RAFAY

The Definitive Guide to Kubernetes Automation & Governance for Platform Teams

WHITE PAPER

Contents

Introduction

Today, Kubernetes (K8s) is a central force in the cloud-native computing universe and plays a significant role in most major cloud trends today, from containerization and microservices architecture to edge computing to AI and machine learning workloads.

Around 5.6 million developers now use Kubernetes, according to research from SlashData[1]. That staggering number equals roughly one-third of all backend developers. Additionally, two out of three developers writing applications for the edge are building on top of Kubernetes. This popularity may sound like a positive trend, and it mostly is. However, there is undeniable risk and complexity which grows as K8s adoption increases. If not managed appropriately inside your organization across development and operations teams, Kubernetes can lead to significant productivity issues.

Platform teams are on the rise within enterprise organizations to help wrangle Kubernetes, and for good reason. According to a Gartner study[2], an enterprise's first order of business when scaling DevOps workflows in multicluster Kubernetes environments is to establish platform teams that standardize cluster management practices, build an integrated container ecosystem and reduce operations toil for product teams.

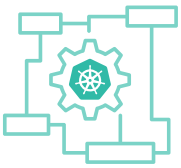
There are a lot of things to consider in order for your K8s investments to be successful. This paper explores how platform teams can align stakeholders, considerations for creating a standard operating model, and the critical features needed in a shared self-service platform.

1

Kubernetes Growth is Leading to Chaos

Kubernetes' reputation for power is matched by its reputation for its complexity—especially when organizations go it alone and build out their own Kubernetes environments in-house. What might start out as a seemingly simple deployment of a single cluster can become a production inhibitor in short order and has been known to drown even an experienced DevOps team in operational overhead.

The breakneck pace of innovation and change in the Kubernetes community—and the broader cloud-native landscape—are actually leading to a counterintuitive outcome: Many companies aren't achieving the expected product innovation and progress because development and operations teams are struggling to keep up with software updates, tools, integrations and more. This is further exacerbated by the acute shortage of operations and DevOps engineers with Kubernetes and cloud-native experience.



Updating & managing clusters one by one is often a **tedious & manual job** for operations teams **prone to human error & requires a significant investment** in headcount for K8s experts as you scale.

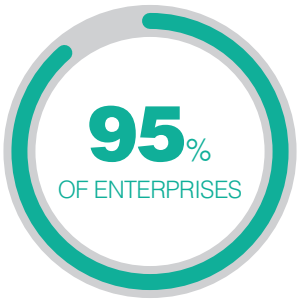
Increased Cost and Security Risk

Wild wild west workflows are becoming more common across Kubernetes shops, slowing down innovation, introducing risk, and increasing support and cloud costs.

Cloud service providers make it easy for teams to spin up Kubernetes clusters at will, frequently turning developers into shadow Kubernetes admins. When every development and operations team has its own separate workflow for creating and managing clusters, it becomes impossible to manage. Leaders have no centralized visibility or control over their infrastructure.

Without standardization and controls, clusters across teams very quickly become snowflakes and introduce significant risk. There's no way to ensure the proper software add-ons have been applied for security and reliability, exposing your production workloads to cyberattacks. There's also no way to ensure each cluster is up to date with the latest Kubernetes version, which is critical for security and interoperability purposes.

Updating and managing clusters one by one is often a tedious and manual job for operations teams, prone to human error and requires a significant investment in headcount for K8s experts as you scale. This significantly increases the cost of these K8s environments.



will **fail to scale DevOps initiatives** without a shared self-service platform approach by 2025[3]

Planning for cloud costs and enforcing infrastructure standards is also tricky since it's hard to get visibility across all your siloed clusters. This significantly increases cloud spend as teams are blind to their actual usage until a much larger than expected cloud computing bill arrives.

Productivity Drains for Operations and Development Teams

Kubernetes complexity is straining operations teams and slowing down developers. Ops teams are inundated with managing all the tools and ongoing maintenance needed to support K8s environments properly. Developers are overburdened and blocked by K8s-related ops tasks, needing to master a multitude of tools and skills to test and deploy code across different environments, microservices, and clusters. Because of this, developer productivity is significantly hindered, leaving developers with two choices: master all the tools and processes necessary in deploying infrastructure or wait through lengthy ticket-based processes for infrastructure tasks gated by operations teams.

