Rancher 2.5:
Technical Architecture

October 2020

Contents

1 Background

According to 451 Research[1], 76 percent of enterprises will standardize on Kubernetes within the next three years because it promises a consistent set of capabilities across any infrastructure – from data center to cloud to the Edge.

By unifying their IT operations with Kubernetes, enterprises can realize dramatic benefits, including:

- Consistently deliver a high level of reliability on any infrastructure
- Improve DevOps efficiency with standardized automation
- Ensure enforcement of security policies on any infrastructure

However, relying on upstream Kubernetes alone can introduce overhead and risk because Kubernetes clusters are typically deployed:

- Without central visibility
- Without consistent security policies
- And, they must be managed independently

Rancher 2.5 is a Kubernetes management platform that addresses these challenges by delivering the following key functions:

- **Consistent Cluster Operations** – simplified Kubernetes upgrades, backups and deployments, from Data Center to the Edge.
- **Security Policy & User Management** – Consistent RBAC, PSP and user management.
- **Shared Tools & Services** – Out-of-the-box access to tools and services.

[1] "Kubernetes and Beyond – Effective Implementation of Cloud Native Software in the Enterprise" by Jay Lyman, Principal Analyst 451 Research – Download White paper

2 Rancher 2.5: Built for Production-Grade Kubernetes

Rancher is a complete container management platform built on Kubernetes. As illustrated in Figure 1, our recipe for production-grade Kubernetes management at scale consists of five primary components: a certified Kubernetes distribution (including Rancher's RKE and K3s), consistent cluster operations, security/authentication/policy management, shared tools and services, and developer services.

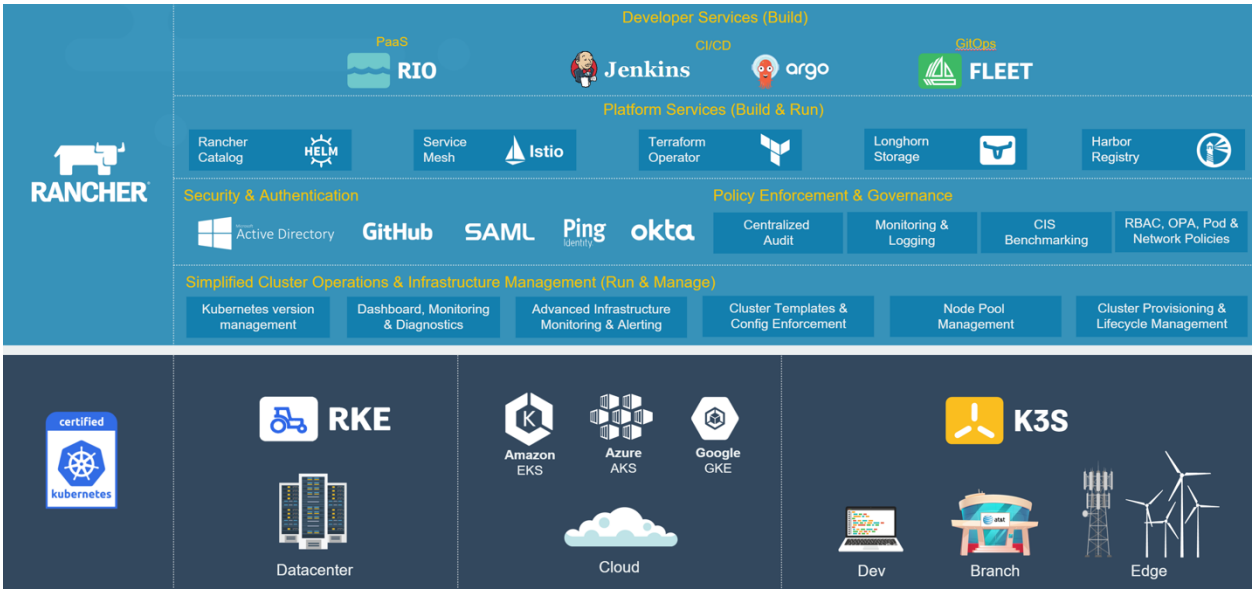


Figure 1: Overview of Rancher's recipe for production-quality Kubernetes at scale

