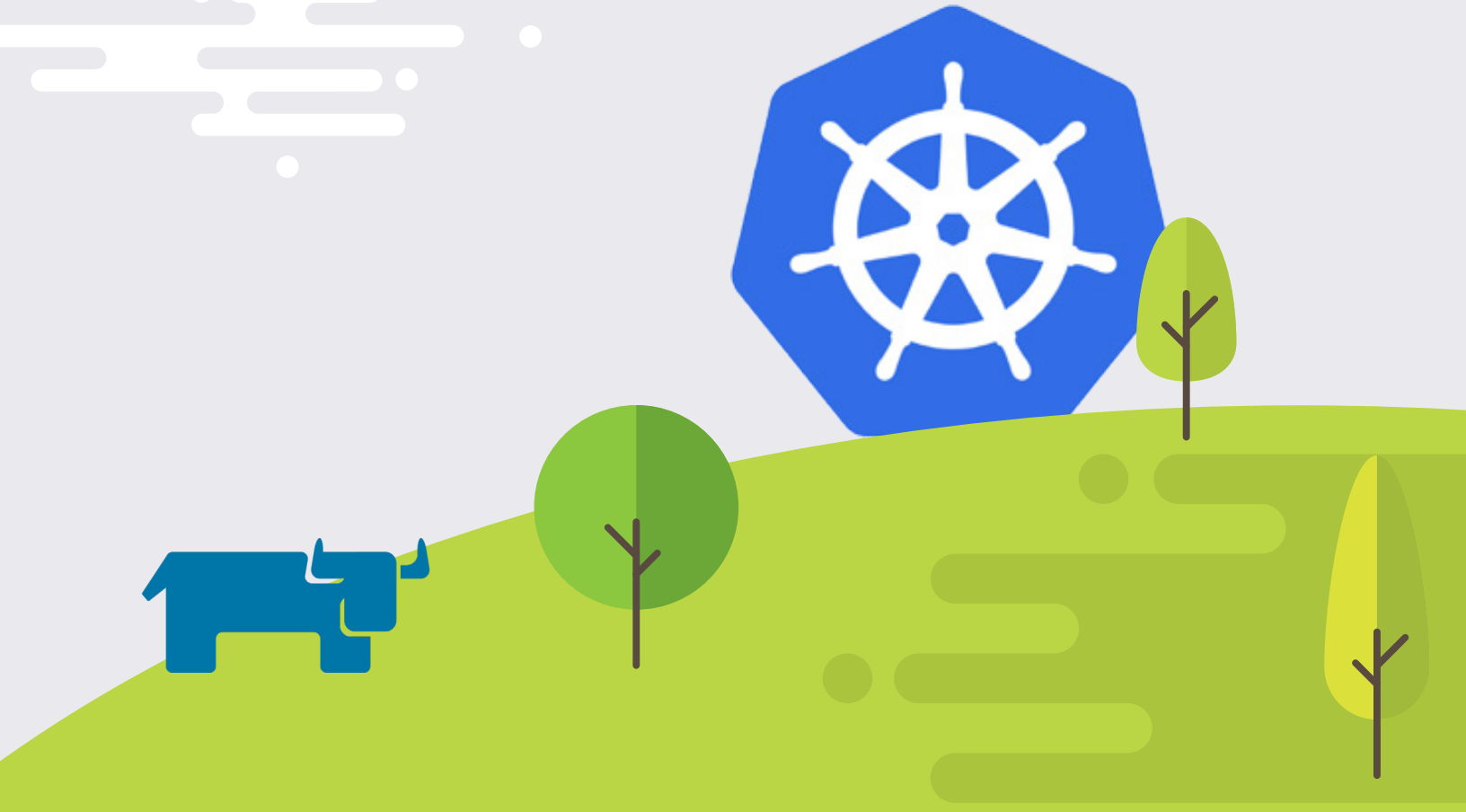
RANCHER

How to Build an Enterprise Kubernetes Strategy



AUTHOR
Shannon Williams
Rancher Co-Founder

How to Build an Enterprise Kubernetes Strategy

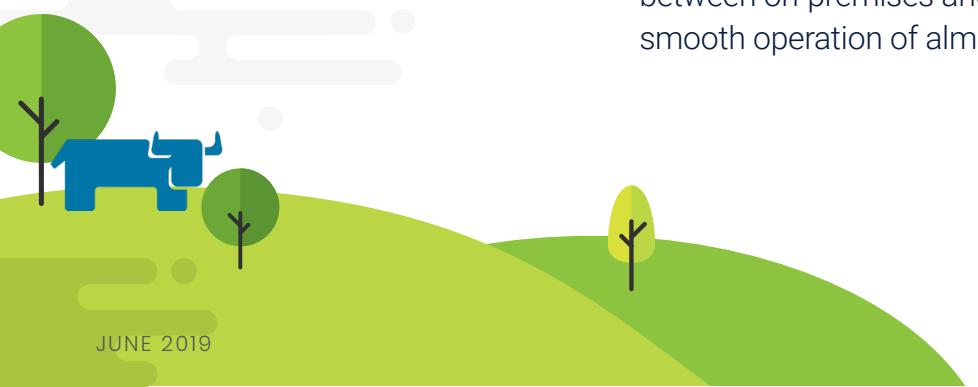
In today's emerging cloud-native environments, Kubernetes is everywhere.

Organizations love Kubernetes because it helps significantly increase the agility and efficiency of their software development teams, enabling them to reduce the time and perils associated with putting new software into production. Information technology operations teams love Kubernetes because it helps boost productivity, reduce costs and risks, and moves organizations closer to achieving their hybrid cloud goals.

Simply put, Kubernetes makes it easier to manage software complexity. As enterprise applications become more complex, development and operations (DevOps) teams need a tool that can orchestrate that complexity. They need a way to launch all the services dependent on these applications, making sure the applications and services are healthy and can connect to one another.

Containers have dramatically risen in popularity because they provide a consistent way to package application components and their dependencies into a single object that can run in any environment. By packaging code and its dependencies into containers, a development team can use standardized units of code as consistent building blocks. The container will run the same way in any environment and can start and terminate quickly, allowing applications to scale to any size.

In fact, development teams are using containers to package entire applications and move them to the cloud without the need to make any code changes. Additionally, containers can make it easier to build workflows for applications that run between on-premises and cloud environments, enabling the smooth operation of almost any hybrid environment.



The Dangers of Too Many Good Things

The problem is that as more containers are deployed throughout organizations and in the cloud, operations teams need a way to keep track of them. Otherwise, it quickly becomes too much of a good thing becoming a bad, or at least an unmanageable, situation. That's where orchestration comes into play.

