गोपाल [1]: Open Source Kubernetes cluster management framework

Deepak Sharma
Founder
गोपाल *[1]*
Madhya Pradesh, India
sadysnaat@gmail.com

***Abstract*—Since its inception on June 6th, 2014, Kubernetes has come a very long way. It has been adopted by 8,000 companies [2] and boasts a flourishing ecosystem with open-source tools like etcdadm, kubeadm, and Cluster API to provision and manage clusters. However, managing and operating thousands of clusters, along with managing and upgrading all cluster versions across an organization, can be a daunting task. To address this challenge, meet गोपाल (pronounced go paal) [1], a GitOps [3] first framework.**

***Index Terms*—kubernetes, cluster, management, cluster-api, ui, ux, git, git ops**

I. Introduction

Kubernetes provisioning has significantly evolved, from "Kubernetes the Hard Way" by Kelsey Hightower [4] to the fully declarative solution of Cluster API. Cluster API is gaining popularity and adoption [5] across many companies. Creating a cluster (Day 1) and managing subsequent operations (Day 2) on a Kubernetes cluster remains a routine task for many operations teams, regardless of their size or the number of clusters they manage for their organization. In the following sections, I will outline the challenges faced by operations teams and discuss possible solutions, paving the way for a holistic approach.

II. Challenges

With the introduction of Kubernetes cluster provisioning tools like kubeadm, etcdadm, and Cluster API, we have achieved the ability to launch and manage clusters declaratively. Cluster API has proven to be an invaluable tool.

For operations teams, creating and updating clusters is not the only goal. They must repeatedly create similar clusters and sometimes launch such clusters across different clouds. Organizations need to uniformly apply security best practices across all clusters and clouds.

Teams have to manage fleets of clusters and face several challenges, some of which are highlighted below:

1) Challenge 1: Users have to manage several cluster operations impacting hundreds of clusters. All of these operations are done via kubectl, presenting risks of mistyping [6], using incorrect cluster kubeconfig, and allowing actions to be performed by a single individual, thus missing a maker-checker workflow to approve and review such changes.
2) Challenge 2: When users modify cluster properties, there is a large impact radius. Changes done to a cluster can lead to misconfigurations and, in some cases, can cause clusters to be completely annihilated [7].
3) Challenge 3: Users lack an open-source tool/framework that links Cluster API capabilities with GitOps principles to provide end-to-end deployment capabilities.

III. Current Solutions

As the fleets of clusters grew, organizations faced numerous challenges and responded by creating ad-hoc or dedicated solutions, or by relying on external tools. Let's review a few of these solutions:

1) Build a Solution from Scratch:

   Inspired by organizational constraints or due to the "Not Invented Here" (NIH) syndrome, some companies have created CI/CD pipelines.

   This solution addresses Challenge 1 very well. Instead of manually changing the resources, operators are required to go through CI/CD pipelines, reducing the chance of errors compared to manually editing files. However, depending on how rollouts for different clusters are implemented, Challenge 2 might still remain unresolved. Challenge 3 cannot be addressed by this solution.
2) Use vendor based solutions:

   Vendor managing clusters for you. This allows companies to choose from a list of vendors [8]. Some are Open Source, some are closed sources. Though this approach is really good from Open Market perspective, but takes away lot of cluster management control away from users.

   This solution addresses challenge 1 depending on how good the solution is you will have varying functionalities. Challenge 2 is hard to predict, but it is highly likely that vendor bugs have taken down your clusters, caused you downtimes. Challenge 3 depends on vendor organization policy for code sharing.

IV. गोपाल [1]

गोपाल presents a solution to these problems: गोपाल. This GitOps-enabled framework, combined with Cluster API, simplifies the creation and management of clusters. Imagine not having to install any agents in the cluster, subscribe to or buy third-party tools, or build your own tooling from scratch. This is the vision I have for this framework. Since the inception of GitOps [9], it has adhered to four primary governing principles. Let's review how our framework can assist Kubernetes clusters in this area.

1) Declarative:

   A system managed by GitOps must have its desired state expressed declaratively.

   Leveraging Cluster API allows us to be declarative in state management and cluster configuration.

2) Versioned and Immutable:

   The desired state is stored in a way that enforces immutability, versioning, and retains a complete version history.

   Git and Cluster API declarative templates together provide versioning and immutability properties.

3) Pulled Automatically:

   Software agents automatically pull the desired state declarations from the source.

   गोपाल provides a framework to pull changes from hosted environments, making it easy for developers and operations pipeline building teams to get the latest changes.

4) Continuously Reconciled:

   Software agents continuously observe the actual system state and attempt to apply the desired state.

   गोपाल offers out-of-the-box framework functions that can help create Internal Developer Platforms to run and manage Kuberntes Clusters. The choice of how to leverage these features belongs to the user.

