

Simplify application management in Kubernetes environments

Streamline life-cycle management tasks with Helm and Operators

See what's inside

Page 1

Automation simplifies cloud-native applications, software, and environments

Page 2

Automate common processes within your Kubernetes environment

Page 6

Choose the right tool for your objectives

Page 7

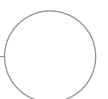
Gain more choice with Red Hat OpenShift

Page 8

Realize business value through automation

Page 10

Ready to start automating?



Automation simplifies cloud-native applications, software, and environments

Across industries, organizations use cloud-native applications to deliver innovative services and products, analyze business metrics, and support e-commerce. Many of these organizations establish container- and Kubernetes-based environments to develop and deploy their cloud-native applications. However, managing applications and the related infrastructure software across these environments can be challenging. Common management processes – like deployment, configuration, and updates – often involve many repetitions of smaller tasks, complicating operations and increasing the risk of errors.

Automation can help you simplify these processes to deploy and maintain applications and software more easily. By automating routine tasks, you can increase speed, efficiency, and scale to deliver more value. Automation also reduces the risk of errors due to manual intervention, increasing deployment accuracy and consistency. Developers can create a better experience for application users and users can run applications more effectively to support business outcomes.

This e-book discusses two automation technologies for managing applications and software within Kubernetes environments and provides guidance on when to use each.

Key concepts

This e-book uses several key terms related to cloud-native application and infrastructure software management.

- ▶ A **cloud-native application** is any application that relies upon cloud infrastructure or services to run in a private, public, or hybrid cloud environment.
- ▶ **Infrastructure software** provides infrastructure services – including storage, networking, load balancing, and security – to applications.
- ▶ **Day 1 operations** are tasks associated with installing and setting up applications and resources. Examples include deployment and initial configuration.
- ▶ **Day 2 operations** are tasks associated with application and resource maintenance. Examples include updates, back ups, recovery, monitoring, and scaling.

Reasons to automate in a cloud-native environment



Increase agility.

Accelerate application and software life-cycle operations like deployment, updates, and backups.



Boost security.

Ensure applications and software are up to date and running according to security rules and policies.



Simplify operations.

Codify expert knowledge to help others deploy and run applications and software more easily.



Scale effectively.

Give your staff the tools they need to share applications, be productive, and achieve more, faster.



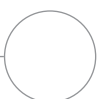
Improve reliability.

Reduce the risk of manual errors and ensure tasks are performed consistently each time.



Deliver a great experience.

Offer users a simpler experience with fast, straightforward access to applications and infrastructure.

