes facilitates the startup and shutdown of thousands, millions, and perhaps billions of containerized applications weekly or monthly.

Common Kubernetes Terms

To understand the requirements for backing up and recovering containers in Kubernetes, it helps to first understand commonly used terms. Here are terms enterprises will commonly encounter when managing containers in Kubernetes with their associated definitions or explanations.

Container. A lightweight, standalone, executable software package that includes all the code and its associated dependencies needed for an application to run.

Labels. The metadata assigned to any API object (container, node, pod, etc.) in Kubernetes. Labels are arbitrary names that identify and classify each object. Each object may possess one or more labels. Each label contains a key-value pair. A key may be BuildDate with a value of 01/01/2021. Kubernetes uses these labels to identify, manage, and organize objects.

Kube-scheduler. The default scheduler for Kubernetes that runs as part of its control plane.

Kubernetes. A system platform to deploy, scale, and manage containerized applications. It interfaces with the underlying hardware infrastructure to manage computing, networking, and storage resources. It automates container management by providing commands to deploy, monitor, and scale applications.

Kubernetes Cluster. A set of node machines for running containerized applications. A cluster minimally contains a control plane and one or more nodes. The control plane determines which applications run on which nodes in the cluster and when they run.

Node. A Kubernetes worker machine that may be either a physical or virtual machine. A node may consist of one or more pods. A node always contains a Kubelet that handles communication between the Kubenetes Master and the node. It also always contains a container runtime that pulls the container image from a registry, unpacks it, and runs the application.

Pod. A pod always runs on a node. It contains one or more containers and some shared resources for the containers in it. These resources include storage, networking, and information on how to run the container.

Kubernetes Master. A master node that controls and manages a set of worker nodes.

Service Discovery. Provided as a Kubernetes Service, it capitalizes on the labels and selectors to associate a service with a set of pods. Since a pod may only live a short time, a pod's network resource allocations may change. The Service Discovery enables external applications to dynamically discover each pod and access it regardless of its life.

The backup solution must have sufficient resources to detect all the containerized applications, schedule the backup jobs, and manage them. The solution must also possess sufficient resources to do backend backup data management and perform recoveries.

Deployed in cloud infrastructures, Kubernetes further exacerbates the backup challenges. The number of active containerized applications may scale up or down dramatically and with little or no advance warning.

To respond to and manage this environment, the backup solution must automatically and cost-effectively adapt to it. As containerized applications start up, the backup solution must dynamically scale up. Similarly, as containers go offline, the backup solution must dynamically scale down.

#4 – Breadth of Kubernetes Backup Capabilities

An enterprise's choice of a backup solution will depend, in part, upon the maturity of its Kubernetes deployment. In a stable, more mature Kubernetes environment, an enterprise often writes scripts that automate deployment of core Kubernetes components. In these environments, an enterprise may only back up its containerized applications hosted in its Kubernetes environment.

In contrast, an enterprise that has more recently adopted Kubernetes may have different backup challenges. It may have not yet finished constructing its Kubernetes environment. Alternatively, if it has, the enterprise may question if its current Kubernetes infrastructure represents the one it wants going forward.

In either case, an enterprise has likely not yet created scripts to recreate its Kubernetes control place. As a result, it will need a solution with a greater breadth of backup capabilities. The solution will need to back up both the Kubernetes control plane and the containerized applications and data.

#5 – Recovery

While backing up containerized applications presents challenges, recoveries of containerized applications present even greater challenges. To perform a recovery, the backup software must initially associate each backup with a specific application, user, or other environmental constant.

Should an enterprise need to perform a recovery, it must have some means to identify which backup to recover. The

temporal nature of containerized applications in Kubernetes environments requires reliable procedures to initiate the recovery.

As part of the recovery, an enterprise also needs to determine the breadth of the recovery in the Kubernetes environment. The software may only need to recover a single containerized application. However, interdependencies may and often do exist between multiple containers. This may necessitate the concurrent recovery of multiple containers.

The breadth of the data loss will also influence which Kubernetes components the backup software must recover. Should a Kubernetes deployment become compromised, the backup software may also need to recover the Kubernetes control plane.

Enterprises should also consider emerging requirements to recover containerized applications into other Kubernetes deployments. More enterprises use hybrid clouds with different Kubernetes versions running in each cloud. When performing recoveries, they may want to recover applications to a different cloud than where they were originally backed up.

The Kubernetes deployments used in different clouds possess many similarities. However, differences do exist. One must ensure the backup solution supports the Kubernetes deployment in any cloud into which they may need to recover.

Meeting Kubernetes Backup Challenges

To meet these five challenges associated with backing up Kubernetes environments, enterprises need to manage backups differently. This change in backup management begins in how enterprises deploy containerized applications in Kubernetes.

Effectively Utilize Container Labels

All containerized applications may optionally possess metadata that contains labels. Using these labels becomes mandatory to backing these containerized applications up in Kubernetes.

