spectro cloud

CDW PEOPLE WHO GET IT

BUILDING THE MULTICLOUD GITOPS FOUNDATION WITH KUBERNETES

GitOps, meet Kubernetes

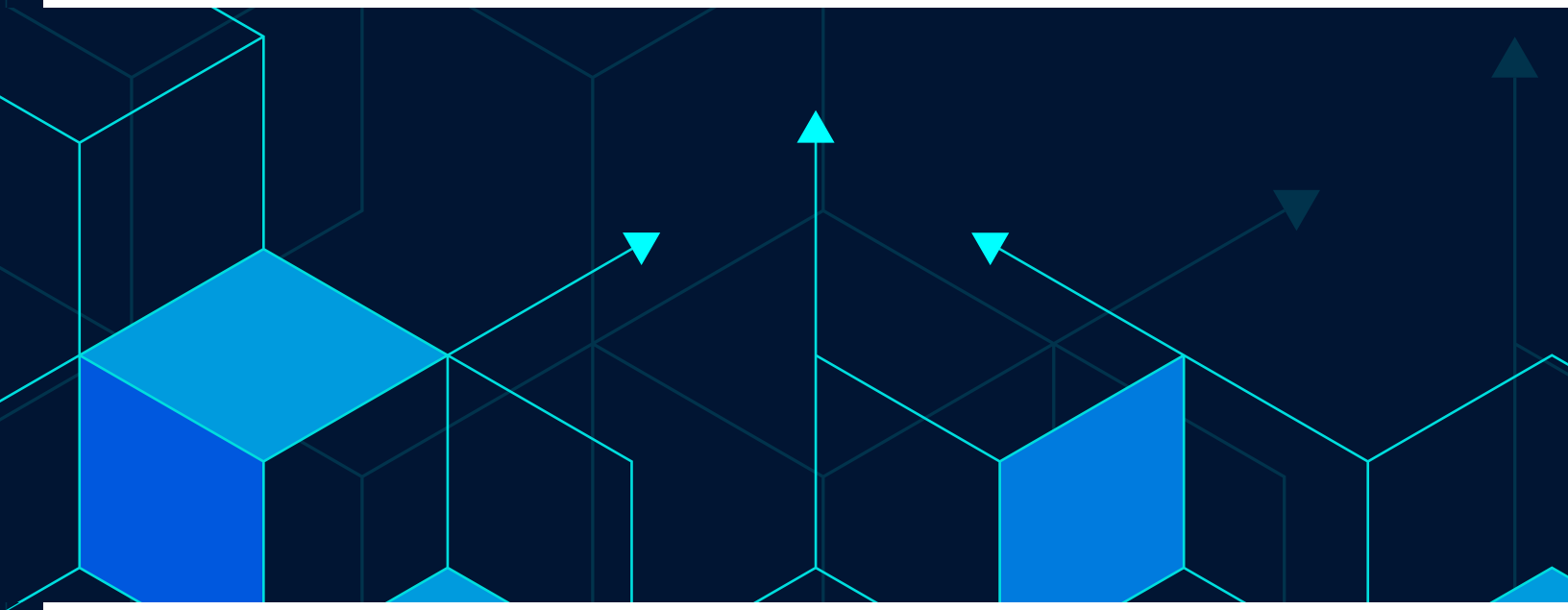
If you're adopting GitOps patterns to accelerate software delivery, you're probably also choosing Kubernetes (K8s), lured by its flexibility and power. However, if your organization is new to the game, you'll quickly discover that standing up K8s clusters across multiple clouds can be extremely complex and challenging to implement and maintain.

Application developers find they need to be Kubernetes experts to test and deploy their code. Platform teams are soon burning hours on provisioning and managing clusters and on the whole stack of observability, security, and other supporting components.

At CDW, our Digital Velocity team has partnered with Spectro Cloud to solve for some of these challenges.

GitLab, Terraform Cloud, and Spectro Cloud Palette are the building blocks of a prescriptive services engagement we call the Multicloud GitOps Foundation (MGF). Think of it as a fast way to establish a "Landing Zone" for Cloud Native applications.

In this article, we will explore the ways Kubernetes has shaped how organizations build modern applications, and how — together with the right components — K8s can be an order of magnitude less complex to manage at scale than existing software deployments. We'll touch on what we're seeing in the Kubernetes landscape, along with trends in multi-cloud that influenced our design of the MGF engagement.



Kubernetes is the cloud operating system

Kubernetes adoption has exploded. In the 2022 Stack Overflow Developer Survey, nearly 60,000 developers identified Docker and Kubernetes as the number 1 and number 2 most loved tool and the ones most targeted to adopt in the future[1].

It's easy to see why. Docker and Kubernetes promise a modular and portable framework to build and run applications.

