

Pure Service Orchestrator and Rancher: A Story of Two Halves



In my interactions with many companies over the years, I've seen traditional IT and system admins transform into new personas - like DevOps admins or infrastructure engineers. Yet, as their roles have evolved, they predominantly manage and automate servers, network, and storage infrastructure in traditional ways.

These transformations have led to challenges that generally fall into two categories:

- Traditional functional data-center roles face a rapid pace of change driven by modern application workflows taking center stage over infrastructure.
- Providing for data storage, accessibility, protection, and management is increasingly important for applications and end users.

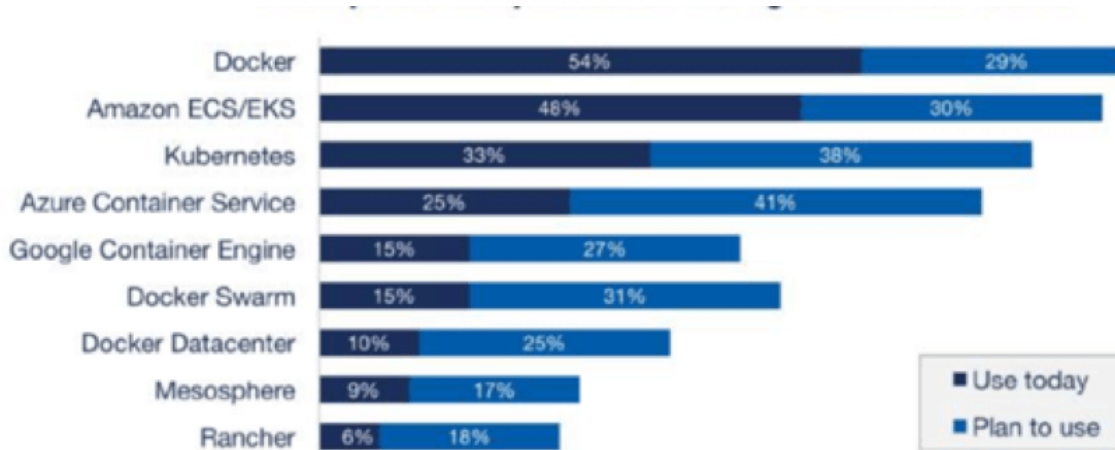
Containers are becoming a mainstream tool for virtualizing applications across industry vertical workflows. They provide modularity and portability on any platform in on-premises or public-cloud environments. Transitioning to containers opens up challenges including:

- Moving or rewriting legacy applications
- Updating workflows with new automation
- Managing container lifecycle and orchestration
- Disaggregating storage from compute for application scalability

Infrastructure engineers or DevOps admins face a variety of pain points when implementing container orchestrators like Kubernetes alongside a container runtime environment and disaggregating storage from compute for application scalability.