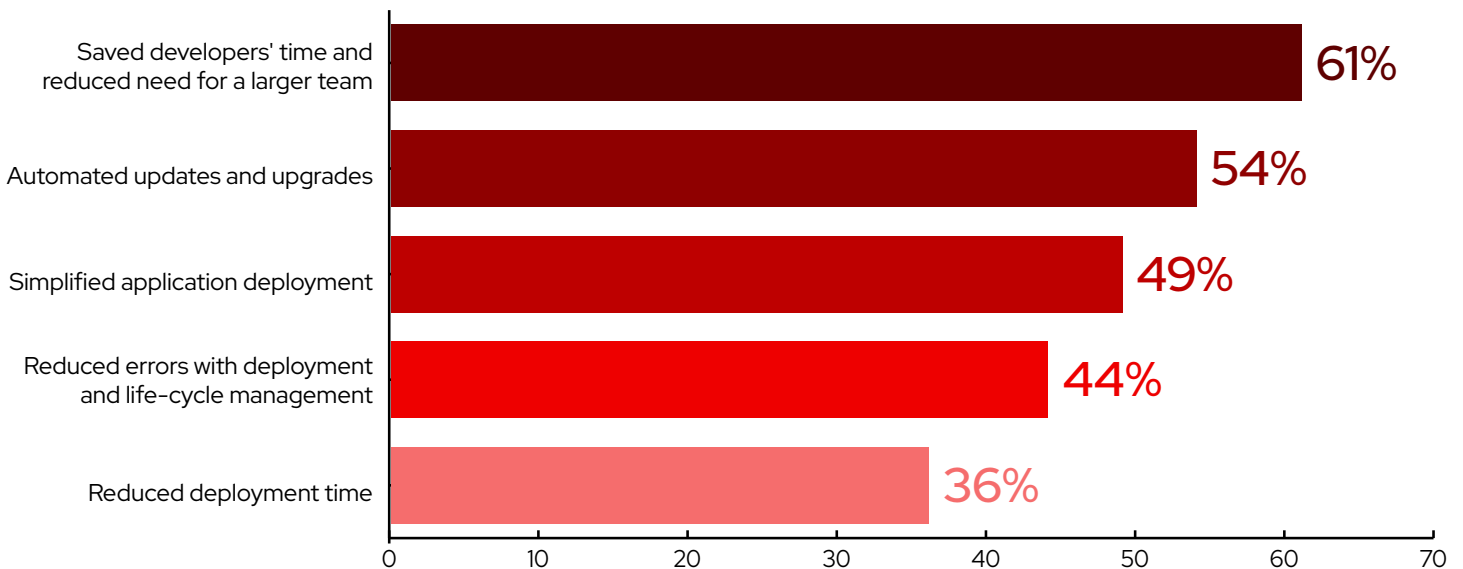


Automate common processes within your Kubernetes environment

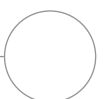
Helm and Kubernetes Operators are two popular choices for automating the management of application and infrastructure software within your Kubernetes environment. These tools help to improve developer productivity, simplify application deployment, and streamline updates and upgrades.

While both Helm and Kubernetes Operators deliver many benefits, each technology supports different capabilities and automation scenarios. The following pages give a summary of each technology, its capabilities, and common use cases.

Top benefits of using Helm and Kubernetes Operators¹



¹ Pulse, sponsored by Red Hat. "State of workloads adoption on containers and Kubernetes," June 2021.



Helm

Helm is a popular package manager for Kubernetes-based application environments. It provides packaged, templated versions of applications and infrastructure software with their dependencies.

Main objective

Helm automates day 1 and some day 2 operations, helping users to quickly deploy and update applications and software with basic customizations. Using Helm, developers can package and share their applications more easily and define how their applications are deployed.

How does Helm work?

Helm organizes the required Kubernetes resources and configuration details into charts and value files. Helm charts contain a list of everything your application or software needs to run. Value files define the variables needed to configure the required resources. When you invoke a Helm chart, the Helm client combines these files into a release of your application or software and deploys the appropriate Kubernetes resources.

Helm relies upon built-in Kubernetes capabilities and can only automate native Kubernetes tasks like deployment, configuration, updates, and retirement. Helm runs only when invoked and cannot monitor application or software status.

Where does Helm run?

The Helm client invokes Kubernetes application programming interfaces (APIs) from outside your cluster. Helm cannot respond to cluster events or integrate with external services.

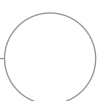
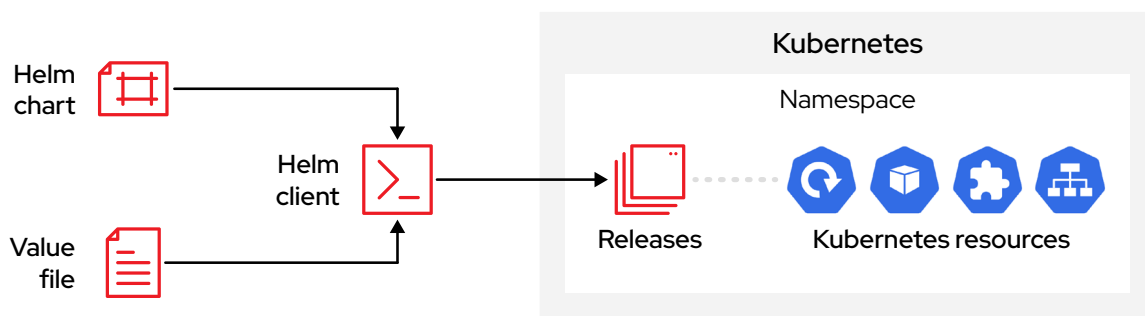
How are Helm charts managed?

Helm charts are managed manually through a catalog. When an update to a Helm chart is available, users must upload the new chart to the catalog and then run the chart to redeploy the application or software with changes implemented.

What can you accomplish with Helm?

Helm helps you streamline day 1 operations:

- ▶ Simplify and speed application deployment processes.
- ▶ Reduce deployment errors due to manual intervention.
- ▶ Enhance continuous integration/continuous deployment (CI/CD) pipelines.
- ▶ Package and share applications and software more easily.