2.1 Certified Kubernetes Distributions

2.1.1 Rancher Kubernetes Engine (RKE)

RKE is a straightforward, lightning-fast Kubernetes installer that works everywhere. RKE is particularly useful in standing up Kubernetes clusters on VMware clusters, bare metal servers and VM instances on clouds that do not yet support a Kubernetes service. In addition, many people use RKE in cloud providers that already support Kubernetes services so that they have a consistent Kubernetes implementation everywhere. In Rancher 2.5, clusters can be provisioned on Linux x86_64 and Arm64 architectures and Windows 20.04 systems.

RKE within Rancher manages the complete lifecycle of Kubernetes clusters from initial install to ongoing maintenance. Rancher users can:

a. Automate VM instance provisioning on many clouds using machine drivers.
b. Install Kubernetes control plane and etcd database nodes.
c. Provision worker nodes on Windows and Linux Arm64 and x86_64 nodes.
d. Add or remove nodes in existing Kubernetes clusters.
e. Upgrade Kubernetes clusters to new versions.
f. Monitor the health of Kubernetes clusters.

For more information about RKE, visit https://rancher.com/products/rke/

2.1.2 K3s – Lightweight Kubernetes Distribution Built for IoT & the Edge

K3s is packaged as a single binary, which is about 50 megabytes in size. Bundled in that single binary is everything needed to run Kubernetes anywhere, including low-powered IoT and Edge-based devices. The binary includes the container runtime and any important host utilities like iptables, socat and du. The only OS dependencies are the Linux kernel itself and a proper dev, proc and sysfs mounts (this is done automatically on all modern Linux distributions).

K3s bundles the Kubernetes components (kube-apiserver, kube-controller-manager, kube-scheduler, kubelet, kube-proxy) into combined processes that are presented as a simple server and agent model. K3s can run as a complete cluster on a single node or can be expanded into a multi-node cluster.

Besides the core Kubernetes components, we also run containerd, Flannel, CoreDNS, ingress controller and a simple host port-based service load balancer. All of these components are optional and can be swapped out for your implementation of choice. With these included components, you get a fully functional and CNCF-conformant cluster so you can start running apps right away. K3s is now a CNCF Sandbox project, being the first Kubernetes distribution ever to be adopted into sandbox.

Learn more information about K3s at https://k3s.io

2.2 Consistent Cluster Operations

With Rancher 2.5, you can choose to manage your own existing Kubernetes clusters provisioned with existing tools or use Kubernetes clusters managed by a cloud. Kubernetes services like EKS, GKE and AKS can easily be provisioned or imported into your Rancher installation. In addition, you can provision and operate RKE Kubernetes clusters on any cloud, virtualized or bare metal infrastructure.

