Industry Whitepaper

January 2022

How To Build An Enterprise Kubernetes Strategy

A Definitive Guide For Government IT Leaders

RGS
Rancher Government Solutions

Contents

Introduction

> 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.

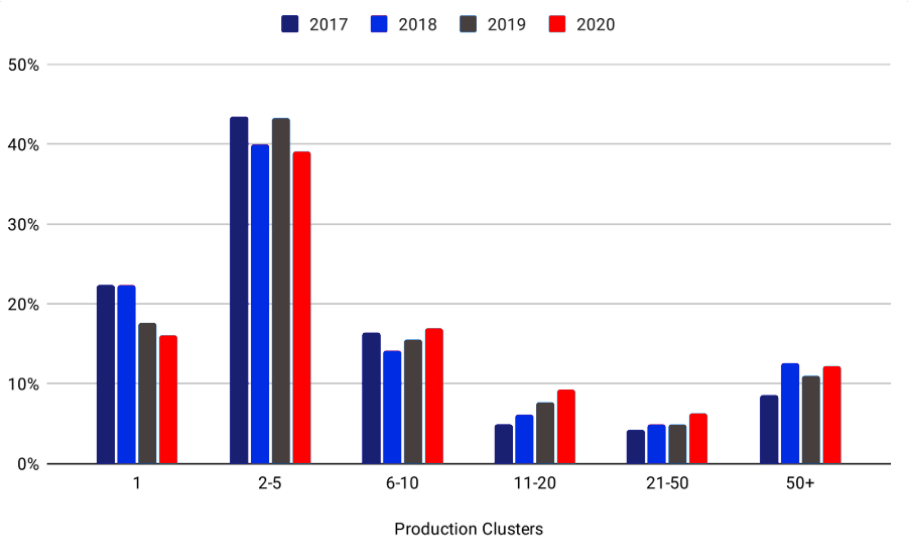
In fact, development teams are using containers to package entire applications and move them to the cloud without the need to make any code changes. Additionally, containers make it easier to build workflows for applications that run between on-premises and cloud environments, enabling the smooth operation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

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

Kubernetes has moved from development and testing to production environments in many enterprises. According to the CNCF Survey 2020, 83 percent of the respondents are running Kubernetes in production, up from 78% in 2019. There has also been a 50% increase in the use of all CNCF projects in just one year.



There is also 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.

Clearly, 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's going to change over the next decade. Over the last several years, accessing Kubernetes has become much easier. Open-source tools make provisioning and upgrading a Kubernetes cluster (a set of nodes that run a containerized application) quick and easy. Cloud providers are now offering Kubernetes as a hosted service. Any team using Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure can provision a Kubernetes cluster in minutes using their popular EKS, GKE, and AKS services respectively.

Self-managed K8S	Hosted K8S
KubeADM	Google GKE
Kops	Amazon EKS
Kubespray	Azure AKS
Rancher RKE	Oracle OKE
K3s	Digital Ocean
Hand-built	Many, many more

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 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 enterprise, be on the lookout for individuals who show existing expertise in containerization. As you progress in building your strategy, developing a team of experts who can administer your Kubernetes clusters and deploy applications to them will be critical to driving adoption.

Where Will You Be Running Kubernetes Over The Next Five Years?

Building an organization-wide Kubernetes strategy means prioritizing your goals for this new technology.

If your team sets out to use Kubernetes 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 your team focuses instead on using Kubernetes to accelerate innovation, 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 organization may be using it in the future.

During the next five years, for example, you may use Kubernetes to do any of the following:

- **Create microservice-centric applications.** Kubernetes is a great way to run modern, microservice-centric applications. It offers a rich set of functionalities that allow teams to determine how different services within modern applications are run, handle unexpected events, connect with each other, and connect with other applications and application programming interfaces (APIs).
- **Rapidly deploy Kubernetes clusters.** 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 that even the most well-organized company is still running dozens of Kubernetes clusters.
- **Move onto the edge.** 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. Running applications this way is referred to as "running on the edge."