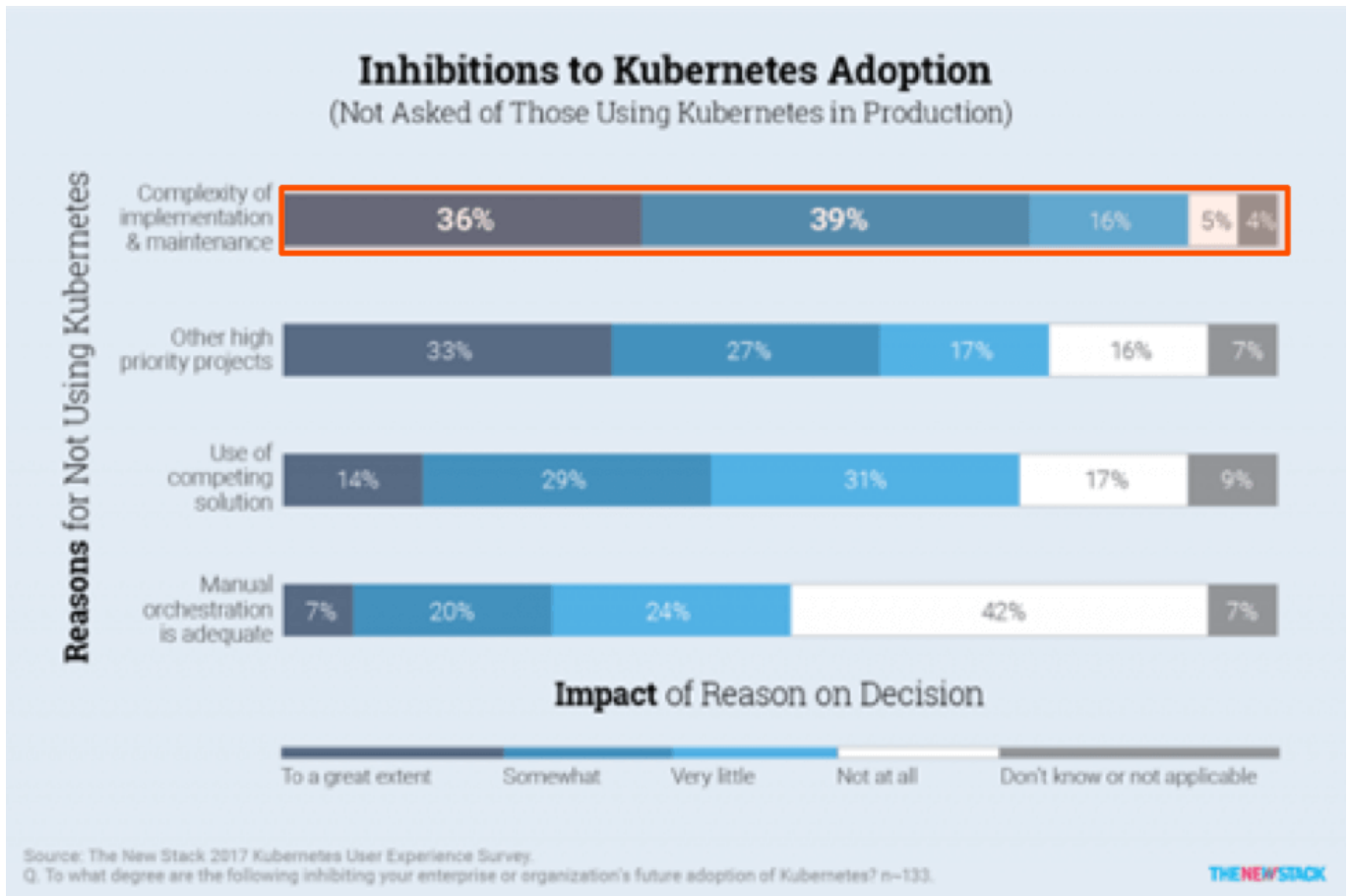
Kubernetes and Docker Lead the Rise of Containers

Open-source Kubernetes and Docker are among the most commonly deployed in production – apart from public cloud service offerings – according to RightScale. Kubernetes as an orchestrator and Docker as a container runtime environment. Kubernetes is becoming an industry standard for distributed implementation of applications. It’s robust, extensible, and portable. And it simplifies migrating applications within Kubernetes environments both on-premise and in cloud.