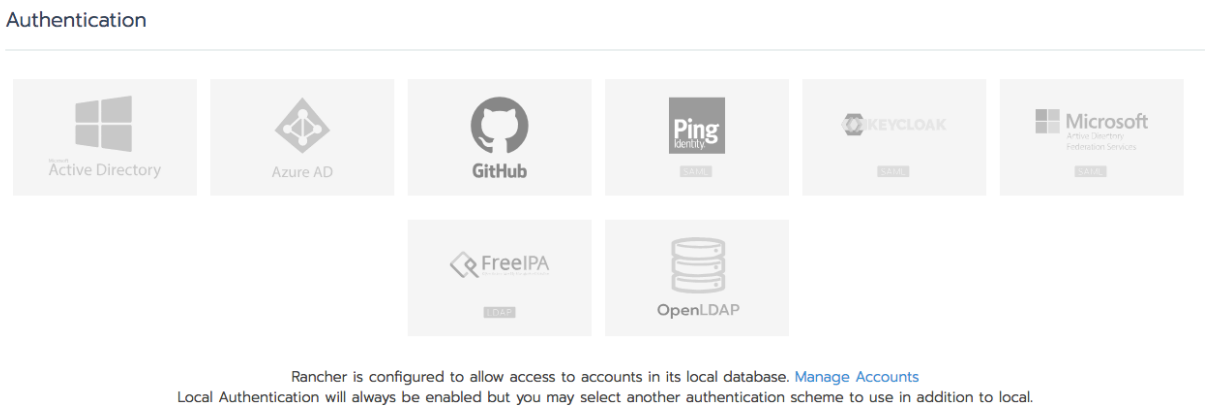
Rancher can easily be managed with Infrastructure-as-code with the Rancher 2.0 Terraform provider. From this you can easily store your configurations for clusters, namespaces, secrets and catalog apps in Git. Rancher has a robust API that can be scripted against to perform routine tasks.

2.3 Security, Authentication, Policy and User Management

Rancher admins can work with their security teams to centrally define how users should interact with Kubernetes and how containerized workloads should operate across all their infrastructures, including managed cloud providers like AKS, EKS and GKE. Once centralized policies are defined, assigning them to any Kubernetes cluster is instantaneous.

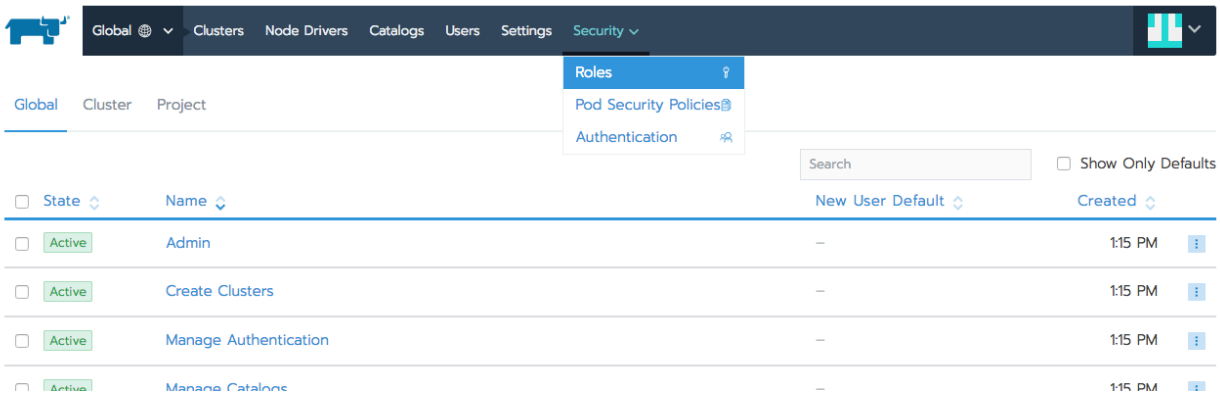
2.3.1 Authentication & RBAC

Rancher not only installs secure clusters, but it proxies all communication to those clusters through the Rancher server. Rancher plugs into several backend authentication providers, such as Active Directory, LDAP, SAML, GitHub and more. When connected in this way, Rancher enables you to extend your existing corporate authentication out to all of the Kubernetes clusters under Rancher's umbrella, no matter where they're running.



Rancher enables roles at the global, cluster and project level, and it makes it possible for administrators to define roles in a single place and apply them to all clusters.

This combination of RBAC-by-default and strong controls for authentication and authorization means that from the moment you deploy a cluster with Rancher or RKE, that cluster is secure.



