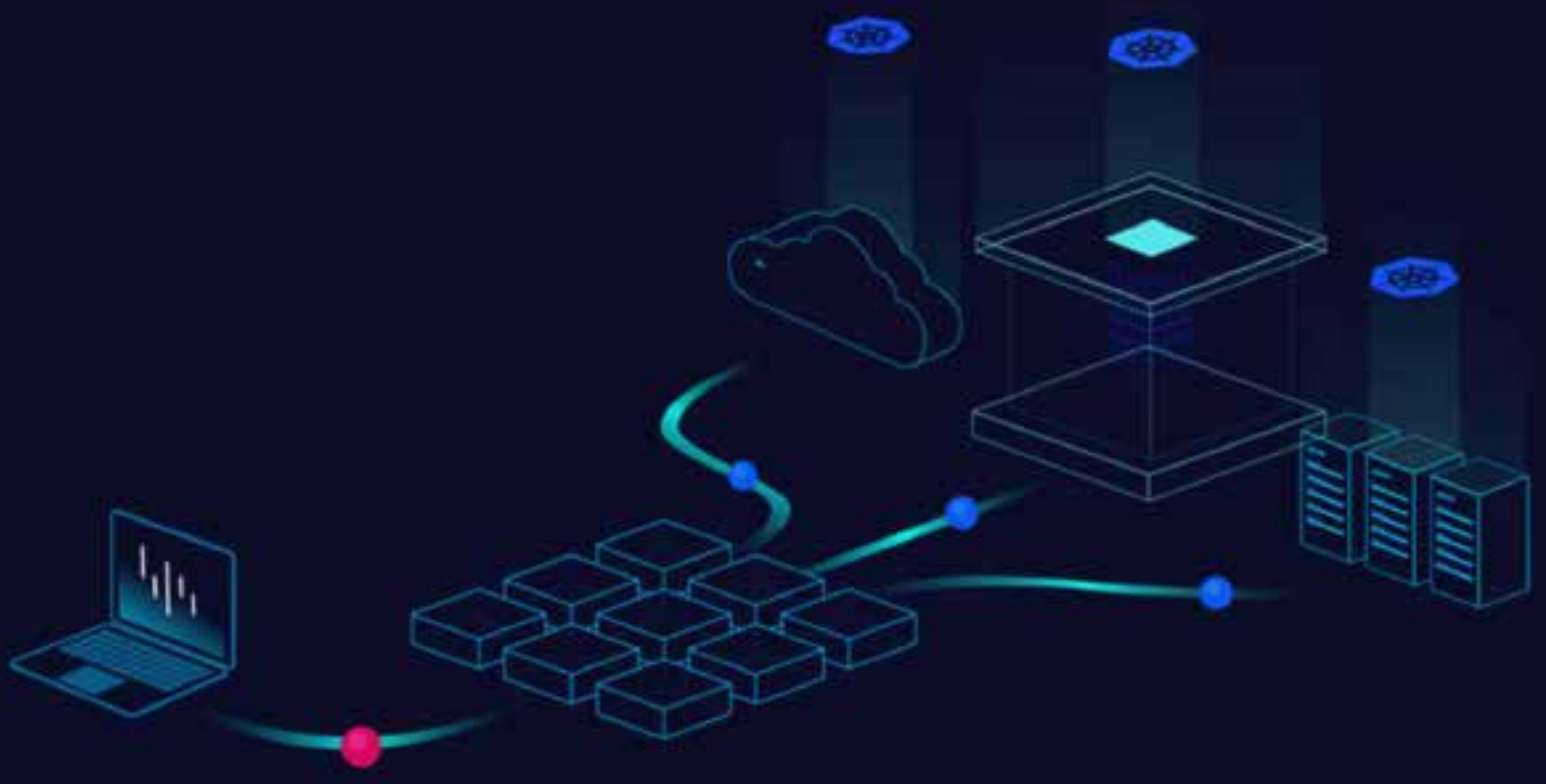
PLATFORM9

White Paper

# Kubernetes in the Public Cloud — Challenges and Solutions



# Introduction

Public clouds are a natural place to start your journey with Kubernetes as they offer robust, convenient access to extensive infrastructure. However, despite the availability of managed services such as EKS, GKE, and AKS, operating Kubernetes in a public cloud isn't easy. That's one of the reasons why 90% of enterprises aren't leveraging the full potential — the scale, speed, and flexibility — of Kubernetes.