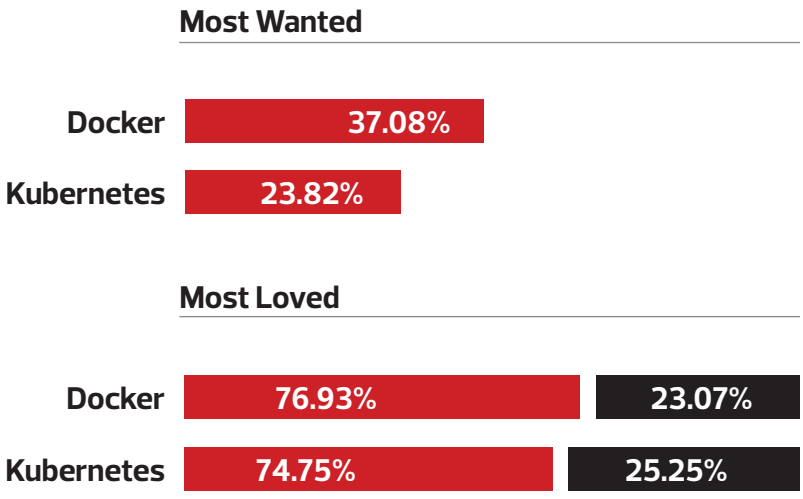
Additionally, because Kubernetes offers the flexibility to uniformly run workloads as VMs where necessary, containers for longer running workloads, and native for serverless, it is positioned as the new cloud operating system that teams are running toward.

Complexity: the dark side of Kubernetes

However, we also recognize the challenges with Kubernetes. Various flavors of managed cloud services have emerged, along with dozens of Kubernetes distributions.

And that's just the start. Kubernetes clusters are never complete with just Kubernetes itself. There are adjacent technologies and supporting stack resources that teams need to deploy in clusters to deliver core functionality like networking, observability, storage, and security.

As of this writing, there are over 1,100 projects tracked by the Cloud Native Computing Foundation (CNCF). These projects can be overwhelming to navigate as teams look to adopt an approach to Kubernetes that works for them.



[1]Stack overflow 2022 survey https://survey.stackoverflow.co/2022/#most-loved-dreaded-and-wanted-tools-tech-want

The Kubernetes distribution providers also have their sets of opinions you can consume, but this can feel more like train tracks than a paved road, with no ability to deviate from those opinions. It's the Henry Ford model for Kubernetes with these opinionated stacks, "*Any customer can have a car painted any color that he wants, so long as it is black.*" – Henry Ford.

With opinionated K8s deployments, you must rely on the ecosystem of that provider. However, with the MGF tools, you can be as opinionated or un-opinonated as wish; you don't just have to buy what that provider wants you to buy.

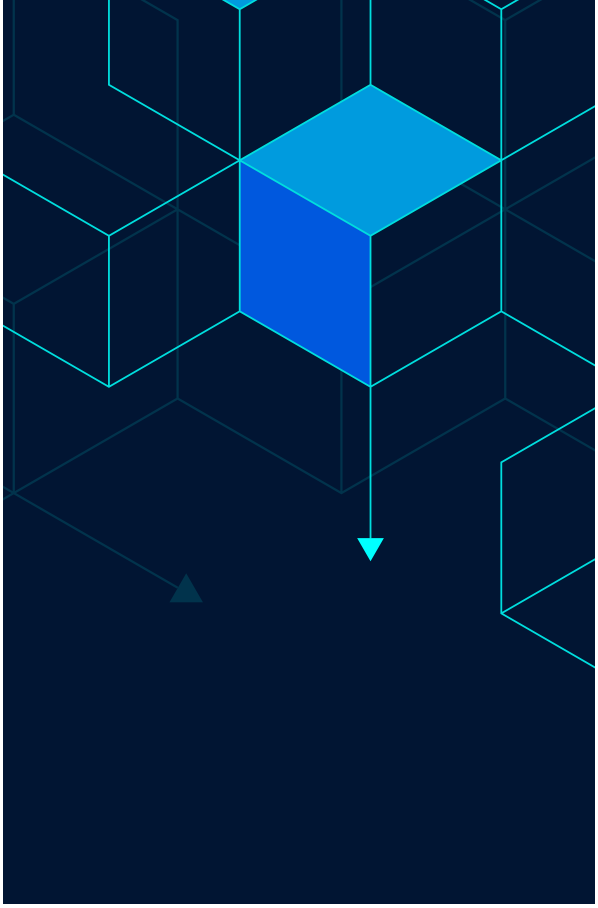
Teams are often looking for an easy way to lasso the power of Kubernetes while maintaining control of their platform. We feel the tools and services described in this blog answer those needs.

Multicloud adoption – patterns not a place; application anywhere

There is an industry trend with organizations choosing to be in multiple public or private clouds.