Source: RightScale 2018 State of the Cloud Report

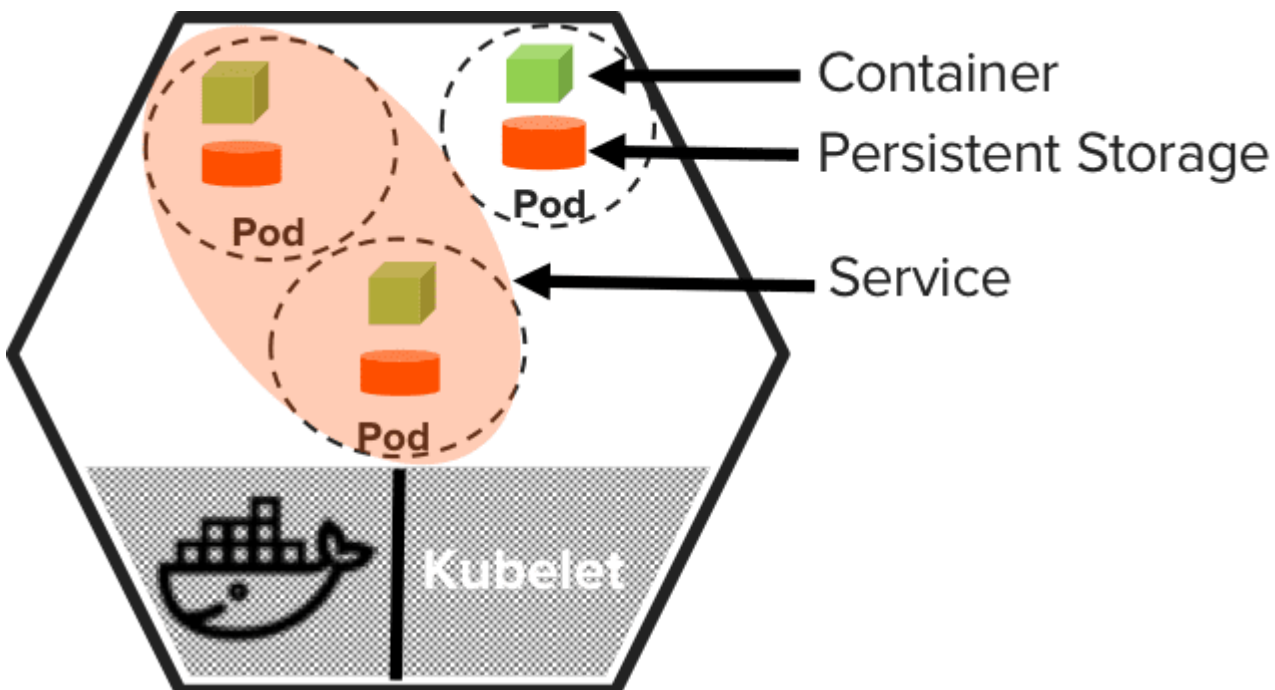
However, 75% of responses from a survey by The New Stack suggest that implementing and maintaining Kubernetes is complicated. In many cases, it’s a lack of skillsets and knowledge about automating stateful applications running in containers to consume persistent storage on demand that inhibits Kubernetes adoption.



Here are some tips on implementing Kubernetes clusters and their dependencies in a production environment.

Kubernetes Essentials

A pod is a basic functional unit that runs on Kubernetes cluster nodes. Each pod consists of one or more containers. For simplicity, assume that every pod runs a single container. A service is an abstraction layer that groups a set of pods accessed over the network.



For the pods to communicate within a node and allow external access to the service requires two sets of virtual IP ranges that are not used elsewhere in the network. It also requires configuration of another private IP range on the physical interfaces of each node for internode communication in the cluster. Apart from the network setup, a few other configurations are required on the cluster nodes during the planning phase.

More and more applications are being deployed in Kubernetes for scalability and resiliency. [Pure Storage Pure Service Orchestrator™](#) (PSO) is designed to disaggregate the underlying storage – FlashArray™ and FlashBlade™ – for different latency- and bandwidth-sensitive workloads. Pure Service Orchestrator is an out-of-tree dynamic provisioner for different storage-class objects like block and file. The provisioner is responsible for identifying the storage class requested by the persistent volume claim (PVC) and provisioning the persistent volume (PV) for the corresponding PVC. Pure Service Orchestrator is a first-class Kubernetes citizen.

For more details, refer to the [Rancher and Pure Service Orchestrator](#) white paper.

Automating the Kubernetes Installation

For me, it was intimidating to run through a sequence of manual steps to validate and install the cluster. I wanted a tool that could automate the installation and validation processes for setting up a production-grade cluster.

Many available open-source Kubernetes installers can significantly reduce the complexity and the time to set up and configure a Kubernetes. I chose Rancher 2.x. (Others include kubernetes, kops, and Kubespray.) Rancher is an open-source Kubernetes management tool that provides cluster operations and application workload management backed by enterprise support. It's easy to use and outfitted to help enable installing and configuring production-grade Kubernetes clusters.

Rancher can add and remove nodes, upgrade Kubernetes along with the Pure Service Orchestrator versions, and install applications from the Rancher catalog or from stable helm starts from the Rancher GUI – without disrupting the workload cluster. This builds confidence in new players like me to kick-start a new Kubernetes cluster with ease.