- **Develop for single-node devices.** Even single-node devices such as point-of-sale terminals, outdoor advertising, medical devices, 5G-enabled communication equipment, security cameras, or automobiles now benefit from the ability to deploy and run applications easily using microservices. We're witnessing the sprawl of tens of 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's 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 to build your Kubernetes strategy with an expectation that you'll need to be able to easily provision and manage multiple 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 (PaaS) 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 most common that one team takes the lead on delivering on the strategy.

Two teams that often lead the Kubernetes strategy are the shared services team (responsible for supporting developers and DevOps) and the central IT team (responsible for computing platforms).

Putting either team in charge of Kubernetes strategy provides the following benefits:

- **Shared Services:** The shared services team brings key insights on how an organization is modernizing its approach to application development, as well as the requirements teams have identified they need in a Kubernetes platform. They often understand other key systems that have been built for DevOps, such as continuous integration/continuous delivery (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. They should be a critical part of developing your organization's strategy.
- **Central IT:** The central IT team, focused on cloud computing and other computing platforms, is also a logical team to lead a Kubernetes strategy. They have a strong understanding of platform operations, infrastructure, security, multi-tenancy, and existing IT investments, and they usually have significant experience running critical projects. A project led by the IT platforms team will benefit from their understanding of the broad requirements of many different teams across a large, complex organization. Note that 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 owns your strategy, one of the critical questions that will emerge is how much standardization is possible without impacting innovation. Many teams will have experienced projects around OpenStack and PaaS that struggled to get adoption because users weren't 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 hindrance to some teams looking for a platform to just run standard apps.

One of the most exciting developments in the Kubernetes space in the past few years has been the emergence of lightweight projects that run on Kubernetes but provide frameworks that simplify application management. These approaches allow containers to "scale to zero" and provide simple declarative languages to build, connect, scale, and monitor services. They can deliver a powerful experience without requiring a deep understanding of the underlying technology, which can benefit teams using CI/CD and stateless applications. Our Epinio project (https://epinio.io/) is an example of this approach to running containers that simplify some of the complexity of Kubernetes.

Commented [SE2]: Keep? Or just remove the SUSE here?

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 everyday tasks such as Kubernetes cluster lifecycle management, role-based access control (RBAC) policies, infrastructure management, and other day-2 operations. At the same time, give your teams options for where they can get access to Kubernetes clusters and whether they can use a shared cluster or a dedicated cluster. Focus primarily on maintaining visibility into all the provisioned clusters, not necessarily forcing teams to use a set of preapproved clusters in a specified way.

Containerization and Kubernetes Will Disrupt Some of Your Other Plans

If your organization has decided to expand the adoption of containers, Kubernetes will accelerate innovation and become critical to your IT strategy. It's also important to consider how Kubernetes will impact other projects that are already happening. Below, we'll look at a few common projects that Kubernetes might affect. 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 and government agencies focused 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 will 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 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 aims to minimize cloud lock-in and achieve broad application portability. These teams will try not to use 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 use 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 data center 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. Alternatively, 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. If you've chosen this route, a good place to start is our cloud native HCI solution, Harvester. As open-source software, Harvester is free to download and deploy as well as integrating seamlessly with the agency's multi-cluster management platform, Rancher (more on this shortly).

Commented [SE3]: Should this be removed? Is Harvester available for Fed agencies to deploy?

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. An excellent persistent block storage solution to investigate is Longhorn. First developed at Rancher Labs, Longhorn is 100% open-source software whose development is now governed by the CNCF.

We Are Trying To Modernize Our Existing Applications To Improve Security And Stability

A while back, a customer 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 can 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 consistent security, support of the latest operating systems, advanced 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 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 mention earlier, if you find that your enterprise 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 your team's skill level, you'll almost certainly have team members who need to be trained on either using or administering Kubernetes. Luckily, there is no shortage of free Kubernetes training providers and online courses including the Rancher Community Academy.

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 rigorous and help you ensure that you build strong internal knowledge about using containers and Kubernetes.