In this whitepaper, we outline common operational challenges customers face:

- Handcrafted pet clusters grow and become unmanageable
- Configuring applications: the try-fail loop
- Advanced troubleshooting scenarios
- Cost-optimizing clusters across instance types and flavors
- Finding the talent to scale up Kubernetes

We also show how Platform9 addresses these issues.

# Handcrafted pet clusters grow and become unmanageable

After users start their Kubernetes journey by deploying a cluster with EKS, GKE, or AKS, they tune several configuration parameters such as networking policies, cluster add-ons for ingress, and node pools/groups. This process typically requires numerous iterations and can take weeks or months. As time goes by, users deploy more applications onto the cluster, which means the cluster is now a shared resource that grows organically in the number of nodes, the number of application pods, and the number of end developers who depend on it.

Unfortunately, this can lead to serious management problems:

- **Blast radius** — Any outage can affect multiple teams, significantly risking end-user productivity and/or application uptime.
- **Configuration conflicts** — Different applications are typically better served by different cluster configurations. For example, applications that depend on different Kubernetes versions cannot co-exist on the same cluster. Applications that must scale quickly may be much better served by a dynamic cluster configuration (rapid auto-scaling using nodes that are very quick to spin up). A different application might require a highly robust static cluster running on a small number of beefier nodes that don't auto-scale.
- **Operational constraints** — Large clusters take longer to upgrade, are harder to rebuild in the event of a disaster, and are prone to other operational problems because of their scale.

The solution is to manage multiple clusters via declarative profiles that can be treated as versioned source code. This makes it significantly easier to provision multiple clusters for new applications or teams, to rebuild clusters, or to compare clusters and analyze differences.

> *The Platform9 Profile Engine manages a wide range of declarative profiles and interoperates with relevant open source applications such as the Cluster API project and the ArgoCD framework.*

## Configuring applications: the try-fail loop

To efficiently run an application on Kubernetes, developers have to balance the application's needs, the infrastructure's capabilities, and the cluster configuration. This "impedance matching" involves repeatedly testing application performance against different configurations until it's just right. Unfortunately, this process can be time-consuming (and frustrating), and it's not something that AWS, Microsoft, or Google do for their EKS, GKE, and AKS users without expensive enterprise support plans.

Kubernetes enables numerous feature and configuration options and developers and operators have to learn about the implication of each of these and experiment to understand the trade-offs when using different options. For developers or SREs who haven't already experienced each of these controls and understand the trade-offs, it can be confusing and somewhat overwhelming.

Because of this breadth of configuration options, applications rarely perform to expectations the first time out, and it takes a try-fail loop to get things right. Identifying the configuration setting that's causing a problem takes time. And once identified, it takes time and research to figure out the appropriate value for it. For example, consider configuring pod CPU requests (guaranteed) and limits (ceiling) and the consequences for undersizing or oversizing this value. It's a try-fail loop and many developers new to Kubernetes describe the resulting exhaustion and frustration as "the try-fail fog."

Of course, getting things right in a development environment doesn't mean the configuration will work when it's tried in production.

> *Platform9 automatically detects and reports on known configuration mismatches. Plus, all Platform9 support staff are Certified Kubernetes Administrators who quickly diagnose problems and identify the right path forward.*

## Advanced troubleshooting scenarios

Kubernetes' breadth of functionality also means that there are numerous real-world scenarios that require sophisticated troubleshooting cycles. Here are just a few examples:

- **Node management problems** — Nodes can appear healthy to the Kubernetes control plane when in reality they are severely degraded and are reducing performance.
- **Inadequate cluster auto-scaling** — Auto-scaling is a critical feature because it allows for handling unexpected demands while optimizing cloud costs and spinning down excess cluster capacity. However, the cluster auto-scaler is notorious for several limitations:
  - **Silent failures** — If a new workload doesn't fit in any of the instance types used by existing nodegroups, the autoscaler (rightfully) refuses to scale up. This is a silent failure, and the user is left to troubleshoot why things aren't working as expected.
  - **Slow scale-up due to a lack of pre-warming** — Cluster scale-up can also be a problem if nodes aren't pre-warmed. Cold nodes can take up to 15 minutes to be ready for workloads, which can be problematic when dealing with spikes in load.