Kubernetes is an open source container orchestration platform, allowing large numbers of containers to work together in harmony, and reducing operational burdens. In fact, Kubernetes, originally developed by Google and now managed by the Cloud Native Computing Foundation (CNCF), has become a standard for cloud container orchestration, providing a platform for automating deployment, scaling and operations of application containers across multiple clusters of hosts.

There is even an emerging ecosystem growing around Kubernetes as it expands within enterprises. DevOps teams can leverage the incredible tooling that is coming out of the open source software movement, such as new databases, big data tools, artificial intelligence, data analytics, search and many others.

Over the past two years, Kubernetes has moved from development and testing to production environments in many enterprises. According to the 2018 CNCF global survey of the container management marketplace, 40 percent of the respondents are running Kubernetes in production. CNCF received responses from 2,400 developers and IT operations managers as well as personnel from enterprise companies (with 5,000+ employees) worldwide, primarily from North America and Europe.

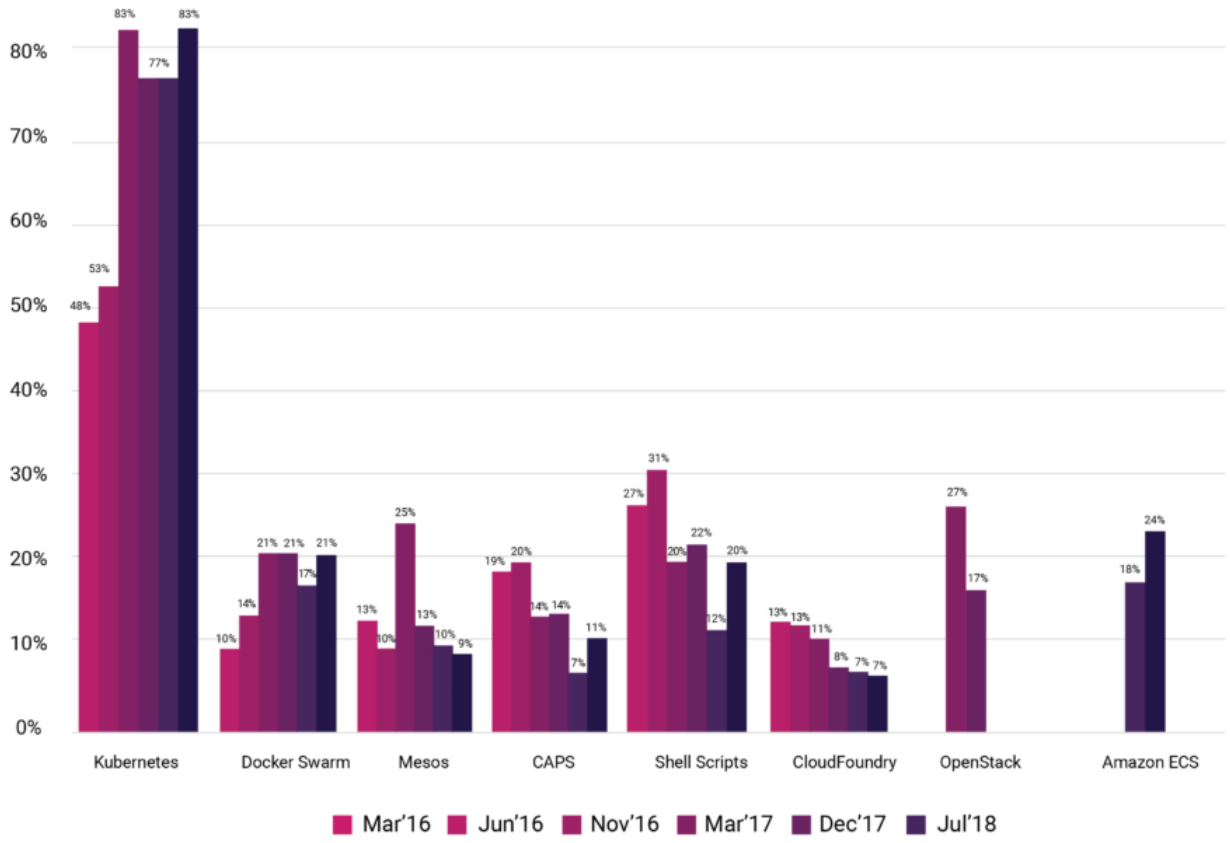


Chart of 2019 CNCF survey results of container management tools

As for container management tools, Kubernetes remains the leader with 83 percent of respondents using it, which was an increase from 77 percent in previous CNCF surveys. Amazon's Elastic Container Service (ECS) was a distant second, with only 24 percent of respondents saying it was in use. And Docker and Shell Scripts brought up the rear, with 21 percent and 20 percent respectively, according to the CNCF.

Kubernetes is not a flash in the pan - it is here to stay - and its prevalence is likely to expand dramatically as software complexity moves to more and more parts of the enterprise.

Understanding Your Organization's Current Kubernetes Adoption

Building an enterprise Kubernetes strategy starts with understanding where Kubernetes is running in your organization, and imagining how it is going to change over the next decade. Over the last two years, access to Kubernetes has gotten dramatically easier. Open source tools make provisioning and upgrading a Kubernetes cluster quick and easy, and cloud providers are now offering Kubernetes as a hosted service. Any team using Amazon Web Services, Google Cloud Platform or Microsoft Azure can provision a Kubernetes cluster in minutes.

- KubeADM
- Kops
- Kubespray
- RancherRKE
- Hand-built

Self-managed K8s

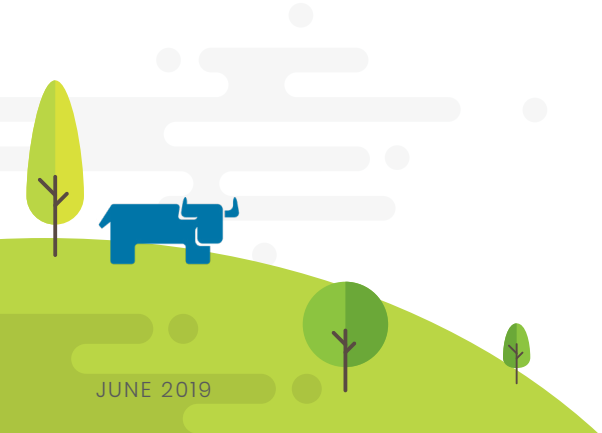
- Google GKE
- Amazon EKS
- Digital Ocean
- Azure AKS
- Many, many more

Hosted K8s

It isn't uncommon for organizations today to approach Kubernetes in the same way they built OpenStack or other shared, centralized services. Teams can use Kubernetes to build large clusters of infrastructure, and then offer development teams shared access to those clusters through Kubernetes namespaces. Using namespaces makes it possible for a cluster administrator to segment cluster resources and define usage quotas and resource limits in order to deliver a reasonably well isolated experience for each team that needs access to Kubernetes.

