



White Paper

# Innovate Faster by Exploiting Data as a Strategic Asset on Kubernetes

Powered by ROBIN

## CONTENT

Why run stateful applications on Kubernetes? .....	3
Requirements for running stateful applications on Kubernetes .....	3
What is Robin.io?.....	4
<b>Use cases and business benefits of Robin.io .....</b>	<b>5</b>
Run Stateful apps on Kubernetes without becoming a storage expert.....	5
Offer Database (DBaaS) and Big Data (BDaaS) as-a-Service .....	5
Innovate faster using agile Dev/Test refreshes .....	6
Reduce infrastructure costs by 70% .....	6
Avoid cloud and infrastructure lock-in.....	7
<b>Try it for free .....</b>	<b>7</b>
<b>ROBIN.IO.....</b>	<b>7</b>

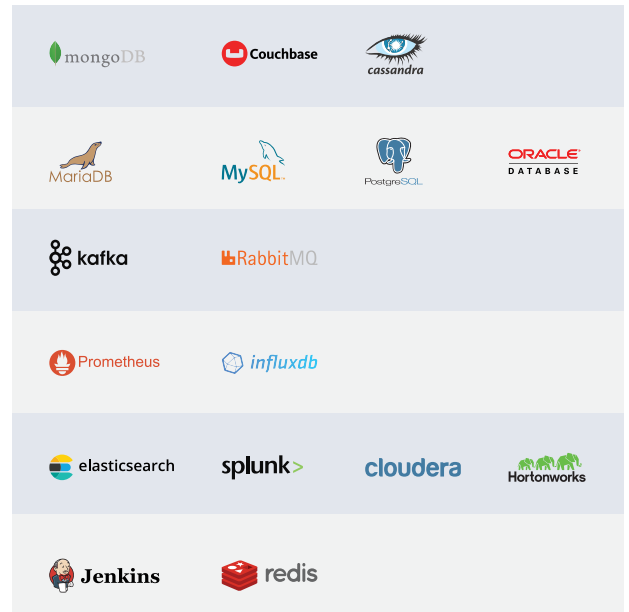
Many enterprises are going through a digital transformation journey to bring agility into their organizations and become more operationally efficient using cloud-native technologies such as Containers and Kubernetes. By 2022, more than 75% of global organizations will be running containerized workloads in production, which is a significant increase from fewer than 30% today<sup>1</sup>. Kubernetes has emerged as the de-facto platform for orchestrating and managing the lifecycle of containers. Enterprises are increasingly relying on Kubernetes to insert velocity into their organization, reduce costs, and enable hybrid cloud strategy.

However, owing to the early days of thought leadership around containers, some developers and architects still believe that Kubernetes is strictly for “stateless” applications, with “stateful” applications running outside it. But an “application” is never made up of just stateless components; even a so-called stateless application depends on state which is either coming over the network or data that is stored in a persistent database and big data platforms.

## Stateless Applications



## Stateful Applications



## Why run stateful applications on Kubernetes?

Every stateless application depends on one or more stateful applications, and not bringing them into the single deployment and management-fold of Kubernetes would constrain application development speed as developers must adopt slow, less automated processes for their data applications compared to the rapid and agile ones for their stateless components. Running stateful apps separately in their own dedicated servers would also mean losing out on the consolidation benefits of Kubernetes. This divide of stateless and stateful apps on Kubernetes results in higher costs, slower innovation velocity, and higher staffing needs from disparate operational workflows.

## Requirements for running stateful applications on Kubernetes

With the promotion of Container Storage Interface (CSI) to general availability in 2018, Kubernetes now has the ability to provide a standardized specification through which external storage solutions can offer storage and data management

1 Gartner Best Practices for Running Containers and Kubernetes in Production, February 25, 2019

capabilities to stateful applications running directly on Kubernetes<sup>2</sup> .

