weaveworks

CASE STUDY

THE CUSTOMER

With more than 8.3 trillion dollars in assets under its administration and 3.3 trillion in total discretionary assets, Fidelity Investments is one of the world's largest asset managers. The firm offers a wide range of investment products and investment services that span retail, workplace, institutional and intermediary markets. Their long-term investment in the latest technology has the goal of simplifying and digitizing support for more than 500 applications that support their customers.

Kubernetes was chosen by Fidelity for cloud application delivery. They teamed up with AWS and Weaveworks to implement secure and compliant Kubernetes platforms.

> *"The challenge is not in the technology itself, or the tools... The challenge is mainly building the structure inside the teams. We're building many centers of excellences across all of our business units and all of our teams. To build a structure across 10,000 developers plus is a major challenge."*
> **- Amr Abdelhalem, Fidelity Investments**

CHALLENGES

Fidelity wanted to take advantage of the main benefits of Kubernetes, and increase velocity, reliability and scalability by moving their applications across multiple clouds. But they needed to make the move securely and keep within the company's extensive regulatory guidelines. In addition to this, platforms needed to accommodate several different development teams, some with specific business requirements like machine learning.

Safeguarding Kubernetes in a highly regulated environment

Like all financial services organizations, every application Fidelity creates must meet a unique mix of regulatory, security and governance requirements. The team at Fidelity wanted the scalability and reliability that comes with adopting Kubernetes, and also sought to leverage the ecosystem of available cloud native open source projects in order to remain innovative and improve their time to market. However they needed to implement cloud native technologies within their highly regulated environment using their existing control and audit guidelines.

Industry: Financial Services
Location: United States of America

HIGHLIGHTS

- Reproducible clusters across environments and multiple public clouds
- Met governance and compliance regulations
- Reliable cluster lifecycle management through pull requests

KEY BENEFITS

- Self-service Kubernetes platform management for application engineers
- Mean time to recovery from hours to minutes
- Secure and reliable platform configuration across multiple backends

CONTACT US

www.weave.works

sales@weave.works

Managing unique business units

Booting a Kubernetes cluster is dead simple these days, but managing secure and reliable complete cluster platforms including all of the required add-ons across environments and on multiple clouds can be a challenge. In addition Fidelity needed to implement specialized machine learning stacks and other specific cluster stacks to meet their innovative business requirements.

Reliable and reproducible cluster configuration

Configuring and replicating cluster platforms across environments can be time consuming and error prone. Some teams may also need specific types of platforms that require a different toolchain and that also need to be reproduced by different teams across multiple backends. If these platforms are manually configured each time, it can slow down the team. In the worst case scenario, you can end up with a set of snowflake clusters that are impossible to update, secure and otherwise maintain.

SOLUTION

Fidelity built a platform with an abstraction layer on top of Kubernetes that allowed them to maintain regulatory compliance, yet didn't impact the speed or innovation across their teams. The FIDEKS platform they built operates and manages applications on premise as well as AWS managed Elastic Kubernetes Service (EKS) and on Microsoft Azure using Weaveworks' solutions and GitOps as the underlying architecture.

Platform configurations representing specific toolchains were bundled together as configuration models and kept in Git. Configuration models allow applications engineers to use GitOps to spin up and manage complete platforms without the assistance of the platform team. It also freed up the engineers from having to worry about operational tasks like configuring tools to work together for a complete Kubernetes platform.

Automating platform configuration and managing it with GitOps provides a way for the SREs to easily perform upgrades on these configurations and to introduce controls through pull requests. The result is a standard cluster development platform that spans environments with security guardrails and a common set of cloud native patterns, like secrets management, in place that any development team can deploy to and from.

Self-service Kubernetes platform managed with GitOps

The FIDEKS platform adds several core capabilities on top of Kubernetes and implements GitOps best practices to deliver and manage them.