Other organizations have left it to individual departments or DevOps teams to decide for themselves how and where to use Kubernetes. In these organizations, it isn't uncommon to have dozens of clusters deployed across public clouds and company data centers.

Over time, it is possible for tension to develop between individual teams wanting to run Kubernetes in exactly the way they need it, and an IT organization that wants to maintain security and control over how Kubernetes gets implemented.



The incentive for the development teams is flexibility: having cluster level administrative control allows them to configure the cluster to run exactly how they need it in terms of storage, security policy or which infrastructure it runs on. IT teams are especially nervous about clusters that are deployed and left unpatched and unmanaged. They would like to centralize the operations and policy around clusters, and provide access to teams who need it.

If Kubernetes and containers are going to become the primary platform for running applications across any infrastructure, IT managers must collaborate with DevOps to develop a plan and a strategy for Kubernetes that satisfies the needs of the development organization, and meets IT's own needs, as well.

As you document and understand where Kubernetes is running in your organization, be on the lookout for individuals who show existing expertise around containerization. As you progress in building your strategy, developing a team of experts who can help with both administration of Kubernetes clusters, and deployment of applications on Kubernetes will be critical to driving adoption.

Where Will You Be Running Kubernetes in Five Years?

Building an organization-wide Kubernetes strategy means prioritizing your goals for this new technology. If your team sets out to use Kubernetes and containers as a way to reduce infrastructure costs, you'll probably focus on building big clusters and trying to get as much density as possible out of them. If, instead, your team focuses on using Kubernetes to accelerate development and support teams look for continuous delivery across different computing platforms, you'll take a different approach emphasizing flexibility and delivering more tooling around Kubernetes, such as monitoring and CI/CD integration.

To prioritize your goals, try to understand the potential of Kubernetes, and imagine how your company might be using it in five years. Kubernetes is a great way to run modern, microservice-centric applications. It offers a rich set of functionality that allows teams to determine how different services within modern applications are run, handle unexpected events, connect with each other, and connect with other applications and APIs.

Today, every major cloud provider has made it easy to deploy Kubernetes clusters within minutes. Teams are continuously building new applications, deploying them to different clouds and using Kubernetes to run them. Between clusters used for development, staging and production, and the need to deploy Kubernetes clusters across different data centers and cloud providers, it isn't hard to imagine even a well-organized company running dozens of Kubernetes clusters.

What's interesting is that the same modern application architectures that we think of as cloud-native are now beginning to move out of the data center. Teams building software for factories, hospitals, and stores now want to run applications with rich data analytics and complex architectures as close to their customers and production facilities as possible. Today we think of Kubernetes as a tool for running data center and cloud workloads, but within a few years, we will very likely be using it to run applications across any infrastructure. Even single-node devices such as point-of-sale terminals, medical devices, communication equipment, or cars will benefit from the ability to easily deploy and run applications using microservices. We could be looking at thousands of edge deployments, all running as individual Kubernetes clusters, and presenting an API that needs to be managed.

Between clusters running in different clouds, data centers, and the edge, it is almost certain that your organization will be running more than one Kubernetes cluster. Unless you know you'll only be running a single application in one location, it probably makes sense for most teams to build their Kubernetes strategy with an expectation that they will need to be able to easily provision and manage many Kubernetes clusters running in many different places.

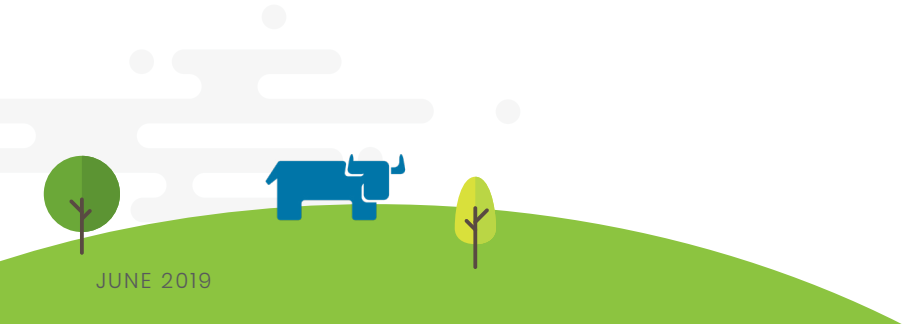
New technologies like Kubernetes are exciting to work with and it isn't uncommon for many teams to try to take ownership of building a containerization and Kubernetes strategy for their company. It isn't uncommon for individual DevOps teams, shared services groups, central IT, cloud platform, or platform-as-a-service groups to feel that they should be responsible for building a strategy around Kubernetes.

Who Should Own the Kubernetes Strategy?

As always, there isn't one correct answer for determining the team who should own your strategy. Successful teams often bring together talent from across the organization, and collaborate to determine requirements. Still, investing in a strategy and building a platform means finding budget, so it is definitely most common that one team takes the lead on delivering on the strategy. Two teams we often see leading the container strategy are the shared services team responsible for supporting developers and DevOps and the central IT function responsible for computing platforms.

The shared services team brings key insights on how an organization is modernizing its approach to application development, and the requirements teams have identified for what they need in a Kubernetes platform. They often understand other key systems that have been built for DevOps such as CI/CD tools, development environments, data services, and application monitoring tools. Whether these teams own the strategy or simply contribute to it, they represent at the very least one of the primary consumers of containers in the organization and should be a critical part of developing your organization's strategy.

The central IT team focused on cloud computing and other computing platforms is also a logical team to lead a Kubernetes strategy. These teams have a strong understanding of platform operations, infrastructure, security, multi-tenancy and existing IT investments, and usually have significant experience running critical projects. A project led by the IT platforms team, will definitely benefit from their understanding of the broad requirements of many different teams across a large, complex organization. Still, projects coming out of central IT often suffer from too little engagement with end users and too much influence from existing technology vendors. These teams often have very little experience with the latest application architectures and benefit enormously from working closely with teams leading innovation around application development.