However, the challenges with running stateful applications on Kubernetes in a reliable and performant way is often not fully recognized by those in the initial evaluation stages of this journey. Typically, their planning is limited to selecting a storage solution that offers a CSI compatible plugin, but storage is just the first step as enterprises deploy stateful applications to production. Stateful apps bring with them requirements of resiliency from data loss and corruption from faulty disks; application availability in the event of server failures; data protection and disaster recovery to ensure business continuity, etc.

These requirements are not new and customers already know how to manage stateful applications when deployed on physical servers or virtual machines. In such environments, storage systems have relationships with hosts rather than directly with the applications. By contrast, in a Kubernetes environment, applications are abstracted from their underlying host and can be dynamically placed on any host in the cluster. Storage systems must, therefore, offer a highly resilient, secure and performant solution in such a dynamic environment. Furthermore, achieving data protection, security, and collaboration requires acquiring a very good knowledge of complex storage-centric concepts like snapshots, backups, QoS, clones, etc. Expecting developers or DevOps engineers to become storage experts to be able to run stateful applications on Kubernetes is not practical from a staffing and budget perspective.

Therefore, when evaluating a solution to run stateful applications on Kubernetes, one needs to ensure that all four of the below requirements are accounted for so that they are not surprised after they become operational on Kubernetes:

1. High-Performance Kubernetes-native storage that guarantees availability and resiliency in a highly dynamic environment even under server, disk, and software faults.
2. Powerful automation capabilities to achieve day-2 operational simplicity through developer-centric workflows without expecting DevOps to become storage experts.
3. Recognizing that organizations still rely on several complex legacy enterprise applications that predate Kubernetes, such as Oracle, Cloudera, Splunk, etc., and extending the agility and operational efficiency benefits of Kubernetes to them. That is, do not leave any applications behind in your digital transformation journey.
4. Provide a cohesive and consistent solution that runs on the cloud, on premises, at the edge, and across multiple cloud providers as well so that developers don't need to change their tooling or workflows as they port their applications across different environments.

## What is Robin.io?

Robin.io enables enterprises to accelerate their digital transformation journey by allowing them to deploy and manage both stateless and stateful applications on any open source or commercial distribution of Kubernetes, on-premises and in any cloud. It offers a radical simplification for both cloud-native and legacy enterprise workloads by allowing them to be offered back to the business in a self-service model on an agile containerized Kubernetes platform. It also allows them to innovate faster by exploiting data in stateful applications as a strategic asset on Kubernetes. By doing so our customers are inserting velocity into the enterprise, reducing costs, and enabling a diversified hybrid cloud strategy if they so choose.

---

<sup>2</sup> CSI is only a specification by which an external storage solution can offer storage and data management capabilities to apps running on Kubernetes; it by itself is not a storage solution.

Robin.io solution, at its core, includes an enterprise-grade Kubernetes native storage which is at least 2 to 6 times faster than other solutions. But instead of focusing on storage only, Robin.io elevates the experience to applications, making traditional storage-centric constructs such as data protection, security, governance, and portability accessible to developers without requiring them to become storage experts.

Robin.io is uniquely positioned to do this because we have architected a storage and data management stack from the ground up for Kubernetes and made it application-aware. Application-awareness allows Robin.io to radically simplify day-2 operations such as taking local and remote backups, cloning databases from production to run test/dev workflows, perform upgrade validation without risking the integrity of production apps and also deliver predictable performance SLAs to mission-critical apps. A built-in git-like workflow makes applications, and their data, portable across clouds and Kubernetes clusters.

Robin.io's solution runs natively on open source Kubernetes, Red Hat OpenShift, as well as on any managed Kubernetes in AWS, Azure, GCP, and IBM Cloud. Because of its capabilities, Google has partnered with Robin.io for running stateful database and big data apps on Anthos/GKE Kubernetes platform.