In a June 2022 Forrester study, 47% of organizations said they would expand their multi-cloud approach. 90% said that using multiple clouds has helped advance their organization's business goals.

Why is this? A multi-cloud approach allows organizations maximum flexibility and allows cloud and platform teams with the



ability to say "yes" to developers who need to quickly use a specific cloud service to access a new feature or capability.

But there are many challenges in how organizations take advantage of mult-icloud targets for the deployment of applications and services. The top cited barriers are centered around skills, costs, security, and complexity.

However, platform engineering concepts have answered some of the questions around tackling these challenges. Platform engineering teams are built to reduce the cognitive load of developers and the value stream-aligned teams, while working to stay productive as they move the business forward. Development teams no longer have to spend two months or two years learning Kubernetes to test and deploy their application quickly.

What works?

Platform engineering practices can focus on:

- Defining and measuring reliability
- Continuously reducing toil and increasing automation
- Enabling education and encouraging platform adoption
- Building in security and compliance
- Building internal communities through advocacy
- Providing a delightful developer experience ; allowing developers to quickly test and deploy applications
- Employing pragmatic standardization
- Optimizing costs with "chargebacks" and "showbacks"
- Aligning goals to business outcomes

With these in mind, organizations adopting multi-cloud are largely turning to standardized tools that allow uniform approaches and workflows across clouds and environments. They are shifting to a platform mindset to focus on iteration and ease of use, leveraging a "platform-as-a-product" mentality.

Bringing GitOps into the mix

The GitOps Working Group (WG) is under the CNCF App Delivery SIG and has largely led the effort to codify GitOps and create a "vendor-neutral, principle-led meaning".

Its definition of GitOps can be encapsulated in four foundational concepts:

1. Declaratively expressing the desired state
2. Storing that desired state in a way that enforces immutability, versioning, and retention of a complete version history
3. Having software agents automatically pull the desired state declarations from the source
4. Having software agents continuously observe the actual system state and attempt to apply the desired state

The WG states that "individuals, teams, and organizations who implement GitOps experience many benefits, including:

- Increased developer and operational productivity
- Enhanced developer experience
- Improved stability
- Higher reliability
- Consistency and standardization
- Stronger security guarantees"

Together, GitOps concepts and Palette provide a basis for simplifying K8s platform adoption across any cloud, delivering multi-cloud operating capabilities.

Why does this matter? Users can now start to place a cloud anywhere and use any version of resources together. They can also now add something like Crossplane that gives a consistent way of automating cloud resources.

Lets review Palette features:

Cluster profiles

- Version cluster resources and supporting software
- Flexibly select resources not tied to opinions
- Simplify project deployments like service mesh and observability into clusters at the time it's provisioned
- Blueprints for successful deployments

Cluster deployment

- Use cloud accounts to reduce credential sprawl
- Deploy to any cloud
- Create fault-tolerant clusters in multiple availability zones

Virtual clusters

- Quickly create clusters inside namespaces
- Quickly provide users with entire clusters for their use
- Use Virtual Clusters to fight Kubernetes sprawl and optimize workloads on existing deployments

Security

- Integrate with your Identity Provider (IdP), RBAC, and SSO
- Use RBAC for Palette and cluster access
- Out-of-the-box configuration, performance, and penetration testing
- Scanning deployed software; Palette includes a Software Bill of Materials (SBOM) that tracks software development metadata such as version, origin, and license. This enables organizations to track vulnerabilities, perform regular software maintenance, and ensure compliance with regulatory requirements

Organizing the Multicloud GitOps Foundation

Sometimes, organizations can spend months or years ***planning*** an IT infrastructure project, when they could be ***doing it***. Spectro Cloud Palette, helps teams get started sooner and brings enterprise-ready Kubernetes orchestration and management for organizations with existing K8s workloads.

As GitOps builds on ideas established in DevOps and Infrastructure as Code (IaC), we felt the growing need for ease of provisioning infrastructure must rely on these principles. So when we organized the MGF, we established the following criteria:

- **Portability**
  Teams must be able to deploy clusters and workloads on any of the big three cloud providers, to bare metal, to VMware, and to the edge
- **Code-first approach**
  We apply the benefits of IaC (repeatability and consistency) across clouds
- **Short-lived resources**
  We recreate clusters on-demand and spin up new resources quickly
- **Simple**
  Reduce complexity with easy to deploy supporting software
- **Customizable**
  Customize the modularized platform
- **Scaleable**
  Deploy and reduce the administrative burden of managing Kubernetes
- **Reduces toil**
  Reduce toil by using end-to-end automation
- **Consistency**
  Standardized and codified higher and lower environments

We needed to check the box on all these criteria, and we quickly realized that only Spectro Cloud provides the basis of orchestration.

> **To learn more about Spectro Cloud's Palette management platform, click here.**