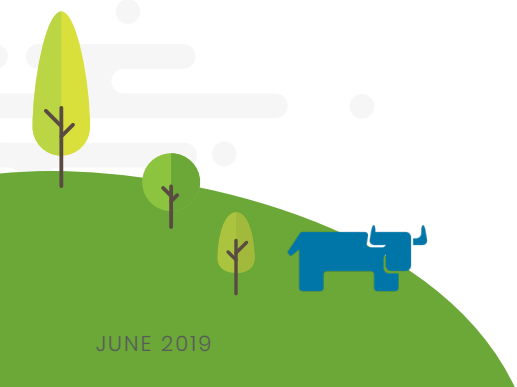


Centralized vs Decentralized Kubernetes Management

Regardless of who drives the strategy, one of the key questions that will emerge is around how much standardization is possible without impacting the innovation benefit that is a central goal of the IT platform. Many teams will have experienced projects around OpenStack and Platform-as-a-Service that struggled to get adoption because users were not able to get enough flexibility to deploy the next generation applications they were building. With Kubernetes, there is enough flexibility in the platform and the ecosystem to satisfy any team. Exposing that flexibility is critical to delivering value. Any strategy that abstracts away Kubernetes, will probably face resistance from your most innovative teams. At the same time, the flexibility of Kubernetes and its ecosystem can be a hinderance to some teams looking for a platform to just run standard apps.

One of the most exciting developments in the Kubernetes space in 2019 is the emergence of lightweight abstractions or "MicroPaaS" software. These are projects that run on Kubernetes, but provide frameworks that simplify application management. At Rancher we've developed a MicroPaaS called Rio that incorporates Kubernetes, Prometheus, Knative, and Istio to deliver a simple, integrated experience that helps teams quickly jump into Kubernetes and realize a ton of value. Google Run is another example of this approach to running containers that abstracts away some of the complexity of Kubernetes. These approaches take lessons from the serverless space, allowing containers to scale to zero for example, and provide simple declarative languages to build, connect, scale, and monitor services. For teams using CI/CD and stateless applications, they can deliver a powerful experience without requiring teams to develop a deep understanding of all the underlying technology.

As you build your Kubernetes strategy, consider blending the best of a decentralized approach with enough controls and management to ensure compliance and remove repetitive tasks. Try to centralize and automate common tasks such as Kubernetes cluster lifecycle management, RBAC policies, infrastructure management, and other "day two" operations. At the same time, leave teams with lots of options for where they can get access to Kubernetes clusters, and whether they can use a shared cluster or a dedicated cluster. You should focus primarily on maintaining visibility into all of the clusters that are provisioned, not necessarily forcing teams to utilize a set of pre-approved clusters in a specified way.



Containerization and Kubernetes Will Disrupt Some of Your Other Plans

If your organization has decided that expanded adoption of Kubernetes and containers will accelerate innovation and is critical to your IT strategy, it's important to consider how it will impact other projects that are already happening. Below, we'll look at a few common projects that might be impacted by Kubernetes. This is by no means an exhaustive list. Be sure to consider your own organization's existing projects and how your container strategy might augment or impact them.

Our Organization is Heavily Investing in Cloud Computing

For organizations that are focusing on a "cloud-first" IT strategy, Kubernetes adoption will almost certainly be part of a larger cloud strategy. Every large cloud provider offers hosted Kubernetes clusters, and often other types of container-oriented services such as registries, monitoring, CI/CD, and platform services. A key question for these organizations is whether they should build a strategy for Kubernetes independent of the cloud computing strategy. If an organization has chosen to commit to one cloud provider, it is likely they will defer to their cloud provider's strategy around containers. For example, if an organization runs or plans to run most of their critical applications in Azure, it would make a lot of sense for them to have a deep familiarity with the different containerization services in Azure. Obviously, they may find that these services don't meet their needs for some reason, but even if they implement their own Kubernetes in Azure, they will want to implement it in a way that takes advantage of other Azure services.

On the other hand, if an organization has decided to focus on multi-cloud as a strategic initiative it is likely that they will see Kubernetes as an opportunity to unify how they interact with all of their cloud providers. There are two competing theories for the best way to build a multi-cloud strategy on Kubernetes:

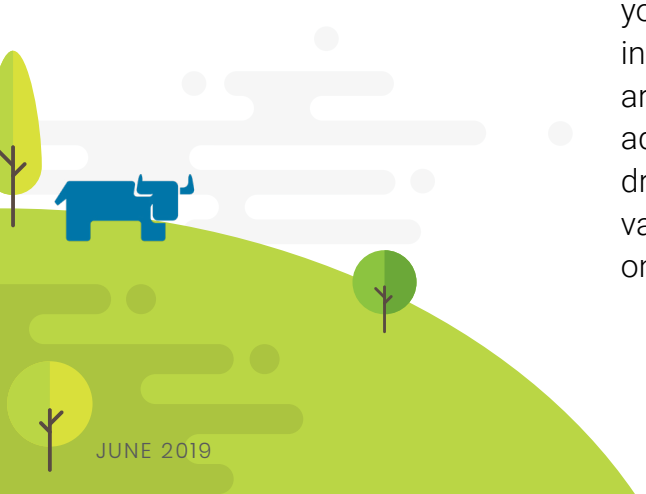
- The first suggests that you should use the cloud providers only for core infrastructure provisioning, and build a consistent platform based on Kubernetes on top of this infrastructure. With this approach, teams would develop a consistent implementation of Kubernetes and any of its dependent services, and then build a common platform on top of the cloud infrastructure. This approach is aimed at minimizing cloud lock-in, and achieving broad application portability.

These teams will try not to utilize any of the proprietary services that different cloud providers offer, instead opting for open source, or multi-cloud solutions. The large platform software companies, often recommend this approach, suggesting that using their PaaS platforms across different clouds can alleviate cloud lock-in.

- The second approach suggests that teams standardize policy and management around Kubernetes, but assume that wherever they run Kubernetes, their developers will probably want to utilize other services that might be unique to that computing environment. This approach suggests that if you are running Kubernetes in AWS, you shouldn't hesitate to use other services that might be unique to AWS. These teams worry less about lock-in and more about giving application teams the flexibility to use the native capabilities of different platforms. With this approach, the focus needs to be on providing common management and tooling around different implementations of Kubernetes.

We Are Investing in Hyper-Converged Infrastructure as Part of a Data Center Upgrade

For teams that are building new datacenter capacity using hyper-converged infrastructure, Kubernetes will almost certainly be a workload on these platforms. Deploying and operating Kubernetes on hyper-converged infrastructure isn't any more complicated than running it anywhere else. However, you may find that the hyper-converged infrastructure provider offers a Kubernetes implementation as part of their platform. With these services, integrating them into a larger Kubernetes strategy could offer additional value. Alternately, because most hyper-converged infrastructure platforms offer APIs for provisioning of hosts or VMs, it might make sense to incorporate the control plane into your Kubernetes management layer, to enable auto-scaling of infrastructure as cluster sizes go up and down. Storage is another area where integrating with hyper-converged infrastructure can add a lot of value. Many of these platforms have Kubernetes drivers that simplify volume creation, and can provide additional value around backup and recovery of stateful workloads running on your Kubernetes platform.