Adopting a Platform Approach to Kubernetes

Platform engineering has emerged to help standardize the software development process and improve developer and operations efficiency. Building a platform that acts as the middleman between developers and cloud infrastructure has given teams major gains in productivity. Kubernetes, however, is far more complex than the cloud infrastructure of the past. It's time to bring that same level of standardization to Kubernetes infrastructure and reign in the chaos before it gets out of control. In fact, by 2025, 95% of enterprises will fail to scale DevOps initiatives if shared self-service platform approaches are not adopted.[3]

Leading organizations leverage a shared self-service platform for Kubernetes that helps enable and enforce a Standard Operating Model organization-wide. Ultimately, as Gartner[3] states, shared self-service platforms enable platform teams to increase customer satisfaction, achieve economies of scale, and establish high standards of governance.

2

Aligning Stakeholder Requirements

Lack of standardization of K8s workflows and tools has a trickle-down effect to every team involved in the software delivery lifecycle. Because of the complexity of operating K8s, the business can even be affected. New product innovations rely on operational support and developer productivity, so the lack of K8 standards can result in slow and costly innovation for the business.

To reverse the K8s chaos trend described above, platform teams at leading companies apply a shared service approach to K8s. But to be successful in this endeavor, a variety of requirements across groups inside an organization must be met.

REQUIREMENTS

Operations Team

AUTOMATION & EFFICIENCY

Development Team

AUTONOMY & SELF-SERVICE

Platform Team

STANDARDIZATION & CONTROL

Operations and Development Team Requirements

Operations teams need automation and efficiency. They require a centralized place to automatically manage the full lifecycle of clusters between different data centers and cloud accounts. They also need to efficiently grant developers the right access to clusters without adding unnecessary risk.

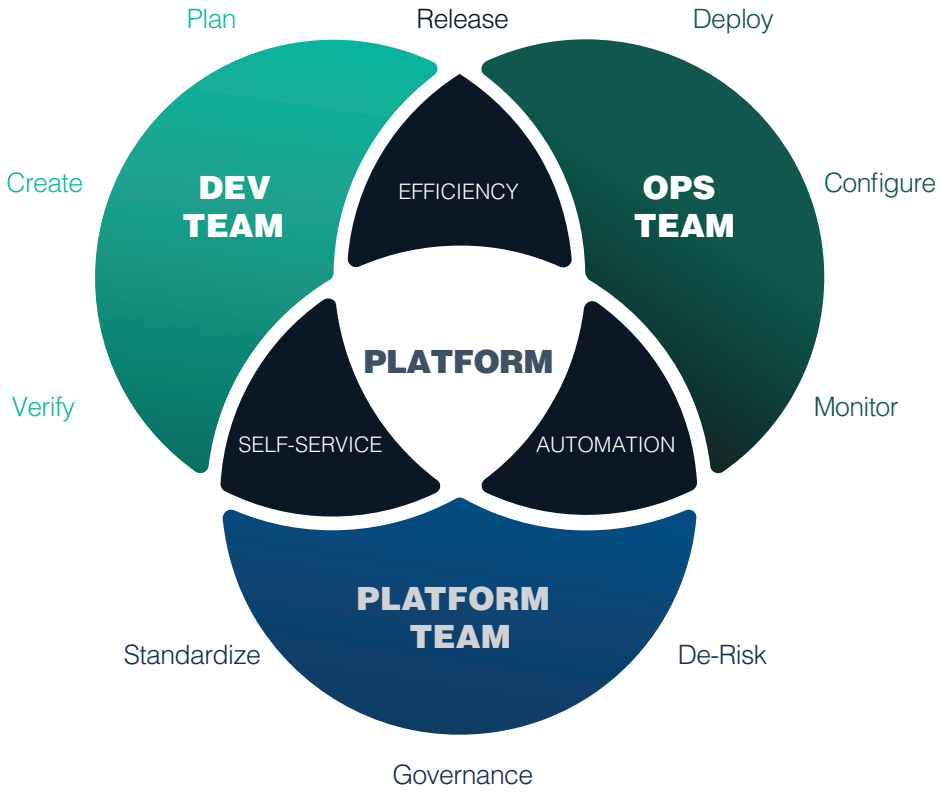
Development teams need autonomy & self-service to deploy applications and fixes quickly. But they are often bottlenecked by operational tasks and ticket-based workflows that slow them down. They need the ability to safely spin up infrastructure at will that has all the requirements for security, policies and governance built-in. They also need to be able to gain visibility into their cluster and application health so that when issues happen they can resolve them quickly.