    - **Not scaling down well** — This can cause clusters to consume more cloud capacity in unexpected ways.
- **Persistent workload placement** — Clusters that have a high degree of node churn, such as nodes being terminated or added, require careful oversight to ensure that the cluster configuration changes from onboarding and offboarding doesn't disrupt persistent workloads that are interacting with these nodes.

How does a team address these challenges? There is no substitute for Kubernetes expertise, having the experience to anticipate them and design configurations to minimize their impact. Most teams end up scrambling to hire Kubernetes experts, at great expense, to keep up with these problems.

> *Platform9 fully managed KaaS seamlessly imports EKS, AKS, and GKE clusters and works as a partner to eliminate these issues.*

## Cost-optimizing clusters across instance types and flavors

There are hundreds of possible instance flavors in each of the public clouds. In addition, there are also considerations regarding using reserved instance capacity vs. lower-cost spot instance nodes vs. regular, on-demand node configurations.

Each of these specific choices has deeper trade-offs — using spot instances can, in theory, save up to 90% on instance costs. However, the application and cluster topology has to be designed to withstand "spot massacres" where large portions of spot instances in use can be terminated within a very short timespan. If not carefully designed for, this can be catastrophic.

The resultant configuration sprawl is so large, and so difficult to fully understand and evaluate for each application's needs, that most teams get by with some general configuration without maximizing the potential for cost savings on their expensive cloud bills.

> *Platform9 reduced its cost-per-management-plane instance in the cloud by 60% over a 2-year timespan by optimally leveraging the range of instance types and configurations available. All customers have access to these best practices and product capabilities and leverage them to their advantage.*

## Finding the talent to scale up Kubernetes

Most teams get started with Kubernetes in a fairly limited scope. And as they become more adept with it and realize its vast potential, they try to scale up their usage, converting more of their applications to Kubernetes and pushing more clusters to production.

Yet the issues we've outlined in this white paper are real and pervasive, and they don't go away. Kubernetes is extraordinarily powerful, but it's hard. And it takes considerable experience and expertise to operate it cost-effectively in production environments. It should come as no surprise that top Kubernetes engineers and administrators are hard to find and don't come cheap. The best are snatched up by providers like Platform9 and the hyperscalers: AWS, Google Cloud, and Azure.

- A do-it-yourself approach with kops and open source tools works when enterprises are starting to experiment with Kubernetes or when the enterprise is so large and profitable that it can absorb the overhead of a fully staffed DevOps contingent.
- Partially managed EKS, GKE, and AKS work if an enterprise has started their Kubernetes journey, is willing to shell out for enterprise support, and isn't concerned about being locked-in to one vendor for it's infrastructure services. However, with the growing importance and number of multicloud deployments, lock-in is increasingly problematic.
- For a growing number of enterprises, offloading Kubernetes management to fully managed Kubernetes-as-a-service (KaaS) is the easiest, most cost-effective way to quickly deploy their Kubernetes clusters in production environments and minimize Day-2 operational overhead. This kind of service can take advantage of any enterprise infrastructure from a mix of public cloud providers to hybrid edge deployments at thousands of sites.

*Platform9 fully-managed KaaS and on-premises, air-gapped solutions provide the automation, tooling, and SLAs of public cloud Kubernetes services plus the responsiveness of highly skilled, in-house Kubernetes architects and CKAs.*

## Learn more about Platform9

Visit platform9.com for detailed information about our comprehensive solutions for managing Kubernetes deployments of every size on any infrastructure.

### Resources

- Platform9 Managed Kubernetes
- Platform9 Managed Bare Metal
- Platform9 Managed VMs with KubeVirt
- Webinar — kops on AWS is painful at scale but EKS isn't the solution
- Webinar — On EKS but need to go multicloud?
- White Paper — Managing Kubernetes: kops vs. AWS EKS vs. Platform9 KaaS
- White Paper — DIY or Managed? Understanding the True Cost of DIY Kubernetes




Platform9 North America — Mountain View, CA
800 W El Camino Real #180, Mountain View, CA 94040
650-898-7369 • info@platform9.com