We Are Trying to Modernize Our Existing Applications to Improve Security and Stability

A company shared that they had more than 5,000 existing applications that they were responsible for delivering, and thought containerization and Kubernetes might be a good solution for improving how they manage these applications. If you are building a Kubernetes strategy, at some point, you will need to decide how your strategy applies to existing, legacy workloads. Most of these applications are stateful, with loads of complex dependencies and hard coded connections to other services. They don't look like the cloud-native applications that Kubernetes was built for, and yet it is certainly possible to containerize them and run them on a Kubernetes cluster. A recommended option is that teams postpone diving too deeply into legacy applications until they have already been using Kubernetes for new workloads.

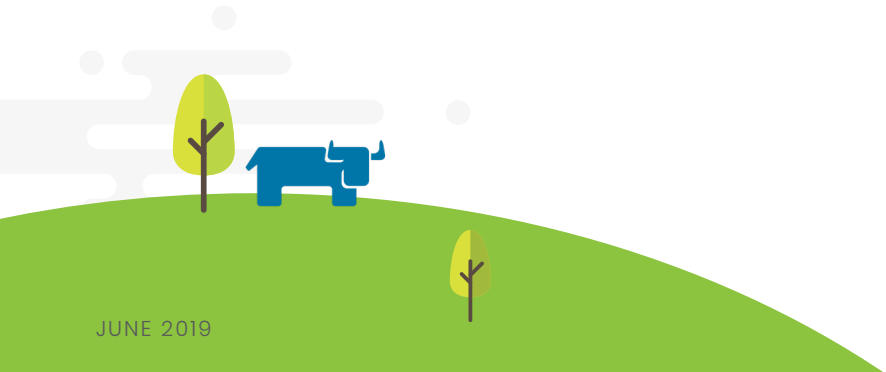
When your familiarity with Kubernetes expands and your team is able to manage multiple production clusters running stateless, cloud-native applications, that is a good time to start looking at running legacy applications in containers. An entire paper could be written on best-practices for migrating legacy applications to Kubernetes, but the bottom line is that it almost always makes sense to run these applications in dedicated clusters with different approaches to infrastructure management. These applications are architected with an expectation of stability and infrequent failure scenarios. You can mimic that in Kubernetes and still get a lot of the other benefits Kubernetes offers, such as security, support of the latest operating systems, automation, and great monitoring and visibility.

We Need to Cut Our Infrastructure/Cloud Spending

If your organization is actively trying to cut costs, the potential of using Kubernetes to improve density can be pretty appealing. Kubernetes clusters offer multi-tenancy and it isn't unreasonable to expect that you can get more bang for your infrastructure spend using containers and better resource scheduling.

However, try not to orient too much of your business case for containerization toward cost savings. Most organizations will take years to migrate a significant portion of their existing application footprint to containers and Kubernetes.

Most of this time will be spent figuring out the right strategy for each application, specifically whether to replace, rearchitect, or just migrate it. Kubernetes certainly can help you get great infrastructure utilization, but it will take time, and the strategy you are developing now is more likely to impact your organization by enabling rapid innovation than by cutting infrastructure spend.



Preparing Your Teams for Broader Kubernetes Adoption

A critical part of any Kubernetes strategy is determining how you'll train your teams to leverage Kubernetes. As we said earlier, if you find that your organization already has some staff members with expertise in containers or Kubernetes, consider how you can incorporate them into your initiative. This doesn't mean necessarily pulling them off their existing work, but perhaps they can work as part of the team setting requirements, evaluating tools, or developing policies.

Regardless of the level of skills your team has, you'll almost certainly have team members who need to be trained on either using Kubernetes or administering it. Luckily, there is no shortage of Kubernetes training providers and online courses. The best ones are oriented around two key certifications available from the CNCF for Kubernetes. The Certified Kubernetes Administrator (CKA) program is focused on individuals who will be managing Kubernetes clusters, and the Certified Kubernetes Application Developer helps teams understand how to build and run applications on Kubernetes.

As you build your core team of early Kubernetes admins and users, consider setting a goal to train and certify as many members of your team as possible. The tests are quite rigorous and will ensure you build strong internal knowledge about using containers and Kubernetes. Once you have some initial expertise, you may want to wait to do further training until you're out of the design phase of your strategy and are beginning to onboard more teams onto the specific implementations of Kubernetes your organization is adopting.

Evaluating Container Management Platforms and Delivering Kubernetes as-a-Service

Up to this point, we have talked about building an enterprise Kubernetes strategy based on understanding how your DevOps teams will be using Kubernetes over the next few years and orienting your platform to support these teams. We have talked about the importance of maintaining flexibility while still providing centralized controls and management. It is at this point, that teams begin to look into technical options for managing containers across the organization.

Analyst firms like Gartner and Forrester describe this class of software as a Container Management Platform (CMP). It is used to describe tools like RedHat OpenShift, Docker Enterprise, and Rancher that provide a platform to manage an organization's containers. In this document, we won't try to compare these different offerings, but we do want to highlight some capabilities you should be considering when determining how they can help you implement your Kubernetes strategy.

Kubernetes Distribution, Cluster Provisioning and Lifecycle Management

Most products in this space will include a standard Kubernetes distribution and should be able to provision a cluster and support your team in upgrading it to the latest version of Kubernetes. Some will also include integrated monitoring, etcd backup and recovery, and infrastructure provisioning and auto-scaling. If you will be using a Kubernetes distribution provided by your CMP vendor, it is important to consider whether that distribution is consistent with upstream Kubernetes or modified in some way that might make it slow to support the latest features being developed in the community.

Multi-Cluster Kubernetes Management

If you expect to manage multiple Kubernetes clusters at the time of launch or in the future, it is worth reviewing how a CMP approaches multi-cluster management. Look at how the CMP manages multiple clusters, and whether it can manage different types of clusters including cloud-based Kubernetes services. Does the platform only manage clusters it deploys, or can you import existing clusters that may already exist? Most importantly, what does "management" of these different clusters mean? What type of actions can you take across these different applications?

Visibility is great, but you will also want to implement policy and controls, automate operations, provide application catalogs, and possibly offer other shared services. If multi-cluster management is key to your strategy, be sure that you understand what it means and how different CMP platforms implement it.