Fidelity maintains a centralized Git repo where all of the various base platform versions are kept as well as specialized add-ons included as configuration models. This approach allows for business units to include their customizations and variations on top using GitOps.

Building a specialized platform is a two step process:

1. The cluster and its basic add-ons are configured and then checked into Git as a base cluster.
2. Business units who require additional tools can check out a specific configuration model from Git and then apply it to the base cluster.

If additional tooling is required that is not already in Git, teams may define a specific configuration model with the tools they need and then check that into the repository for future use. With configuration models kept in Git, they are trackable, and auditable and can be used by anyone on the team for consistent and reproducible cluster platforms.

> *"...when I talk about platforms, what we're really talking about is the infrastructure component of things. Obviously both EKS and Kubernetes play a big role. We also have fifteen or sixteen different components like the ELB ingress controller, external DNS and other components that are open source, plus we provide the autoscalers to developers as well."*
>
> **-- Niraj Amin, Director, Cloud Platform Architecture**

Team workspaces and tenancy

In addition to cluster add-ons and specific tool configuration management, teams were able to self-manage cluster tenancy across multiple backends using the same developer experience that GitOps provides. Git-based cluster tenancy management through workspaces enabled multiple development teams to securely make use of single clusters for different environments like development, and QA and therefore saving on costs and time.

Guardrails, security and control

A dedicated platform team manages Fidelity's Kubernetes implementation and serves the needs of developers. Their job is to get out of the way of the developer, so they can focus on innovation and feature development.

Implementing GitOps meant the team could easily meet internal regulations through pull requests. Compliance is further supported through Git's built-in auditory trail and security that guarantees author provenance.

By implementing operations by pull request, such as managing tenancy, Fidelity was able to maintain internal security standards. They could also manage secrets and other core capabilities that the platform controls with GitOps workflows.

> *Rolling out an update to an existing process when you have a complex workflow and many security gates is very difficult, so packaging an application's context as part of an automated platform is of immense value to us."* -- **Rajarajan Pudupatti, Cloud Platform Architecture Director**

Full cluster lifecycle management

Rollouts for all cluster updates are also managed with GitOps. With all base cluster configuration as well as the configuration models representing add-ons and toolchains kept in Git, it's also simpler and more secure to use GitOps best practices for cluster updates and rollouts.

Because GitOps allows for all platform configurations to be bundled as models and versioned in Git, developers never have to worry about operational tasks. Security patches and tool upgrades are fully automated through pull requests. The result is a standard platform with the guardrails and security in place that any development team can deploy to and from.

Results

Fidelity was able to migrate their portfolio of over 500 applications onto Kubernetes across multiple clouds such as EKS and Azure. One of the goals of the team was to create a platform that gave the team the agility as well as the autonomy between teams while at the same time maintaining regulations and other compliance within their organization.

Increased deployment frequency

Providing teams with the autonomy to spin up platforms when they need them and to meet specialized requirements, like machine learning, significantly increased the deployment output of the engineering teams at Fidelity.

Decreased Lead time for changes

Giving development teams the tools and processes that can be managed through Git with GitOps best practices dramatically decreased the time it took for Fidelity to release new applications to production.

Declarative definitions kept in Git also allowed developers to implement familiar workflows in order to manage infrastructure on their own without the platform team's help. This significantly reduced the time it takes to spin up platforms for any environment on multiple backends and across clouds.

Mean time to recovery from hours to minutes

The time it takes to recover from a cluster disaster is also decreased with GitOps. Since your entire system is described in Git, you have a single source of truth from which to recover after a cluster failure, reducing your meantime to recovery (MTTR) from hours to minutes.

Note: This case study was compiled from the 2019 KubeCon joint talk delivered by Alexis Richardson, CEO Weaveworks and Rajarajan Pudupatti, Cloud Platform Architect at Fidelity Investments: "Fidelity's Move to "Finance Grade" Kubernetes with GitOps"