Platform Team Requirements

Platform teams want to enable development and operations teams to move fast, but with the right level of standardization and guardrails the business requires. Their challenge with K8s is that they often lack the ability to automate, secure, and govern K8s fleet-wide. When platform teams don't have the tools needed to create and enforce a Standard Operating Model for Kubernetes, reliability and support costs, unique configurations, and security risks increase. By providing a central platform, platform teams deliver self-service tools and workflows to development and operations teams that enable a consistent approach to clusters and apps and eliminate skills gaps that hinder the pace of innovation.

Standardize on One Consistent Set of K8s Workflows and Tools

One consistent organization-wide set of tools and workflows for managing the full lifecycle of your Kubernetes clusters is essential for unlocking all the benefits Kubernetes promises.



Rafay simplifies delivering these Kubernetes standards while providing enterprise-wide guardrails. This allows platform teams to provide a shared self-service platform to operations and development teams that gives everyone what they want while still creating a Standard Operating Model for running K8s across the entire organization.

3

Guiding Principles for a Kubernetes Standard Operating Model

Organizations without a platform-centric approach leave developers to carve out and manage their applications' infrastructure in an ad-hoc fashion, risking inconsistencies and self-made policy. Isolated, stranded cluster environments ultimately cause engineering to suffer from misdirection while exacerbating security gaps.

The platform team model fits perfectly with the needs of today's Kubernetes infrastructure. By leveraging a central platform, your organization can provide a consistent approach with shared services enabling rapid innovation without risk.

There are seven guiding principles for establishing a Standard Operating Model for Kubernetes. Each of these requires a platform-centric approach. A platform team that excels at these principles is positioned to make a significant positive impact by eliminating the chaos associated with the growing complexity of managing K8s.

Automation

As the number of modern applications grows, so does the operational burden of managing the lifecycle of Kubernetes clusters and applications. Automating routine manual tasks is critical to helping operations teams operate their growing fleet of Kubernetes clusters. This includes provisioning, upgrading, and deprovisioning clusters from a centralized location as well as ensuring every cluster is running the latest version of Kubernetes and has all the correct software add-ons installed for security, backup, network policy, etc. Your team should be able to manage and seamlessly upgrade your Kubernetes clusters in-place from a single console across all of your on-premises, bare metal, cloud, and edge environments.

Security

Zero-trust security principles protect Kubernetes environments. Security should allow user access through a hierarchical model with fine-grained permissions (ie, support both hard- and soft-tenancy). With the right access for the right roles, separation of duties protects against unauthorized use. Centralized security and auditing creates comprehensive logs of cluster user activity for review and analytics.

Visibility

A centralized view of all the clusters in your infrastructure and all applications keeps every team on the same page so problems can be resolved quickly. Resource usage, consumption, access, and user activity is at-a-glance for a broad perspective of the entire enterprise. Your teams should always be able to review alerts, verify cluster health, and check metrics.

By centralizing you are also able to create different levels of visibility based on specific teams and roles. There should be hierarchy in place where infrastructure leaders can gain aggregated insights across the entire fleet while maybe operations managers only need to view their specific group of clusters and developers their defined namespace.

Cost Management

It's critical that your team has a FinOps process in place to gain visibility into your Kubernetes cost and take action when necessary. Analyzing (showbacks), allocating (chargebacks) and defining cost boundaries for separate teams and shared clusters ensures your organization stays on budget and there aren't any surprise bills from your cloud vendor. Once you have near real-time visibility into the cost for each team, you can you start optimizing your cloud spend which is critical so you can predictably scale Kubernetes usage.

Governance

Maintaining compliance amid growing infrastructure complexity can be challenging, but automating policy and regulatory compliance provides consistency. Templates and reusable blueprints make it easy to customize policy and governance details while preventing non-compliance.