Kubernetes Operators

Kubernetes Operators are a method of packaging and managing Kubernetes-native applications and software. An Operator is itself a piece of software that offers a specific application or workload on demand or as a service, as well as the automation that surrounds the ongoing management of that application or workload.

Main objective

Kubernetes Operators codify operational knowledge about applications and software. They can automate both day 1 and day 2 operations and react to cluster events to deliver a complete life-cycle experience for users. Developers can define stepwise automation for all types of applications and for complex operations.

How do Operators work?

Operators extend the capabilities of Kubernetes through custom resource definitions that describe applications. They can automate nearly any task, including those beyond Kubernetes' core capabilities, to provide a complete life-cycle management experience.

Once deployed, Operators continuously monitor your cluster for predefined trigger events like upgrade requests, reconfigurations, and low performance levels. When a trigger event occurs, Operators react according to their programming. They also reconcile configurations to ensure applications and workloads remain in the desired state.

Where do Operators run?

Operators run in your Kubernetes cluster with direct integration into the Kubernetes control plane. They can respond to cluster events and connect to external services and APIs.

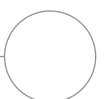
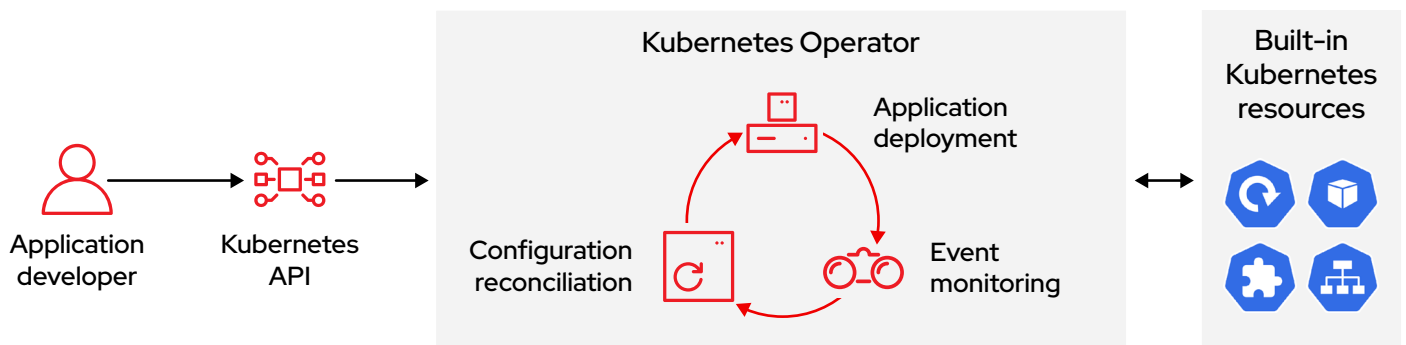
How are Operators managed?

Operators can be managed via a catalog or **Operator Life-cycle Manager (OLM)**, a service that runs in your Kubernetes cluster and provides a direct catalog of all available Operators. OLM also manages the life cycles of the Operators themselves. It offers version control and automated updates for Operators. OLM also automatically redeploys all application and workload instances when associated Operators are updated.

What can you accomplish with Operators?

Operators deliver a complete application and infrastructure software life-cycle management experience:

- ▶ Automate application and software updates and upgrades.
- ▶ Reduce management time and effort.
- ▶ Adopt event-driven automation across your cluster.
- ▶ Increase deployment accuracy and consistency.
- ▶ Free time for developers to focus on high-value tasks.



Creating Helm charts and Kubernetes Operators

Helm charts and Kubernetes Operators require different skills, knowledge, and time investments to create.

Helm charts

Helm charts are relatively easy to create, but can only automate a limited range of tasks. They are coded in YAML, allowing many roles to create and understand the automation within.

- ▶ **Required knowledge:** All components and dependencies needed to install and configure your application in your Kubernetes cluster
- ▶ **Time to create:** Typically hours to days

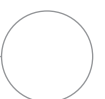
Kubernetes Operators

Operators take more time, effort, and knowledge to create, but can automate all aspects of application life cycles. They are typically written by application developers or others that have a deep understanding of how the application works. Operators require detailed logic and code and are often written in Go, Helm, or as Ansible playbooks.