Quantifying and Responding to the Specific Challenges of Kubernetes Backup

Every time a containerized application starts, it must register with the Kubernetes node service. As it registers, three events occur.

```
"metadata": {
  "labels": {
    "key1" : "value1",
    "key2" : "value2"
  }
}
```

- First, the Kubernetes node service reads the container's pre-existing metadata labels.
- Second, Kubernetes adds key value pairs to the container's labels.
- Third, these key values help identify which application resources belong together as they may need each other to operate.

Synchronize Backups and Recoveries

A backup solution must capitalize on these features. It monitors Kubernetes container start-ups and their metadata content. During container start-up, it checks to see if the container's labels contain any key-value pairs that require the backup solution to protect it.

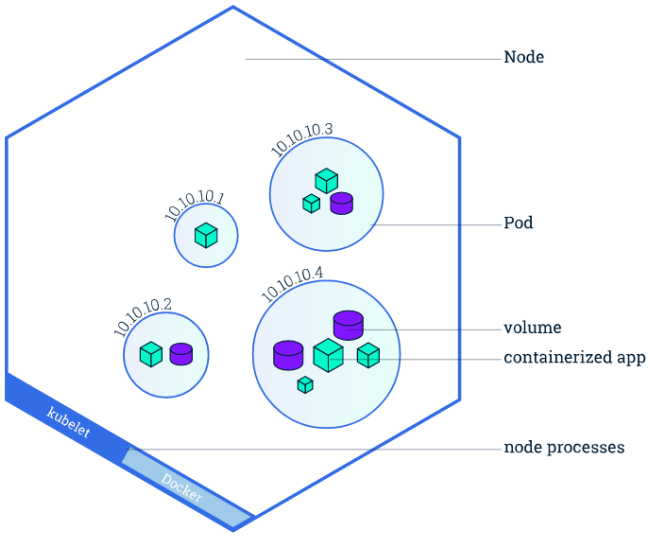
The backup software uses each container's metadata to determine if, when, and how it needs to back up and recover the containerized applications. It creates and assigns the appropriate backup policies to match their recovery requirements. Backups then occur synchronously across containerized applications that rely upon one another in production. This is done to ensure consistent recoveries.

To achieve this feat, the backup solution may schedule a backup job or jobs in kube-scheduler. The backup job will then back up those containerized applications and its associated applications at the appropriate intervals. It may also perform backups when specific events occur, such as at container shutdown.

Assign Constants to Backups

Part of successfully recovering a containerized application requires the backup software to identify exactly when an application backup occurs. To do so, the backup software must assign one or more environmental constants to each backup. In this way, it may pinpoint the appropriate backup to recover.

Environmental constants assigned to the backup may include a date and time stamp and specific node, pod, and container identifiers. Containerized applications must reside within a pod and pods must exist with a node. This combination of variables provides a means to uniquely classify where and when each application backup occurred.



Source: https://kubernetes.io/docs/tutorials/kubernetes-basics/explore/explore-intro/

Cloud-native Architecture

Scaling to meet the backup requirements of thousands, millions, or billions of containerized applications necessitates using a cloud-native architecture. In this way a backup solution may efficiently and cost-effectively scale up or down dynamically to back up applications.

HYCU: A Cloud-native Solution for Kubernetes and VM Backups

Kubernetes creates a new standard by which enterprises need to measure backup solutions. While some principles of performing backups in virtualized environments apply when protecting containerized applications, new backup and recovery requirements exist. In Kubernetes environments, the backup software must perform the following tasks dynamically:

- Detect the creation of containerized applications
- Apply the right backup policy to each one
- Identify the dependencies that may exist between various containerized applications
- Scale up or down to potentially handle millions of different backup-related activities

HYCU grants enterprises the ability to address their backup and recovery needs in a Kubernetes environment. Built

Quantifying and Responding to the Specific Challenges of Kubernetes Backup

upon a cloud-native architecture, HYCU automatically and dynamically scales up and down to meet changing Kubernetes backup workloads.

By interacting with Kubernetes, it protects both the Kubernetes control plane and containerized applications. To protect applications, HYCU reads pre-existing application metadata labels. HYCU then programmatically assigns the appropriate backup policies to each containerized application based on their labels.

HYCU also offers recovery validation services to support recovering application(s) hosted in Kubernetes in other clouds. It already supports the Google Kubernetes Engine (GKE) with support for other Kubernetes cloud services coming shortly.

HYCU's single platform offers a unified user experience to manage data protection across Google Compute Engine (GCE), Google Cloud Storage, and GKE. Enterprises may assign the same policies to GKE applications that they already use to protect GCE instances and applications running on them.

Delivered as a service, HYCU provides enterprises with a single solution to protect both their containerized and virtual applications. Its management console administers both Kubernetes and virtual environments. This equips enterprises to centrally schedule and perform backups; archive and copy backup copies; assign backup policies; and, perform recoveries.

By adopting HYCU, enterprises may continue to meet the evolving backup needs of their existing virtualized environment. HYCU simultaneously positions them to address their emerging Kubernetes backup requirements. In so doing, HYCU provides a great option for enterprises to transition to Kubernetes while ensuring the integrity of their data in both environments. ■

DCIG, LLC // 7511 MADISON STREET // OMAHA NE 68127 // 844.324.4552
dcig.com