User Management and Delegated Administration

The core purpose of any Kubernetes platform is to provide a shared service to your users that makes it easy for them to innovate. Your evaluation of these tools should be oriented toward understanding how you will manage a large number of users, and the experience you can deliver to them. Understanding the user journey starts with defining how users will access Kubernetes. Ensure your platform supports your existing single-sign-on such as LDAP or Microsoft Active Directory. You'll want the ability to authorize both individuals and teams to access specific clusters or name spaces, and the ability to define a wide variety of roles that fit your business requirements.

Once you have ensured your platform supports the necessary access control capabilities, consider what administrative capabilities you can delegate to team leads, cluster owners, and project owners. Does the platform allow you to dedicate resources to specific teams? Can you easily define resource quotas and manage utilization of shared platforms? Can teams collaborate on projects and share application catalogs? However you decide to deliver Kubernetes to these different teams, be sure you are providing direct access to the Kubernetes API and kubectl, as this will ensure they can access all of the features of Kubernetes.

Policy Management

Building a central policy management layer is how you can ensure compliance and adequate controls across all of your organization's implementations of Kubernetes. Most CMPs will have administrator control planes that allow your teams to define policies and apply them to all of the teams using Kubernetes and all the clusters in the organization.

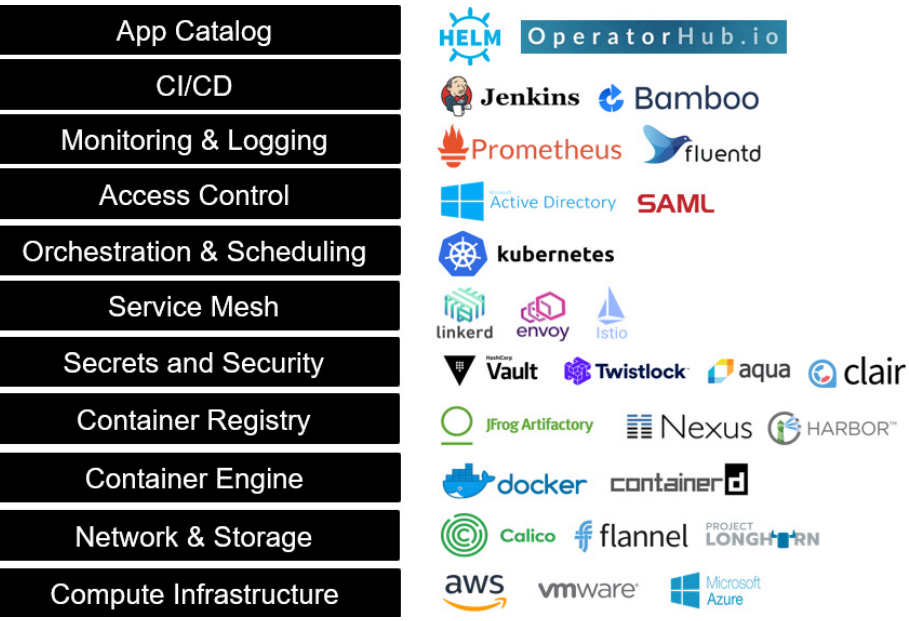
For example, the Kubernetes Pod Security Policy is a cluster-level resource that controls security sensitive aspects of the pod specification. The Pod Security Policy objects define a set of conditions that a pod must run with in order to be accepted into the system, as well as defaults for the related fields. They allow an administrator to control functions such as running of privileged containers, usage of host namespaces and usage of host networking and ports, to name a few. Policy management can also address container image scanning, cluster configuration, and even application deployment. For instance, if your organization decides to implement a container security product like Twistlock, Aqua, or NeuVector, policy management should allow you to ensure that these applications are automatically installed on any new or imported Kubernetes cluster.

User Experience and the Entire Cloud Native Stack

Kubernetes is a powerful engine, with a rich ecosystem of tools around it. Most CMP platforms will provide a full user experience around Kubernetes that incorporates ecosystem tools and delivers a user interface to simplify workload management. As you evaluate these platforms, consider how they have approached integrating adjacent technologies, such as the container engine, overlay networking, automation tooling, container registries, service mesh, monitoring, logging, CI/CD, and application catalogs. Are these tools tightly or loosely coupled with the platform, and how does that impact the flexibility your teams will have to implement new approaches as they develop?

One of the biggest risks of a CMP is that it puts too much emphasis on integrated solutions and ease of use, and ends up limiting flexibility. Kubernetes is very well architected for plug-and-play integration with most of its ecosystem, be sure not to lose that flexibility. For example, if a platform offers an integrated CI/CD, make sure that your teams that already have CI/CD tooling can easily connect their existing process to it as well.

CMP software should address the entire cloud native stack, either through embedded capabilities or third party integrations



Kubernetes Security and Audit

As you consider CMP tools, it is critical that you collaborate with your security experts and ensure the tools will support your broader security requirements. Most CMP offerings address security and auditing at a global level, and can provide you with a good understanding of their approach. At the platform level, some of the most important capabilities to consider include role-based access control, centralized security policies, container image scanning, and the ability to quickly patch Kubernetes and the container runtime (Docker, ContainerD or Cri-o). Some platforms will go so far as to evaluate your clusters against CIS (Center for Internet Security) benchmarks for Kubernetes Security.

In addition to the platform level security, there is a rich ecosystem of organizations that focus on container security specifically, such as Aqua Security, Twistlock, StackRox, and NeuVector. These tools provide different security capabilities on top of the capabilities of Kubernetes or a CMP and are worth evaluating as part of your broader implementation of Kubernetes.

Open Source, SaaS, and Support

If you decide to roll out a CMP, one of the key decisions you'll have to make is how to operate the platform. Most CMP tools today are delivered as software, and many, such as Rancher, Mesosphere DC/OS, and OpenShift are available for free as open source software. Because of this, it should be easy for your team to deploy and evaluate multiple technologies in this space. All of the open source tools offer some option to get enterprise-grade support. As you're looking at these options, consider whether that support requires you to move from the open-source version to an "enterprise edition" that may add features, but can also make it difficult later to move back to the open source version if you don't want to pay any more. When you're evaluating the support offered by these CMP tools, consider how much of the stack is supported. Are you getting support for Kubernetes, what about the container runtime, service mesh, networking implementation, and other cloud native stack components, such as Istio, Prometheus, and Helm? Does the vendor provide root-cause analysis across all of these elements of the stack?