- ▶ **Required knowledge:** Detailed application behavior throughout its life cycle, as well as all components that are part of your application
- ▶ **Time to create:** Typically weeks to months

Comparison of Helm and Kubernetes Operators capabilities

	Helm Share and deploy applications and software	Kubernetes Operators Deliver a complete life-cycle experience
Packaging	✓	✓
Installation	✓	✓
Updates using Kubernetes manifests	✓	✓
Upgrades using data migration and sequential tasks		✓
Backup and recovery		✓
Autotuning and self-healing with workload and log analysis		✓
Integration with external cloud services and APIs		✓
Event-based automation		✓
Stepwise automation		✓



Choose the right tool for your objectives

Helm and Kubernetes Operators are similar in some aspects, but very different in others. Most organizations will use both Helm and Operators for specific purposes, applications, and infrastructure software within their environment. When creating automation for your application and software life cycles, use Helm and Kubernetes Operators according to these recommendations.

Helm

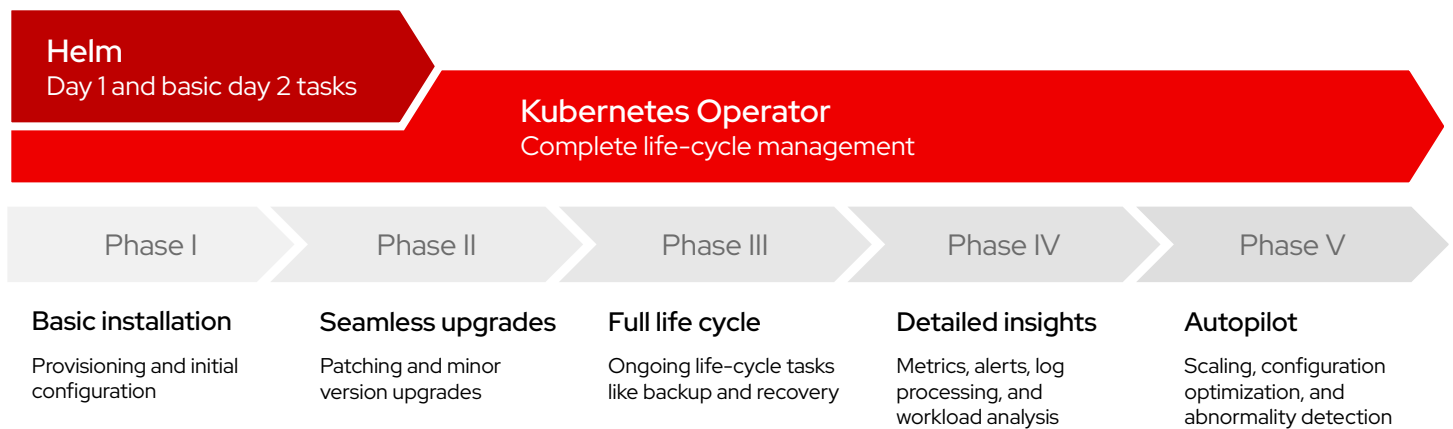
Use Helm for applications and software that:

- ▶ Require automation of only day 1 and simple day 2 tasks.
- ▶ Can be modeled using only core Kubernetes capabilities.
- ▶ Do not need automation of advanced day 2 tasks.

Kubernetes Operators

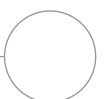
Use Operators for applications and software that:

- ▶ Require a complete, integrated life-cycle management experience, including automation of day 1 and day 2 tasks.
- ▶ Require timed or stepwise automation or conditional logic to manage life-cycle tasks.
- ▶ Require or benefit from event-based automation.
- ▶ Cannot be modeled in Helm charts without manual steps.



Tip:

Watch the [product manager office hours video](#) to learn more about the similarities and differences between Helm and Operators.



Gain more choice with Red Hat OpenShift

Red Hat® OpenShift® is a leading enterprise Kubernetes platform, with full-stack automated operations to manage hybrid, multicloud, and edge deployments, optimized for developer productivity and innovation without limitation. It supports both Helm and Kubernetes Operators, giving you more choice and a path from Helm to Operators as your organization grows and changes. You can use the included **operator software development kit (SDK)** to convert existing Helm charts into Red Hat OpenShift operators and expand their functionality beyond core Kubernetes capabilities.

Red Hat partners with a broad ecosystem of independent software vendors (ISVs) that offer more than 175 **certified operators** and Helm charts for Red Hat OpenShift. Deploy and manage partner applications more easily and confidently with curated assets that encompass detailed vendor expertise. Access Red Hat OpenShift operators developed and supported by the community through **OperatorHub.io**, a centralized repository launched by Red Hat, Amazon, Microsoft, and Google.