Standardizing and governing clusters with reusable blueprints is essential when scaling K8s. Platform teams need to be able to apply standards across infrastructure and either block or get notified when clusters become non-compliant.

Centralizing governance helps prevent violations and creates standardization across your clusters. Whenever a cluster is noncompliant, the platform team should immediately know (and optionally block) via notifications in the platform. Industry and internal compliance is streamlined, and risk is reduced accordingly.

Flexibility

Platform flexibility means you should avoid limiting your Kubernetes to a single distribution, infrastructure or cloud. Vendor lock-in restricts how your teams use the cloud and containerized workloads. Make sure you're able to use Openshift, Rancher, AKS in Azure, EKS in AWS, and GKE in GCP – whatever your company will need to manage now and in the future. Maintaining your platform's flexibility frees you to match the right vendor and services to the workload.

Self-Service

Standardized, automated shared services help organizations scale team access to resources. A platform team can manage, govern, and secure the shared infrastructure, centralizing access, policies, compliance, and cost management. Everyone is able to access resources appropriate to their needs with a streamlined process that improves the user experience for developers and operations teams.

With a smart foundation, all of your internal teams have the access and resources they need to make the most of Kubernetes. A platform team model provides a centralized, self-service infrastructure and workflows that can grow and change with the organization. In turn, you can control cloud costs and gain the visibility and centralization needed to scale.

4

Designing a Shared Self-Service Platform for Kubernetes

A shared services platform (SSP) allows multiple teams to run applications on a shared infrastructure that is managed, secured, and governed by a central platform team. Typically enterprises that reach a certain scale look to share specialized resources, increase efficiencies, and take advantage of economies of scale wherever possible.

Providing standards across common infrastructure and tooling enables organizations to automate workflows and accelerate delivery and access, whether for all employees or those within specific departments. The goal of an SSP is to provide business efficiency by leveraging a common application or infrastructure that is set up once and subsequently leveraged across a large number of users or departments in the enterprise.

SSP for K8s

ACCESS CONTROL

DEPLOYMENT APPROVALS

NETWORKING POLICIES

COMPLIANCE REQUIREMENTS

COST MANAGEMENT

Applying Shared Services to Kubernetes

The concept of an SSP can also be applied to Kubernetes. In this case, the goal of a shared services platform for Kubernetes is to increase deployment and support velocity via self-service capabilities for developers and operations.

The enterprise's platform team maintains centralized control of, for example, access control, deployment approvals, networking policies, compliance requirements, and cost management.

As described above, another benefit of an SSP for Kubernetes is that it is set up and configured once by a central organization and provides a faster and much-simplified process for all downstream users such as developers and operations professionals.

Critical Capabilities for Stakeholders

Another key and very important aspect of a shared services platform is that it has to be able to provide a flexible and comprehensive set of capabilities for different groups within an organization. For example, platform teams require governance and standardization, while developers, operations, and SRE professionals require automation and self-service. Below is a detailed matrix of these capabilities.

	 **PLATFORM TEAM**	 **OPERATIONS TEAM**	 **DEVELOPMENT TEAM**
Overall Goals	Setup & pre-configure shared services Govern & isolate usage Enable self-service throughout the organization	Efficiently manage the full cluster lifecycle Ensure infrastructure reliability & enforce standards across the fleet for access, visibility, policy management, & backup & restore	Develop, test, & deploy quickly Self-service infrastructure components Access & use with isolation
Cluster Lifecycle Management	Create & maintain pre-approved cluster, app, add-on configs for reuse	Self-service cluster acquisition & usage from pre-approved list	Doesnt have to worry about managing cluster upgrades
Secure Access	Enable downstream configurable access control to clusters & workloads	Connect users & groups to appropriate clusters & workloads	Gets access to the necessary resources without risking other applications
GitOps Pipelines	Create multiple pipelines with defined workflows & approvals for different teams	Utilize one or more pre-approved pipelines for cluster & app deployment	Easy workflows that help quickly spin up & deploy infrastructure that has guardrails built in
Visibility & Monitoring	Enable multi-tier visibility & monitoring	Log in & immediately view the health of clusters & apps based on role, group, etc.	Able to view health of their specific workloads
Policy Management	Define pre-approved cluster policies for reuse	Choose the appropriate policy for clusters	Doesn't have to worry about whether the right policy has been applied
Backup & Restore	Create & maintain pre-approved backup & recovery policies to be used by the org	Select & assign the appropriate backup & restore policy per cluster & app	Does'nt have to worry about whether the correct backup & restore policy has been applied

Top 3 Requirements for a Kubernetes Shared Service Platform

It is crucial to enable a self-service platform that removes the complexity of Kubernetes with a single, easy-to-use platform with built-in automation, security, visibility and governance. This can be achieved with three key requirements for an SSP for Kubernetes.