2.3.2 Rancher's Latest Security Features

A recent StackRox report[2] confirmed that security incidents are widespread in container environments, with 94 percent of survey respondents experiencing incidents in the past year. Of these respondents, 69 percent put these incidents down to misconfiguration errors. Rancher's most recent releases include features to address this key vulnerability:

- 'Cluster Templates' allow operators to create, save and confidently reuse well-tested Kubernetes configurations across all of their cluster deployments. These templates leverage controls and best practices from the most recent Kubernetes Benchmarks from the Center for Internet Security (CIS). The Cluster Templates feature also includes an option for policy enforcement, which prevents configuration drift and assures that the clusters you deploy do not accidentally introduce security vulnerabilities as you scale.

[2] The State of Container and Kubernetes Security Report - Winter 2020 by StackRox – Download White paper

- 'CIS Scan' enables security and operations teams to automatically identify misconfiguration errors by comparing their cluster settings with best practice guidance in the CIS (Center for Internet Security) Kubernetes Benchmark. When Rancher runs a CIS Security Scan on a cluster, it generates a report showing the results of each test, including a summary with the number of passed, skipped and failed tests. The report also includes remediation steps for any failed tests.

2.4 Shared Tools & Services

Rancher 2.5 UI does not attempt to hide the underlying Kubernetes concepts and introduces an application deployment framework different from Kubernetes. Rancher provides an easy-to-use UI for native Kubernetes resources like pods and deployments.

The app catalog experience in Rancher 2.5 is based on Helm charts. Helm is a powerful templating mechanism for deploying applications on Kubernetes. But users still need to read through lengthy documentation to understand exactly what variables to set and the correct values for these variables. This is an error-prone process. Rancher simplifies Helm chart deployment by exposing just the right set of variables and guiding the user through the process. Rancher catalog shows the user by asking the right questions and presenting sensible defaults and multiple-choice values. Rancher supports Helm 3 catalogs as well as git-based catalog repos.

Rancher 2.5 works with any CI/CD systems that integrate with Kubernetes. For example, Jenkins, Drone, and GitLab will continue to work with Rancher 2.5 as they do with any other Kubernetes distribution. If the CI/CD system doesn't natively support Kubernetes, then the Rancher CLI can be embedded as a task to allow for deployments to Rancher managed Kubernetes clusters.

Rancher 2.5 works with any monitoring and logging systems that integrate with Kubernetes. For an out-of-the-box experience, users can use the built-in Prometheus functionality. If existing systems like Datadog, Sysdig or ELK are in place, they will continue to work with Rancher 2.5. For log aggregation, Rancher provides simple click deployment of Fluentd and Fluent Bit that will ship logs from the hosts.

2.5 Fleet

Fleet is a Kubernetes cluster controller specifically designed to address the challenges of running thousands to millions of clusters worldwide. While it's designed for massive scale, the concepts still apply for even small deployments of less than 10 clusters. Fleet is lightweight enough to run on the smallest of deployments and even has merit in a single-node cluster managing only itself. The primary use case of Fleet is to ensure that deployments are consistent across clusters. You can deploy applications or easily enforce standards such as "every cluster must have X security tool installed."

3 High-level Architecture

3.1 Rancher Server

Rancher 2.5 has server components that manage the entire Rancher deployment and deploys agent components into Kubernetes clusters.

Figure 2 illustrates the high-level architecture of Rancher 2.5. It depicts a Rancher server installation that manages two Kubernetes clusters: one Kubernetes cluster created by RKE and another non-RKE Kubernetes cluster that could be EKS, AKS, GKE or any other Kubernetes cluster.