Operators are a core part of Red Hat OpenShift. In fact, Red Hat OpenShift itself is deployed and managed using operators. The Red Hat OpenShift user experience incorporates extensive Kubernetes knowledge and experience to make it simpler to use. You can also take advantage of included tools and services – like the operator SDK and OLM – to streamline Red Hat OpenShift operator development and deployment.

Using Helm with Red Hat OpenShift

Red Hat OpenShift fully supports Helm charts.

- ▶ Select and install Helm charts from the developer catalog using the Red Hat OpenShift web console.
- ▶ Create, update, roll back, and uninstall Helm releases using your installed Helm charts.
- ▶ Build and configure custom Helm chart repositories within your Red Hat OpenShift cluster.
- ▶ Visualize, browse, and manage Helm project information directly from the Red Hat OpenShift web console.

Using Red Hat OpenShift operators

Operators are natively integrated into Red Hat OpenShift.

- ▶ Access and install certified operators from the **Red Hat Ecosystem Catalog**.
- ▶ Purchase and deploy third-party operators from the **Red Hat Marketplace**.
- ▶ Install, configure, update, and manage operators across clusters through OLM.
- ▶ Build, test, and package operators more simply and easily using the included operator SDK.



“The operators really allowed us to offer infrastructure-as-a-service to the dev teams almost immediately, on almost day 1, reducing the need for a very large team.”²

Andrew Harrison
Lead IT DevOps Engineer, Omnitrac

² Red Hat OpenShift Commons presentation. “OpenShift @ Omnitrac: A Case Study,” 18 November 2019.



Realize business value through automation



"With built-in operators, we shortened our deployment time from two hours to 40 minutes, managed and extended the cluster, and further reduced costs by using the CoreOS model. We also increased the maturity of our agile processes, making true CI/CD [continuous integration/continuous deployment] pipelines possible."

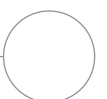
Andrew Harrison
Lead IT DevOps Engineer, Omnitracs



"The operators have significantly simplified the work through the integration and automation that is provided in this platform [OpenShift]. All this integration is deployed and orchestrated in a very connected way within the platform and this gives us the ability to deploy on-premise as well as off-premise."

Francesco Giannoccaro
Head of HPC and Infrastructure, Public Health England

Organizations across industries are realizing measurable business value using Helm and Kubernetes Operators. See how one financial services company streamlined their Red Hat OpenShift developer experience with Helm.





“PGO, the open source Postgres Operator from Crunchy Data, provides a production-grade, cloud-native PostgreSQL experience across the hybrid cloud through Red Hat OpenShift. Our joint users and customers don’t have to worry about critical functions like high availability, disaster recovery, and monitoring. With our Level 5 operator deployed on Red Hat OpenShift, day 2 operations like software upgrades and resource reallocations are automated and can occur with minimal disruption, allowing users to focus on delivering innovative solutions with Red Hat OpenShift.”

Jonathan Katz
VP Platform Engineering, CrunchyData



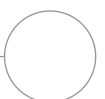
“We are pleased to be part of Red Hat’s hybrid cloud ecosystem, as a provider of a certified application for Red Hat OpenShift. The HashiCorp Vault Helm chart with Red Hat OpenShift support allows applications with no native Vault logic to use static and dynamic secrets sourced from Vault. The Helm chart makes it easier to run Vault on Red Hat OpenShift, and gives you a repeatable deployment process in less time.”

Burzin Patel
Vice President, Global Alliances, HashiCorp



Tip:

Find a complete list of certified partner operators and Helm charts in the [Red Hat Ecosystem Catalog](#).



Ready to start automating?

Automation is at the core of cloud-native application development and deployment. Increase speed, efficiency, and scale and deliver a better user experience by automating application and software management processes. Red Hat OpenShift supports the automation technologies you need to deliver more value to your organization.



Learn more about Helm:
openshift.com/learn/topics/helm



Learn more about Kubernetes Operators:
openshift.com/learn/topics/operators

Take your cloud-native development journey further.

Red Hat experts can help you, your team, and your organization develop the practices, tools, and culture needed to more efficiently modernize existing applications and to build new ones.

Learn more:
redhat.com/en/services/consulting/cloud-native-development