Unified Platform	A unified platform of services enables centralized control of cluster and application lifecycle, the ability to select any best-in-class K8s distribution, and automated scaling of clusters.
Integrated Services	An SSP requires services to be aware of and integrated with each other, such as policy management, security, and monitoring. For example, a centrally controlled policy must be easily applied (and enforced) via cluster and application templates.
Cloud-based (SaaS)	A cloud-based approach provides accelerated adoption of Kubernetes shared services with guaranteed delivery and SLAs with the lowest possible TCO.

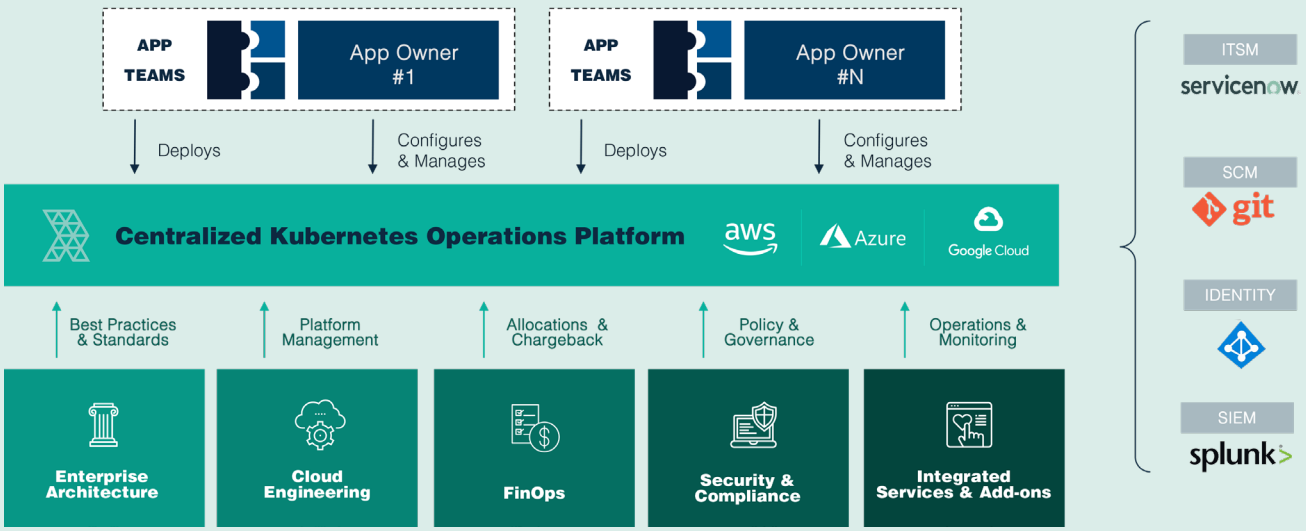
6

How Rafay Delivers a Shared Services Platform for Kubernetes

As enterprises look to accelerate the pace of innovation, building a platform that increases developer productivity, centralizes governance and policy management, and reduces operational overhead can be challenging. Built in the cloud, Rafay's Kubernetes Operations Platform provides deep integrations with Kubernetes distributions such as those from Amazon (EKS), Microsoft (AKS), and RedHat (OpenShift) and delivers operational excellence empowering platform teams to manage, secure, and govern at scale within hours, not months or years.

Many of our customer's platform teams have or are in the process of building an SSP for their organization to manage the fulfillment and operations of all Kubernetes-related services while enabling self-service for the rest of their organization, in particular developers and operations.

Rafay lets you focus on your applications, not on managing and operating Kubernetes.