## Use cases and business benefits of Robin.io

### Run Stateful apps on Kubernetes without becoming a storage expert

Robin.io's solution includes scaled-out block storage that implements the Container Storage Interface (CSI) spec; this block storage is both software-defined and Kubernetes-native. It automates the provisioning of high-performance and highly-resilient storage volumes for any stateful applications. By abstracting the underlying physical media, such as direct-attached disks or SAN hosted luns on-premises, and cloud-drives in public cloud environments, it provides a consistent set of features and capabilities for any stateful app in any environment. With Robin you don't have to become a storage expert to deliver the strict performance and availability SLAs that stateful applications demand. Using Robin's DevOps friendly tooling one can get deep insights and visibility into the health and utilization of all your resources, all the way from applications down into regions of disks where those applications store their data. The visualization provides a clear picture of potential performance bottlenecks, blast radius when disks or servers fail, and consumption and utilization details down to users and applications. When things go wrong, it offers recommendations and simple workflows to keep your stateful applications healthy and performant. Features such as Data availability through replication, Data security through data-at-rest and in-motion encryption, Data reduction through in-line compression and thin-provisioning are built into the core product and exposed in a developer-friendly manner so that one doesn't have to become a storage expert to leverage these features.

### Offer Database (DBaaS) and Big Data (BDaaS) as-a-Service

Provisioning a database or big data typically involves many steps taken by multiple departments including provisioning new servers, setting up storage, configuring networking, tuning the OS, installing database/big data software, creating a schema, configuring user authentication and access control, etc. These tasks take a lot of time and effort in planning, coordination, and execution from multiple teams such as sysadmin, storage admin, network admin, DBA's, and Big Data architects. Using Robin's DBaaS and BDaaS capabilities on Kubernetes, one can radically simplify the provisioning and management effort by enabling single command operations that will abstract all the effort mentioned above. Robin offers 1-click self-service provisioning of any open source or commercial database and big data workloads on Kubernetes, including Oracle RAC, MongoDB, Cassandra, SAP HANA, PostgreSQL, MySQL, MariaDB, SQL Server, Elastic, Splunk,

Cloudera, etc. Provisioning a production-ready database or big data workload in a self-service manner typically takes under 5 minutes even for large complex distributed databases.

## Innovate faster using agile Dev/Test refreshes

Every enterprise wants to roll out new features quickly, but this requires developers to have the ability to quickly make copies of entire application stacks, including the data stored in stateful databases and big data platforms. With Robin, developers can clone entire application stacks, including data, for a range of use cases including test/dev, UAT, analytics, training, security patch testing, and more. Clones can be instantiated in a self-service manner on-premises or in the cloud. To make them truly useful and ready for immediate use, Robin clones not just the data volumes, but also the metadata and configuration of all stateless and stateful components of the entire application stack. This results in creating clones in a matter of minutes instead of the days that most enterprises spend today through manual or exhaustive DIY scripting work. Our customers report up to 200 times improvement by adopting Robin and cutting down their test/dev refreshes from 2 days to as little as 15 minutes. With this, they are able to build new features faster, perform more agile dev/test iterations, and roll them out to production faster.

## Reduce infrastructure costs by 70%

Running stateful database and big data applications outside Kubernetes on their own dedicated server clusters results in significant underutilization of resources. Customers routinely report their servers running at 20-50% utilization. Dedicating server clusters for different stateful applications also means that it is impossible to adjust compute and storage resources to meet changing priorities as they can't easily be brought over from adjacent underutilized servers into more heavily utilized ones. This problem worsens over time as servers gain more compute and storage resources at the same price point with every hardware refresh. Consolidating multiple applications onto the same Kubernetes cluster is the solution, however, consolidating data-intensive workloads without guaranteeing resource and performance isolation would result in unpredictable performance as applications trample on each other's use of resources. Kubernetes already offers mechanisms to restrict CPU and memory consumption on a per-application basis, but lacks any such control on storage bandwidth. Consider the case when two databases are sharing the same physical servers and a user of one database writes a complex query that hoards most of the storage bandwidth on that server starving the other database. This results in unpredictable performance and unhappy users. Robin solves this problem by allowing admins to set limits of the storage bandwidth each stateful application can use. It also allows this limit to be dynamically adjusted to meet changing priorities while applications are still running. Armed with this capability we routinely see Robin customers pack as many as 2 to 3 times more stateful apps onto the same infrastructure without compromising performance SLAs. They also have the ability to run mixed workloads such as Kafka, Postgres, Redis, Elastic, Cloudera, etc on the same Kubernetes cluster. This results in cutting down their IT spend by as much as 70%.