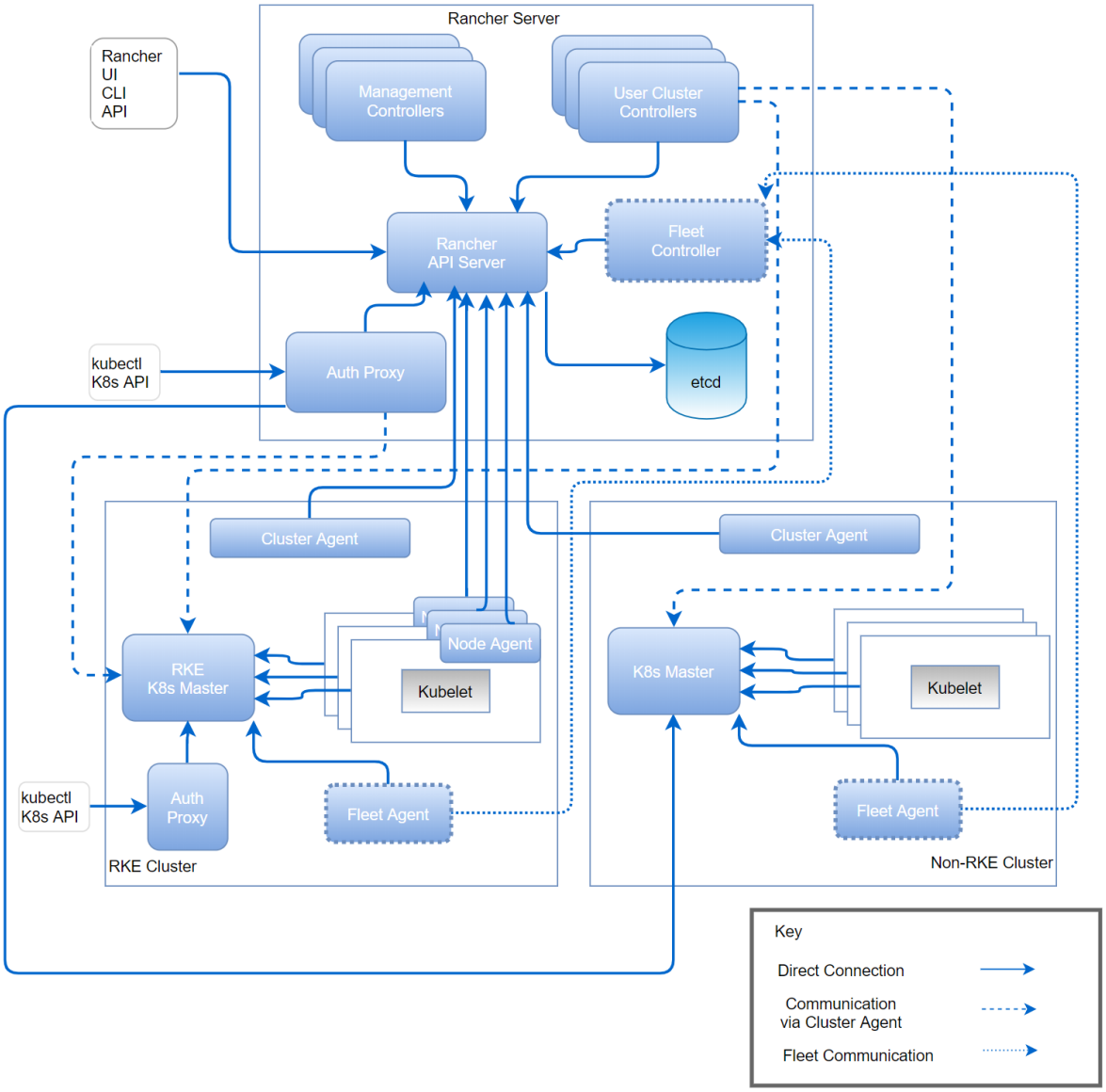


Figure 2 Rancher 2.5 High-Level Architecture

3.2 Fleet

Fleet has two simple high-level concepts:

- Cluster groups: A logical group of clusters that need to be targeted as a single entity.
- Bundles: Collections of resources that are deployed to clusters.

Bundles are defined in the Fleet controller and are then deployed to target clusters using selectors and per-target customization. While bundles can be deployed to any cluster using powerful selectors, each cluster is a member of one cluster group. By looking at the status of bundles and cluster groups, one can get a quick overview of large deployments' status. After a bundle is deployed, it is monitored continuously to ensure that it is ready and resources have not been modified.