RAFAY

Best Practices from Leading Enterprises

Leading enterprises are turning their K8s investment into a competitive advantage. By platforming their Kubernetes with Rafay, enterprises unlock benefits across the entire organization.

PLATFORM TEAM

Standard K8s Operating Model

Platform teams create and enforce a Standard Operating Model for the entire organization to follow. The whole organization gains efficiencies by automating recurring tasks as well as creating guardrails that safeguard the business.

OPERATIONS TEAM

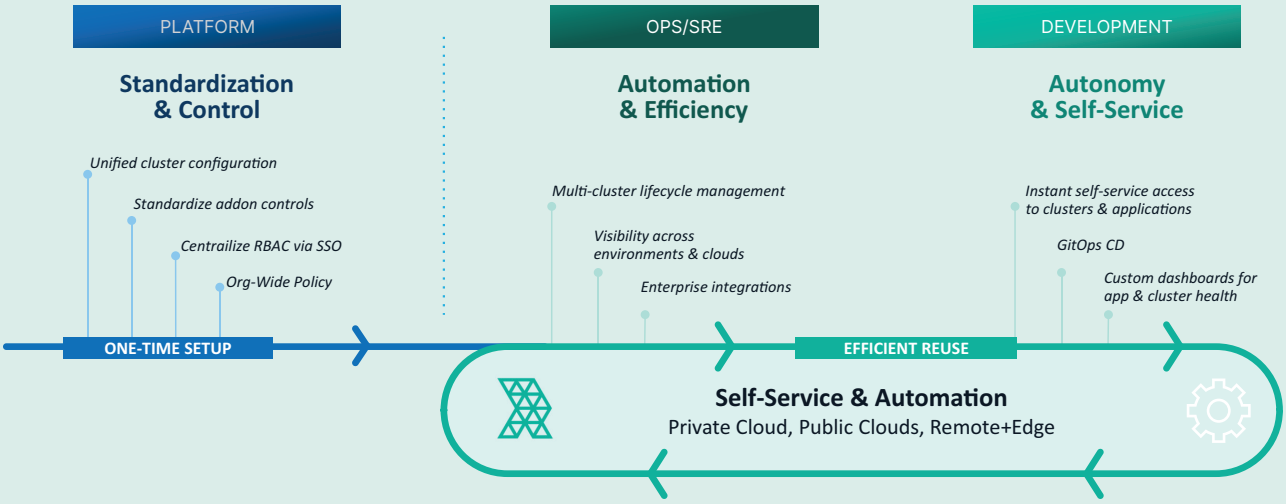
Unified Platform for K8s Lifecycle

Operations centralizes visibility into every cluster they manage, making full lifecycle management easy no matter whether those clusters live in data centers or various cloud accounts. Upgrading becomes seamless across the entire fleet of clusters. They also delegate fine-grained access to stakeholders based on their role by integrating with their identitfy provider (Idp).

DEVELOPMENT TEAM

Self-Service K8s Infrastructure

Developers can safely spin up their own K8s infrastructure as needed removing the back and forth tickets and waiting days for infrastructure to be provisioned. Developers are able to get back to coding and not have to worry about all the intricacies of their applications running on K8s. They also gain visibility into the health of their clusters and applications without having to bug the ops team.



RAFAY

About Rafay Kubernetes Operations Platform

Rafay provides a K8s platform purpose-built for platform teams at companies such as Verizon, MassMutual, and Genentech. With Rafay, platform teams deliver K8s automation and self-service workflows with the guardrails the business needs. Because a shared services approach to Kubernetes provides the standardization and controls enabling rapid innovation without risk.

Sign up for a free trial today or
meet with a Rafay K8s Solution Architect
and streamline Kubernetes Operations with Rafay.

Source:
1 - CNCF: 2021 Cloud Native Survey - Feb 2022
2 - Garnter Report: How to Scale DevOps Workflows in Multicluster Kubernetes Environments
3 - Gartner Report: Why DevOps Success Requires Platform Teams - July 2022

Learn More About Rafay Systems

RAFAY

Rafay Systems, Inc. I 530 Lakeside Dr, Ste 210 Sunnyvale, CA 94085 I 669.247.2551 I rafay.co

© 2022 Rafay Systems, Inc. All rights reserved. All other trademarks and service marks are property of their respective owners.