Extend agility benefits of Kubernetes to legacy applications

Most enterprises still rely on several legacy enterprise workloads such as Cloudera, Hortonworks, Splunk, Oracle RAC, Elastic, SAP HANA, DB2 etc. These workloads are not built following the microservices architectural pattern advocated by Kubernetes. Furthermore, for their proper functioning, these legacy workloads require strict enforcement of complex data affinity, anti-affinity and locality constraints which are beyond the scope of the Kubernetes scheduler. Leaving these applications behind results in a partial realization of the digital transformation journey. This is especially true if most of your organization's data that must be analyzed and visualized resides in these legacy applications. Robin offers a framework called Super Operator that allows one to run any legacy application on Kubernetes. Robin also enables a 1-click

self-service capability to deploy, scale, and manage the lifecycle of these applications without requiring someone to become experts in the internals of Kubernetes. By doing so customers benefit from having a single common Kubernetes platform that can run both cloud-native and legacy applications. It also provides operational consistency, by allowing organizations to apply consistent operational processes across their entire application portfolio, without limiting the benefits of containers and Kubernetes to only a subset of applications.

## Avoid cloud and infrastructure lock-in

Cloud-native architecture is about how you build applications, not where you build them. This would allow enterprises to build and run cloud-native applications in their own data center just as well as in any public cloud environments. Being cloud-native also provides the flexibility to migrate applications from on-premises to cloud, or across different clouds to avoid lock-in take advantage of the unique services and cost economics of different environments. Kubernetes enables this by providing an abstraction that hides the differences among the different environments. However, by itself this is only limited to stateless workloads. The gravity of data for stateful applications and the variance in data management services across the different environments makes it difficult to migrate stateful components of your application stack across on-premises and cloud and across clouds. Even if it can be done, it would require a significant investment in staffing to make this portability possible. Robin's unified application-centric storage and data management capabilities make such portability very easy, with just 2 easy commands. In addition, the capabilities are offered in a fully programmable manner allowing you to integrate them into your automation workflows. By offering these in an application-centric manner, using Robin.io's solution, one doesn't have to be a cloud or storage expert to leverage these features when porting an entire application stack that is comprised of both stateless and stateful applications.

## Try it for free

Head over to <https://get.robin.io> to download and evaluate the full functionality of the Robin solution for free.

Join us on <https://slack.robin.io> to start a conversation on how Robin can help with accelerating your digital transformation journey on Kubernetes.

## ROBIN.IO

Suite 600, 224 Airport Pkwy  
San Jose CA 95110  
[info@robin.io](mailto:info@robin.io)

Website: [www.robin.io](http://www.robin.io)

Twitter: [twitter.com/RobinSystems](https://twitter.com/RobinSystems)

LinkedIn: [www.linkedin.com/company/robin-systems](https://www.linkedin.com/company/robin-systems)

Facebook: [www.facebook.com/RobinSystems](https://www.facebook.com/RobinSystems)

Robin.io, the Robin.io logo and ROBIN Hyper-Converged Kubernetes Platform for Enterprise Applications and Application-to-Spindle Quality of Service Guarantee are trademarks or registered trademarks of Robin Systems, Inc., and are protected by trademark laws of the United States and other jurisdictions. All other product and company names are trademarks or registered trademarks of their respective companies.