After 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 bringing on more teams to work with

the specific implementations of Kubernetes your organization is adopting. At this stage, Rancher Government Solutions provides an array of different consulting services which can help you mitigate the risks associated with large, production grade deployments. For more information visit https://rancher.com/government/services

Evaluating Enterprise Container Management Platforms

Up until now, 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. At this point, teams begin to investigate technical options for managing containers across the organization.

Some analyst firms describe this class of software as Enterprise Container Management (ECM) software and is a term used to describe tools like RedHat OpenShift, VMware Tanzu, Google Anthos and Rancher. While we don't want to spend too much time in this document comparing these different offerings, 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 ECM products 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 ECM vendor, it is important that the distribution you use is certified by the

CNCF. This will ensure that it is consistent with upstream Kubernetes and quickly supports 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 an ECM approaches multi-cluster management. Look at how the ECM 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 ECM 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 ECM tools should be oriented toward understanding how you will manage many 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 service 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

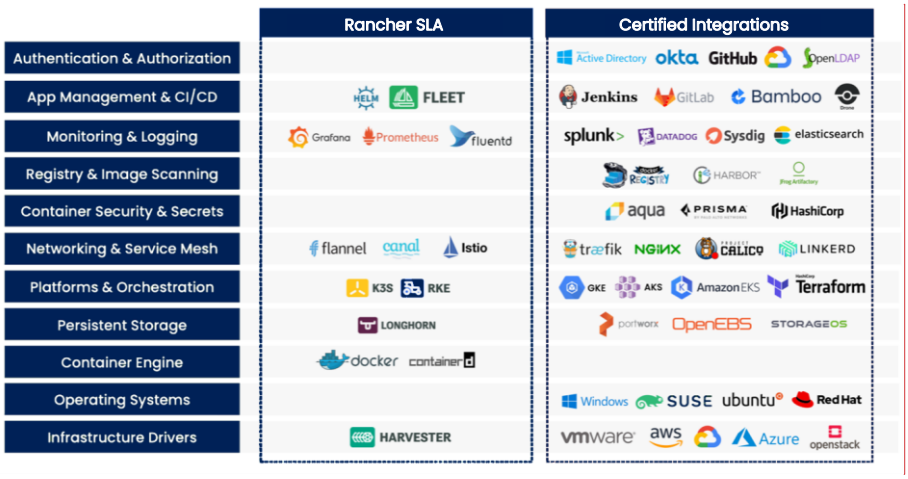
Building a central policy management layer lets you ensure compliance and adequate controls across all your organization's implementations of Kubernetes. Most ECMs will have administrator control planes that allow your teams to define policies and apply them to all the teams using Kubernetes and all 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 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, 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 ECM 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 an ECM is that it puts too much emphasis on integrated solutions and ease of use and ends up limiting flexibility. Kubernetes is well architected for plug-and-play integration with most of its ecosystem, so 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.



Rancher Integrations – SLA Coverage & Certified Integrations

Kubernetes Security and Audit

As you consider ECM tools, it is critical that you collaborate with your security experts and ensure the tools will support your broader security requirements. Most ECM 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 such as Rancher will even assess 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 including NeuVector. Such tools provide different security capabilities on top of the capabilities of Kubernetes or an ECM and are worth evaluating as part of your broader implementation of Kubernetes.

Open Source, SaaS, and Support

If you decide to roll out an ECM, one of the key decisions you'll have to make is how to operate the platform. Most ECM tools today are delivered as software, and many, such as Rancher 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 an option for 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 to move back to the open-source version later if you don't want to pay any more. When you're evaluating the support offered by these ECM 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 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 ECMs. If you decide to move to a hosted ECM, consider how much flexibility you'll have in the future to move off it as your needs change. Is it a shared implementation of the ECM or is it dedicated to your organization? You're going to build lots of policies, templates, best practices, and integrations with an ECM, 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 virtualization 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 agency, 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 that 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.