A bundle can be plain Kubernetes YAML, Helm or kustomize based. Helm and kustomize can also be combined to create powerful workflows. Regardless of the approach you choose to create bundles, all resources are deployed to a cluster as Helm charts. Using Fleet to manage clusters means all your clusters are easily auditable because every resource is carefully managed in a chart and a simple `helm -n fleet-system ls` will give you an accurate overview of what is installed. By combining Fleet with a Git-based workflow like Github Actions, you can automate at massive scale with ease.

4 Rancher Server Components

In this section, we describe the functionalities of each Rancher server component.

4.1 Rancher API Server

Rancher Server provides a robust API. Rancher uses the persistent datastore of the underlying Kubernetes instance that it runs on, typically etcd, to store all configuration data. All Rancher specific resources created using the Rancher API get translated to CRD (Custom Resource Definition) objects, with their lifecycle being managed by one or several Rancher controllers.

Rancher API Server is the foundational layer for all controllers in the Rancher server. It includes the following functionalities:

- User-facing API schema generation with an ability to plug custom formatters and validators.
- Controller interfaces generation for CRDs and native Kubernetes object types.
- Object lifecycle management framework.
- Conditions management framework.
- Simplified generic controller implementation by encapsulating TaskQueue and SharedInformer logic into a single interface.

4.2 Management Controllers

The management controllers perform activities at the Rancher server level, not specific to an individual cluster. These activities include:

a. Configuring access control policies to clusters and projects.
b. Managing pod security policy templates.

c. Provisioning clusters by invoking the necessary Docker machine drivers and invoking Kubernetes engines like RKE and GKE.
d. Managing users – CRUD (Create, Read, Update and Delete) operations on users.
e. Managing global-level catalog, fetch content of the upstream Helm repo, etc.
f. Managing cluster and project-level catalogs.
g. Aggregating and displaying cluster stats and events.
h. Managing of node drivers, node templates and node pools.
i. Managing cluster cleanup when cluster is removed from Rancher.

4.3 User Cluster Controllers

User cluster controllers perform activities specific to a cluster. User cluster controllers are spread out across the running Rancher server pods for horizontal scaling. Activities include:

a. Managing workloads, which includes, for example, creating pods and deployments in each cluster.
b. Applying roles and bindings that are defined in global policies into every cluster.
c. Propagating information from cluster to Rancher server: events, stats, node info and health.
d. Managing network policies.
e. Managing alerts, monitoring, log aggregation and CI/CD pipelines.
f. Managing resource quota.
g. Propagating secrets down from Rancher server to individual clusters.

User cluster controllers connect to API servers in GKE clusters directly, but tunnel through the cluster agent to connect to API servers in RKE clusters.

4.4 Authentication Proxy

The authentication proxy proxies all Kubernetes API calls. It integrates with authentication services like local authentication, Active Directory, and GitHub. On every Kubernetes API call, the authentication proxy authenticates the caller and sets the proper Kubernetes impersonation headers before forwarding the call to Kubernetes masters. Rancher communicates with Kubernetes clusters using a service account.

The authentication proxy connects to API servers in Non-RKE clusters directly, but tunnels through the cluster agent to connect to API servers in RKE clusters.

In Rancher 2.2, the authentication cluster endpoint was introduced into RKE based clusters to bring centralized auth to the local cluster. This provides increased availability by removing the Rancher from the authentication path, allowing disconnected management and operations of your Kubernetes clusters.

4.5 Fleet Manager

The Fleet Manager is responsible for pulling the bundles and definitions from a Git repository. There is only a single Fleet Manager per Rancher Management Server installation.

The Fleet Manager fulfills requests from the Fleet Agents.

5 Rancher Agent Components

In this section, we describe software components deployed in Kubernetes clusters managed by Rancher.