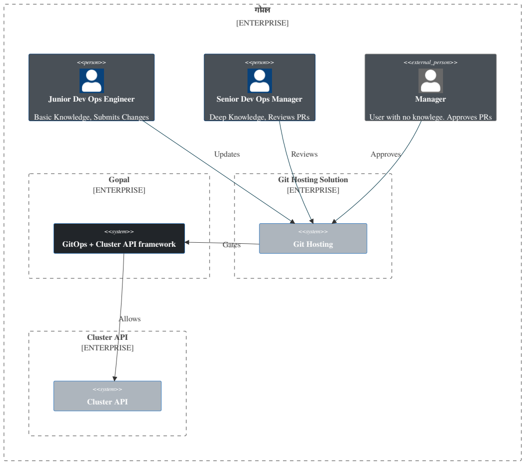

Fig. 1: गोपाल C4 Context

V. Challenges addressed.

- Challenge 1: This is fully addressed by using GitOps. Users no longer have to manually perform any operations, eliminating the margin for human error in the cluster management process. Only reviewed pull requests (PRs) are allowed to merge into the master branch, reducing the risk of unreviewed changes by individuals.
- Challenge 2: This challenge is partially covered with GitOps. गोपाल provides additional capabilities to control upgrades or changes to clusters in a controlled manner (Deployment Strategies Rollout, A/B Testing, and Canary Deployments). Even after reviewed PR changes are merged, users can still manage the blast radius and control the impact.
- Challenge 3: गोपाल by providing a community framework built solely on Open Source projects, no additional tooling is required and can manage end to end cluster life cycle.

VI. Comparison with Current Solution.

Compared to homegrown/vendor solutions, गोपाल is a community-driven, aims to be standard solution that is not overly tied to a particular tool set or deployment style. It will help drive a standard template to specify clusters and updates.

VII. Conclusion

As we are seeing industry wide adoption of Kubernetes, and a few departures [10] as well. One thing for sure is Kubernetes is here to stay for long term. 10 years down the line and companies now widely adopting Kubernetes, presenting a case for Kubernetes/Cluster API native GitOps solution which Organizations can adopt and use with confidence to manage and operate their clusters. गोपाल is one such attempt in this direction. Foundations of this framework cannot be started without thanking Kubernetes [11], Cluster API [12] and Open GitOps [3].

References

[1] "Gopala Krishna." Accessed: Oct. 06, 2024. [Online]. Available: https://web.archive.org/web/20241006062416/https://en.wikipedia.org/wiki/Gopala-Krishna

[2] F. M. K. F. T. B. Bob Killen Chris Short, "10 Years of Kubernetes." Accessed: Nov. 12, 2024. [Online]. Available: https://web.archive.org/web/20241112010816/https://kubernetes.io/blog/2024/06/06/10-years-of-kubernetes/

[3] "Home — opengitops.dev." Accessed: Oct. 05, 2024. [Online]. Available: https://web.archive.org/web/20241005175354/https://opengitops.dev/

[4] K. Hightower, "GitHub - kelseyhightower/kubernetes-the-hard-way: Bootstrap Kubernetes the hard way. No scripts. — github.com." Accessed: Nov. 17, 2024. [Online]. Available: https://web.archive.org/web/20241117184151/https://github.com/kelseyhightower/kubernetes-the-hard-way

[5] CNCF, "Kubernetes Cluster API reaches production readiness with version 1.0 — cncf.io." Accessed: Oct. 05, 2024. [Online]. Available: https://web.archive.org/web/20241005104802/https://www.cncf.io/blog/2021/10/06/kubernetes-cluster-api-reaches-production-readiness-with-version-1-0/

[6] S. E. Stuart Davidson, "How a couple of characters brought down our site — SkyscannerEng." Accessed: Mar. 27, 2023. [Online]. Available: https://medium.com/@SkyscannerEng/how-a-couple-of-characters-brought-down-our-site-356ccaf1fbc3

[7] grumpimusprime, "You Broke Reddit: The Pi-Day Outage : r/RedditEng." Accessed: Mar. 27, 2023. [Online]. Available: https://web.archive.org/web/20230327043551/https://www.reddit.com/r/RedditEng/comments/11xx5o0/you_broke_reddit_the_piday_outage/

[8] “CNCF Certified Parteners & Providers.” Accessed: Dec. 05, 2024. [Online]. Available: https://landscape.cncf.io/?group=certified-partners-and-providers

[9] “GitOps what you need to know — web.archive.org.” Accessed: Dec. 13, 2023. [Online]. Available: https://web.archive.org/web/20231213011557/https://www.weave.works/technologies/gitops/

[10] A. d. B. F. Christian Weichel, “We’re leaving Kubernetes - Blog — gitpod.io.” Accessed: Nov. 17, 2024. [Online]. Available: https://web.archive.org/web/20241117143717/https://www.gitpod.io/blog/we-are-leaving-kubernetes

[11] “Production-Grade Container Orchestration — kubernetes.io.” Accessed: Nov. 17, 2024. [Online]. Available: https://web.archive.org/web/20241121115214/https://kubernetes.io/

[12] “Cluster API.” Accessed: Oct. 05, 2024. [Online]. Available: https://web.archive.org/web/20241005171549/https://github.com/kubernetes-sigs/cluster-api