Appendix – Case Studies

Hypergiant: Putting Managed Compute Power In Space With K3s

Highlights

- 80% reduction in deployment time compared to manual methodology — from weeks to days
- World first: Kubernetes clusters in space for the first time
- Creating a long-term commercial future for space hardware
- K3s clusters created in minutes
- Simple remote update and patching

Products

- Rancher
- K3s Lightweight Kubernetes

What is HyperGiant?

Hypergiant Industries focuses on solving humanity's most challenging problems by delivering best in class artificial intelligence solutions and products. The company creates emerging AI-driven technologies for Fortune 500 and government clients working in a host of sectors, including space science and exploration, satellite communications, aviation, defense, health care, transportation and more.

Previously a 'Cyber Transport Journeyman' in the U.S. Air National Guard, Bren Briggs is used to being deployed to the most hostile environments to design and run command and control communications. Now responsible for DevOps and Cybersecurity at Hypergiant, Briggs draws on his military background, bringing expertise in designing and building repeatable, secure deployments of Kubernetes and other infrastructure to support Hypergiant's AI and machine learning (ML) applications and customers.

Briggs is one member of a wider team of experts working for the U.S. Department of Defense's official DevSecOps enterprise services platform, DoD PlatformONE. Briggs works alongside Rancher Government Solutions' co-founder and Kubernetes Edge Architect Chris Nuber, and Hypergiant's Space Systems Architect Chris Tacke. Together, they are on track to achieve the impossible in the most hostile environment of all — putting K3s clusters to work in orbit on military satellites for the very first time.

The Journey to Kubernetes

Hypergiant and Rancher Government Solutions are working together with DoD PlatformONE to demonstrate the benefits of DevSecOps, Kubernetes and AI/ML apps in remote and often disconnected environments. Working with K3s (recently accepted as a CNCF Sandbox project), the team is developing and integrating their software pipeline with the EdgeONE and SatelliteONE missions, including the launch of Hypergiant's Kubernetes-embedded satellite planned for the first quarter of 2021.

> *"There isn't another version of Kubernetes that can meet our use case. K3s has allowed us to build infrastructure suitable for use in space without the need to create our own edge distribution."*
>
> Bren Briggs, Director of DevOps and Cybersecurity, Hypergiant

Briggs started working with Kubernetes several years ago to solve the problem of managing multiple Docker containers. At that time, containers were built and shipped like virtual machines (VMs) — updated and managed in Puppet. With a growing need for consistency, Kubernetes became an important way to reliably network the estate of containers and manage them centrally.

In 2019, Briggs brought this experience to Hypergiant, where he noticed repetition in existing development methods. He recognized the need for a platform that would allow developers to deploy repeatable patterns with less manual effort. They needed a cloud that could run anywhere, and so Briggs suggested Kubernetes to automate the creation of repeatable workloads. In early 2020, the team started rolling out Kubernetes on several

internal systems. As they ran on AWS, EKS was initially deployed to orchestrate the first cluster, but soon, K3s entered the team's consciousness.

With a mandate to find a way to run Kubernetes in space, Briggs, Tacke and Nuber began investigating. Briggs was impressed. K3s was designed as a lightweight, compact version of Kubernetes that could run in remote, low-power, hostile environments. Available 'off the shelf,' K3s was easy to install and avoided the need for Briggs and the team to 'roll their own' lightweight distribution. Crucially, Briggs estimates deploying a cluster takes just a day using K3s, compared to weeks in a manual methodology. The cluster itself builds in 10 minutes using Ansible plays. Provisioning base services takes another 30 minutes. Production can be ready before launch once the automation has been written.

A World First: Putting Kubernetes Clusters in Orbit

The satellite industry faces several problems. Firstly, space-rated hardware is costly. Software development and delivery processes are slow. And it's cold — really cold. On-orbit satellite software updates are often not possible or incredibly time-consuming and expensive. As a result, AI/ML capabilities are far behind those currently available on earth.

Satellite connectivity and bandwidth are poor, which makes downloading large images and other data difficult. The SatelliteONE mission has been designed to solve this problem. The project will demonstrate DevSecOps in space by leveraging PlatformONE's CI/CD pipeline alongside Kubernetes provisioning and deployment. Importantly, it will evaluate the use of lower-cost hardware on satellite payloads, show how the rapid delivery of software updates in space can be done and demonstrate the use of AI/ML software in orbit.