5.1 Cluster Agents

Rancher deploys one cluster agent for each Kubernetes cluster under management. The cluster agent opens a WebSocket tunnel back to Rancher server so that the user cluster controllers and authentication proxy can communicate with the user cluster Kubernetes API server. Note that only RKE clusters and imported clusters utilize the cluster agent to tunnel Kubernetes API. Cloud Kubernetes services like GKE already expose API endpoints on the public Internet and therefore do not require the cluster agent to function as a tunnel.

Cluster agents serve two additional functions:

a. They serve as a proxy for other cluster services, like Rancher's built-in alert, log aggregation and CI/CD pipelines. Any services running in user clusters can be exposed through the cluster agents. This capability is sometimes called "the magic proxy."
b. During registration, cluster agents get service account credentials from the Kubernetes cluster and send the service account credentials to the Rancher server.

5.2 Node Agents

Node agents are primarily used by RKE to deploy the components during the initial install and follow-on upgrades. Node agents are not deployed on cloud Kubernetes clusters like GKE. Node agents serve several additional functions for all clusters:

a. Fallback for cluster agents: if the cluster agent is not available for any reason, Rancher server will use the node agent to connect to the Kubernetes API server.
b. Proxy for `kubectl` shell. Rancher server connects through node agents to tunnel the `kubectl` shell in the UI. Node agent runs with more privileges than a cluster agent, and that additional privilege is required to tunnel the `kubectl` shell.

5.3 Fleet Agent

The Fleet Agents makes calls to the Fleet Manager and pulls its specifics as a BundleDeployment.

The Fleet Agent does not have to have a constant connection to the Fleet Manager. When the connection is next present, the agent will reconcile with the manager, making this ideal for scenarios where network connections can be inconsistent.

6 Upgrade

Users can upgrade previous versions of Rancher to Rancher 2.5 by upgrading the Rancher server via a Helm upgrade. Rancher 2.5 will automatically upgrade Rancher agents in child clusters. Users will then have the option to upgrade the underlying Kubernetes versions of RKE and K3s clusters to take advantage of new functionality.

In Rancher 2.5, the integration of EKS clusters has been improved, allowing for full lifecycle management of the EKS cluster including more granular control over instantiation, the ability upgrade the underlying Kubernetes version when a new one is available and also the destruction of the EKS cluster.

7 High Availability

Users may use a dedicated RKE cluster to run the Rancher server. The standard Rancher 2.5 installation guide, for example, creates an RKE deployment with 3 nodes, each running one instance of the API server and the etcd database. Rancher server automatically imports the Kubernetes cluster it runs on. It is called "the local cluster." Rancher will leverage the Kubernetes API and indirectly use that clusters etcd as the primary datastore.

Information on installation can be found at https://rancher.com/docs/rancher/v2.x/en/installation/

8 Scalability

8.1 Scalability of Kubernetes Clusters

Users can expect Rancher 2.5 to manage and provision RKE clusters up to 100,000 nodes.

8.2 Scalability of Rancher Server

There is no inherent limit on how many Kubernetes clusters each Rancher server can manage. We do not expect an issue for Rancher 2.5 to manage up to 10,000 clusters.

The real scalability limits of Rancher server are:

a. Total nodes across all clusters
b. Users and groups
c. Events collected from all clusters

Rancher server stores all the above entities in the underlying Kubernetes etcd database. We will improve scalability along these dimensions over time to meet user needs.

8.3 Scalability of Fleet

Fleet is based on a Gitops model so uses code to define state. This is much more efficient in management overhead as there are limitations on how to visualize thousands of clusters in a GUI effectively. Fleet can theoretically scale to one million+ clusters.

9 About Rancher Labs

Rancher Labs delivers open source software that enables organizations to deploy and manage Kubernetes at scale, on any infrastructure across the data center, cloud, branch offices and the network edge. With +35,000 active users and +100 million downloads, their flagship product, Rancher, is the industry's most widely adopted Kubernetes management platform.

For additional information, visit www.rancher.com and follow @Rancher_Labs on Twitter.

All product and company names herein may be trademarks of their registered owners.