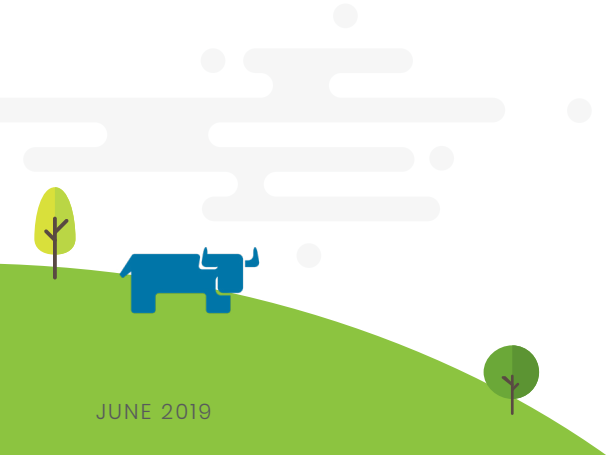
If you don't want to operate one of these platforms yourself, some vendors offer cloud-based or managed versions of these CMPs. If you decide to move to a hosted CMP, consider how much flexibility you'll have in the future to move off of it as your needs change. Is it a shared implementation of the CMP, or is it dedicated to your organization? You're going to build lots of policies, templates, best practices, and integrations with a CMP, so make sure that there is some way for you to extract that logic and move it to a different platform if your requirements change in the future.

A Few Final Thoughts

Adopting a new technology across a large organization is never easy. As technologists, we get excited when new approaches emerge that have the potential to create new, amazing experiences for our customers. Many of us who have been working in technology for the last 20 years, see Kubernetes and containerization as the third phase in a process that started with the emergence of visualization and expanded with cloud computing. As you build an enterprise containerization strategy, be sure to learn from your organization's past successes and failures in adopting these other technologies. If you have team members who were instrumental in rolling out VMware or AWS in your organization, incorporate them into this project and see what insights they can provide that are specific to your organization.

We talked earlier about how to determine who should lead any enterprise-wide strategy around containers. As you develop and implement your Kubernetes strategy, pay special attention to the teams who are already running apps on Kubernetes. Each of these teams should be represented on your strategy team and should be validating that your approach to managing Kubernetes will not introduce constraints that would keep them from adopting it. Focusing on the early adopters will help you avoid over-simplification and delivering a platform that deviates from mainstream Kubernetes adoption.

As you set off on this journey, pay special attention to learning from other organizations who are adopting Kubernetes. Every year, the presentations from KubeCon are recorded and posted to YouTube. You can find a wealth of real-world advice from teams who have gone through rolling out Kubernetes at either a project or company-wide level. The best of these presentations will focus on not just how a company is using Kubernetes, but also on the challenges they ran into while adopting it, and the impact it has had on their business.



Appendix A - Case Study

How Life Sciences Leader Illumina Implemented an Enterprise Kubernetes Strategy

Illumina, a life sciences company and a market leader in genetic sequencing, deployed Rancher with the goal of enabling teams to create containers on any infrastructure while maintaining a strict IT and security policy. The ground-breaking research that comes out of Illumina's offices drives work on disease, drug reaction, agriculture, and much more. The Kubernetes platform deployed with Rancher now supports every facet of that research.
At Illumina, massive amounts of data are a critical part of the equation. The company was eager to find ways to move data through a pipeline and perform copious amounts of analytics, while also managing all the administrative tasks that come with running an organization at scale.

A Complex Puzzle with Many Parts

One of the biggest complications Illumina faced is having an environment where custom-compiled codebases were not accessible or fully organized for company and employee use. The container implementation was a challenge: in trying to migrate processes to a new container system, the company found that using a container model took quite a bit of research and staff-hours.

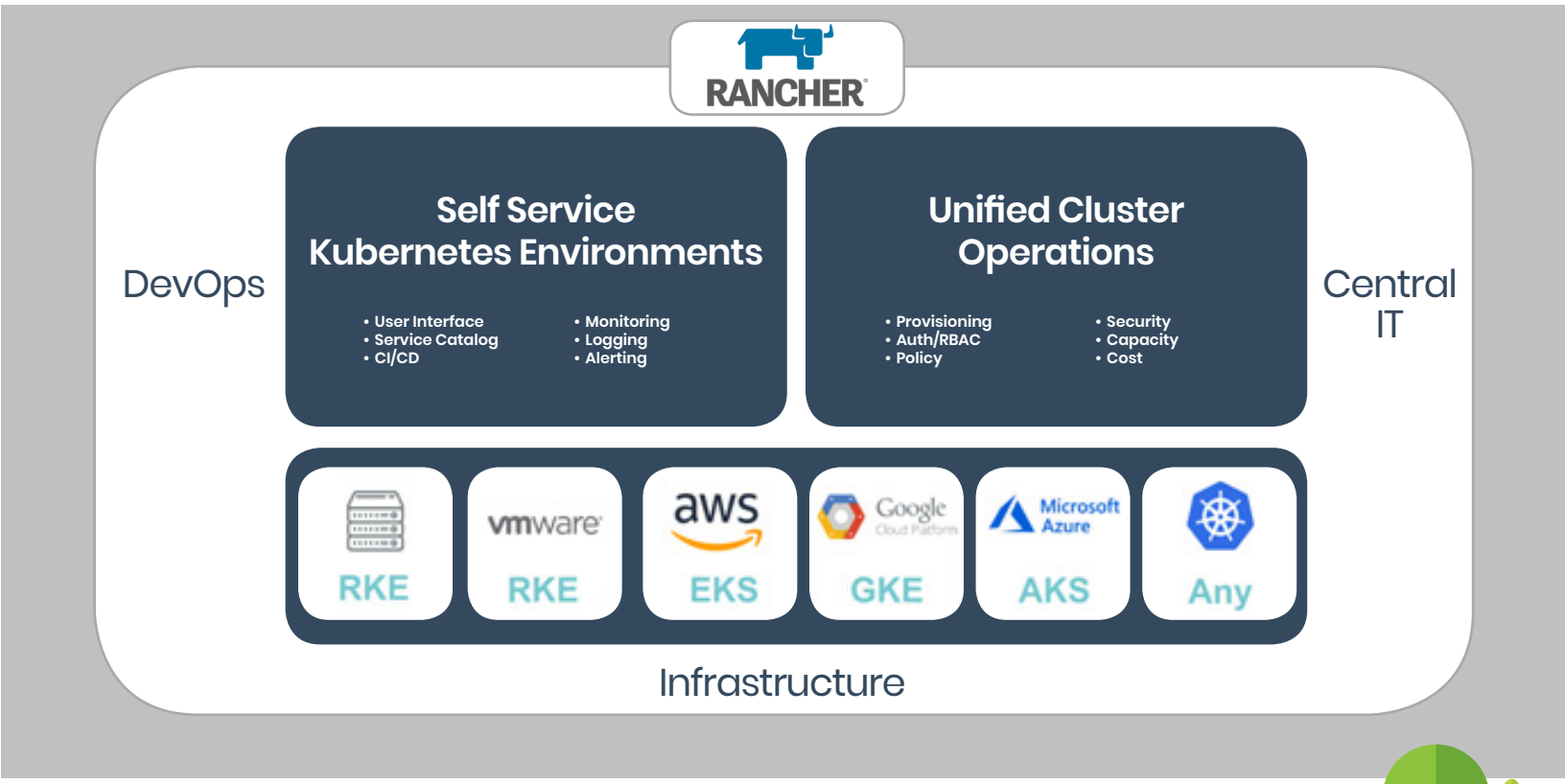
Another component was the efficiency of management:
10 people had to support 300 developers and petabytes of data. Illumina leaders focused on trying to enhance automation and create a better environment in which to run all of their applications. The overall architecture had to support containers running on vendor systems like AWS, as well as bare metal on-premises deployments.