The U.S. Department of Defense, along with governments worldwide, are looking for ways to build longevity and sustainability into satellite fleets. They want to modernize their entire fleets to be managed, maintained, and, crucially, reconfigured for various use cases for the long term. This is a massive shift, but investment in modernization will create agile technology stacks that can be more easily updated and replaced in the future.

Because K3s is packaged as a single <40MB binary, it reduces the dependencies and steps needed to install, run and auto-update a production Kubernetes cluster. Supporting

both ARM64 and ARMv7, K3s works just as well on a Raspberry Pi as it does with an AWS a1.4xlarge 32GiB server. K3s is enabling Hypergiant to deploy the modern, lightweight software systems that will drive the next evolution and commoditization of the satellite industry.

Deploying K3s to Achieve the Impossible

SatelliteONE's initial mission is an important one—to take a picture of Baby Yoda. While this doesn't sound particularly ground-breaking, this exercise's primary purpose is to demonstrate the ability to capture images and perform AI and ML workloads at the edge.

This is an important milestone given the bandwidth constraints experienced in space. When dealing with a link, measured in kilobits per second, only available for a few minutes at a time during a flyover window, transfer efficiency is the ultimate goal. In this use case, the team wants to transfer the minimum amount of information.

The architecture consists of two Raspberry Pi server nodes, independent of each other, with one being a warm backup. There are two Raspberry Pi 4 worker nodes, each loaded with a Pi camera, and another Raspberry Pi 4 with an accelerometer and light sensor. Finally, there is a Jetson nano module (developed by NVIDIA), a $99 IOT AI processor mounted on the satellite and used to capture and process the image of Baby Yoda.

The two master nodes have an additional ethernet adapter attached to another switch and have connectivity to the Cygnus device and external network, allowing communication with the ground (Cygnus is the vehicle the team is hitching a ride on courtesy of Northrop Grumman). There are two independent K3s server nodes in place. This is critical should one server device fail in orbit — the team can quickly communicate with the server node, automatically failover, or run scripts to rebuild the cluster in a matter of minutes. Since the clusters are logically separated, all the K3s images and the application workloads are stored locally on each node—less overhead and one less point of failure.

Importantly, this is about sending analysis rather than the raw image itself. Ultimately, a human can still make decisions as to which photos to downlink. However, the focus here is to process decisions at the edge, sending analysis and results instead of the raw dataset.

The team is also conducting a temperature experiment to observe how the Jetson behaves in a vacuum under a heavy computational load. Why, though, would the team put a whole satellite in space to take a few pictures of Baby Yoda and conduct temperature measurements?

By proving these basic processes work, the team can highlight the use case for Kubernetes in orbit. Industries of all kinds that rely on satellite communications and imaging want to create modern satellite fleets that can be used for longer and have less environmental impact.

By deploying novel compute technologies, like K3s on orbiting devices, their longevity and ongoing usability are assured. Working with K3s, Hypergiant, Rancher Governments Solutions and DoD PlatformONE are creating a commercial future for data processing in space.

About Rancher Government Solutions (RGS)

Rancher Government Solutions is specifically designed to address the unique security and operational needs of the US Government and military as it relates to application modernization, containers and Kubernetes.

Rancher is a complete open-source software stack for teams adopting containers. It addresses the operational and security challenges of managing multiple Kubernetes clusters at scale, while providing DevOps teams with integrated tools for running containerized workloads.

RGS supports all Rancher products with US based American citizens with the highest security clearances who are currently supporting programs across the Department of Defense, Intelligence Community and civilian agencies.

RGS is committed to helping the US Government run Kubernetes securely everywhere.

Rancher Government Solutions
1900 Reston Metro Plaza, Suite 600
Reston, VA 20190
800-796-3700
RancherGovernment.com
Cage Code: 8GLZ3 DUNS: 11738783