Disaggregating Compute and Storage to Drive Scale

Once the production-grade Kubernetes cluster is ready for use, the next step is to figure out how to prevent stateful applications from being held hostage by underlying disks and to eliminate the silos of dedicated servers that limit horizontal scalability. Disaggregating compute from storage to enable horizontal scaling of applications and datasets independently is a must for all stateful applications.

Pure Service Orchestrator not only provides storage disaggregation but allows AI/ML, analytics, messaging, search, log, and other workloads to run on a single platform like Pure Storage FlashBlade, which functions as a data hub for various workloads to scale independently of applications. Together, Kubernetes, Pure Service Orchestrator, and FlashBlade enable capacity and performance scaling on shared storage. FlashBlade capabilities also extend to provide adaptive service-level objectives like latency, IOPs, and bandwidth per workload to function as a data hub.

A composable infrastructure using the Pure Storage dynamic provisioner for a production-grade Kubernetes cluster on FlashBlade provides API-based automation and persistent data- and file-store management that disaggregates storage from compute and memory. The dynamic provisioner runs as a pod in the cluster.

The dynamic provisioner checks that the storage class – file or block – matches supported resources, then

provisions persistent storage as needed. The provisioner uses the FLEX plugin custom driver installed in each node in the Kubernetes cluster to attach/detach and mount/unmount persistent volumes to containers in a pod.

Pure Service Orchestrator not only provides operational efficiency by disaggregating storage from compute but also application resiliency from failures. It does this by providing data continuity to applications and services to enhance the user experience. The file systems that Pure Service Orchestrator provisions on FlashBlade over NFS allow linear performance and capacity scaling with a high degree of data compression for many different workloads, all in a smaller storage footprint.

Kubernetes Made Simple

Simplifying Kubernetes setup and configuration using tools like Rancher 2.x helps admins create and maintain production-grade clusters quickly and effectively. With more and more stateful applications running in containers, Pure Service Orchestrator provides a scalable, efficient, and dynamic way to provision and manage persistent storage by disaggregating it from compute and memory. The [“Pure Service Orchestrator and Rancher”](#) white paper provides more detail on how to simplify Kubernetes cluster creation and configure persistent storage for stateful applications on demand using Pure Service Orchestrator.

Hybrid Strategy

Rancher gives you the ability to provision Kubernetes clusters in both on-premises data centers and within public-cloud providers. The flexibility of Rancher’s “custom” cluster type allows you to provision stretched Kubernetes clusters with worker nodes contained in both environments. However, it’s generally recommended to run two distinct Kubernetes clusters for more typical deployment scenarios.

When examining a future-proof multi-cloud strategy, it can prove beneficial to utilize multiple public-cloud providers. Rancher can help with provisioning of Kubernetes clusters in many clouds, all managed through a single user interface. Rancher’s single-pane-of-glass view into multiple Kubernetes clusters provides the ability to safely deploy cloud-native applications in a way that allows for resiliency and scalability.

Organizations looking for more effective scaling of burst workloads can utilize tools to enable the dynamic scaling of Kubernetes clusters managed by Rancher. For example, it’s possible to use an auto-scaling group to automatically start instances in AWS to increase the capacity of a properly architected cloud-native solution. Also, with the use of a cloud-native block store solution such as Pure Cloud Block Store, it is possible to enable a migration burst path. This allows your application to seamlessly move from an on-premises data center with static scale to the public cloud where it can run with perceived limitless boundaries, all while keeping stateful data intact and replicated.

With Rancher’s built-in Prometheus integration, it is possible to monitor and alert on your deployed cloud-native application. Cloud Block Store lets you look for performance bottlenecks within microservice architected applications, with a large degree of certainty towards the stability and performance of your state store.

In closing, I would summarize the main data points – simplifying [Kubernetes](#) implementation and single pane of management for hybrid cloud environments with Rancher and on-demand persistent storage for stateful applications using Pure Service Orchestrator natively from Rancher. These are three main ingredients to kick start Kubernetes cluster(s) environment on-premise and in the cloud for stateful applications.