Connecting All the Pieces with Rancher and Kubernetes

The company chose Rancher as their Kubernetes platform to accelerate their DevOps processes. The team at Illumina worked to build a Rancher-based system that can function in different environments. For example, they used AWS where it made sense, and put other components on internal systems or on-premise hardware for better cost efficiency and control.

Illumina has also been able to use Rancher's functionality to push new goals in deep learning: something that will, in turn, enhance all company operations. Using Kubernetes as the orchestration layer, teams deployed their first two production machine learning environments with Rancher, where managers can schedule against GPUs for maximum efficiency.

The automation department also uses Rancher orchestrated container models to host persistent microservices related to HTML messaging, identity management, and service desk creation.



How Rancher and Kubernetes Can Work for Any Organization

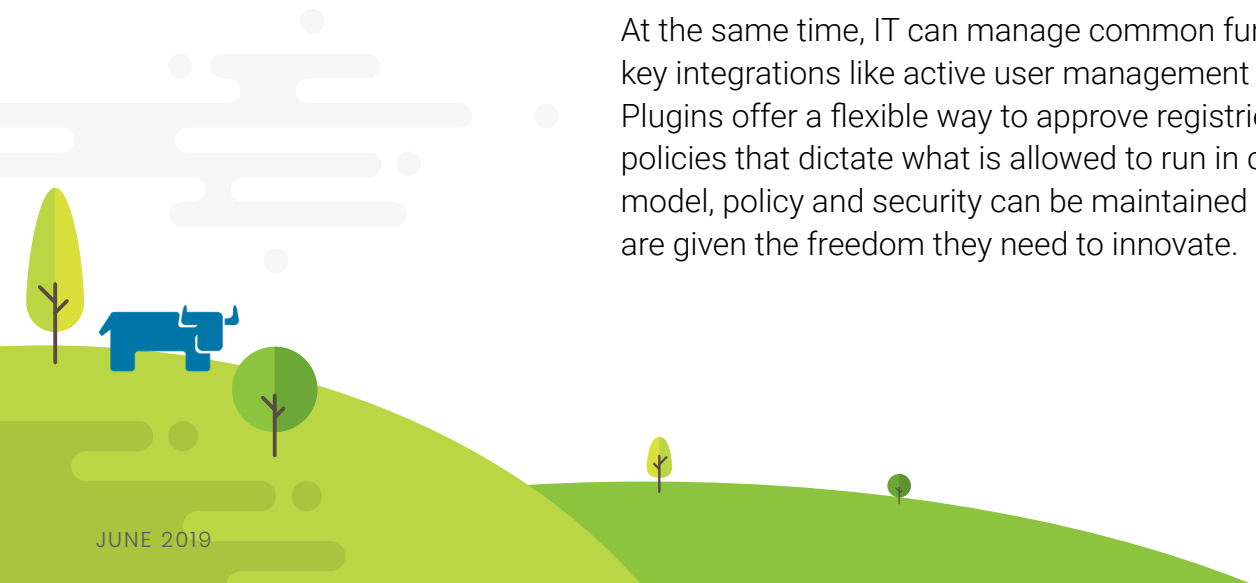
As the case study with Illumina demonstrates, Kubernetes can be deployed in almost any environment and is able to cope with the presence of multiple types of hardware and software in the enterprise, differing network technologies, or even competing desires between DevOps and IT teams. However, it only works at peak efficiency when properly managed through an orchestration platform like Rancher.

As a rising standard for cloud container orchestration, Kubernetes is going to be a part of the enterprise strategies of many organizations. The key to building an efficient enterprise Kubernetes strategy is understanding what a Kubernetes service might look like and then building it in a way to capture the unique benefits of both decentralization and centralization at the same time. Kubernetes and orchestration platforms can help you balance these competing models.

You will want the high degree of autonomy offered by decentralization where teams build what they need, optimized for innovation. On the other hand, you also need the benefits associated with centralization too, such as automating common tasks, integrating systems, easy movement between products, and an ironclad focus on consistent security. Kubernetes-as-a-Service, also known as Container-as-a-Service or Container Management Platforms, will be an integral part of that strategy.

The focus shifts towards managing a framework around Kubernetes. Many organizations are on this journey in different ways, but not everyone means the same thing when they talk about it. Kubernetes-as-a-Service means that the IT and DevOps teams using Kubernetes are on a common platform, which gives them the configuration and operational capabilities required to run exactly the workloads they need.

At the same time, IT can manage common functions and provide key integrations like active user management and plugins. Plugins offer a flexible way to approve registries and define policies that dictate what is allowed to run in clusters. Using this model, policy and security can be maintained even as developers are given the freedom they need to innovate.



Finding the right balance is the primary challenge, and implementing an orchestration platform like Rancher alongside Kubernetes can help get you there. Platforms like Rancher can provide all the benefits of Kubernetes while also reducing complexity. Rancher offers peerless management capabilities regardless of how complex or how unique the platform an enterprise is supporting.

Watch the Kubernetes Master Class Series on Building an Enterprise Kubernetes Strategy to learn more about this topic.

WORLDWIDE LOCATIONS

CALIFORNIA (HQ)
10050 N Wolfe Rd
SW1 STE SW1-272
Cupertino, CA 95014

ARIZONA
1400 E Southern Ave
Ste 1020
Tempe, AZ 85282

NEW YORK
54 W 40th St
5th Floor
New York, NY 10018

THE NETHERLANDS
John M. Keynesplein 12,
1066 EP
Amsterdam

UNITED KINGDOM
Fowler Avenue
The Hub, Farnborough
Business Park
Farnborough GU14 7JF

SHENZHEN, CHINA
1809, Building 2,
Xunmei Technology Plaza,
Kehua Road,
Nanshan District,
Shenzhen, China

www.rancher.com