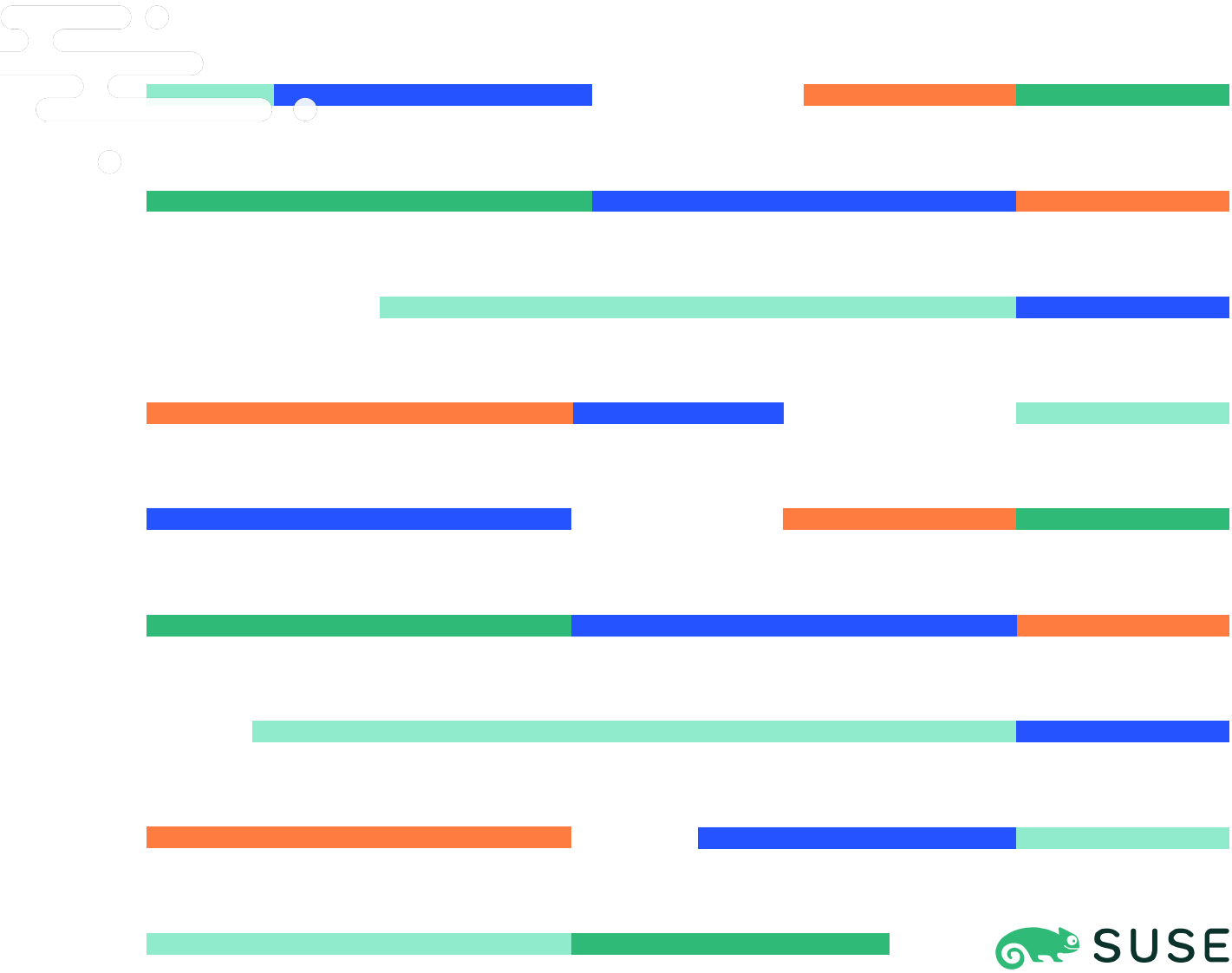
SUSE Rancher 2.6: Technical Architecture Guide

August 2021



SUSE

Contents

1 Background ... 3

2 SUSE Rancher 2.6: Built for Enterprise Production-Grade Kubernetes ... 4

2.1 Certified Kubernetes Distributions ... 4

2.2 Consistent Cluster Operations ... 5

2.3 Security, Authentication, Policy and User Management ... 6

2.4 Shared Tools & Services ... 7

2.5 Fleet ... 8

3 High-level Architecture ... 8

3.1 Fleet ... 8

3.2 Rancher Server ... 9

4 Rancher Server Components ... 10

4.1 Rancher API Server ... 10

4.2 Management Controllers ... 10

4.3 User Cluster Controllers ... 10

4.4 Authentication Proxy ... 11

4.5 Fleet Manager ... 11

5 SUSE Rancher Agent Components ... 11

5.1 Cluster Agents ... 11

5.2 Node Agents ... 12

5.3 Fleet Agent ... 12

6 Upgrade ... 12

7 High Availability ... 13

8 Scalability ... 13

8.1 Scalability of Kubernetes Clusters ... 13

8.2 Scalability of Rancher Server ... 13

8.3 Scalability of Fleet ... 13

9 About SUSE ... 14

1 Background

A recent Forrester Wave report[1] stated cloud-native technologies are fast becoming the preferred way for global organizations to build and modernize their applications and services at scale. The popularity of containers and Kubernetes continues to show with Gartner[2] predicting by 2022, more than 75 percent of worldwide organizations will run containerized applications in production. This forecasted growth demonstrates the value of cloud-native technologies like Kubernetes for enterprise developers and IT operators, looking for solutions to help to build applications faster and manage environments without compromising on reliability, agility and security.

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

SUSE Rancher 2.6 is a Kubernetes management platform that addresses these challenges by delivering the following key functions:

- **Consistent Cluster Operations** – simplified Kubernetes upgrades, backups and deployments, anywhere from core to cloud and at 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] "The Forrester Wave™: Multicloud Container Development Platforms, Q3 2020" by Dave Bartoletti, Charlie Dai with Lauren Nelson, Duncan Dietz, Han Bao, Bill Nagel, Forrester – Download Report
[2] "Gartner Forecasts Strong Revenue Growth for Global Container Management Software and Services Through 2024" by Susan Moore, Gartner – View Press Release

2 SUSE Rancher 2.6: Built for Enterprise Production-Grade Kubernetes

SUSE Rancher 2.6 is a complete container management platform built on Kubernetes. As illustrated in Figure 1, SUSE Rancher 2.6 consists of four primary components: a certified Kubernetes distribution (including SUSE's RKE and CNCF Sandbox Project, K3s), consistent cluster operations, security/authentication/policy management/governance and developer platform services.

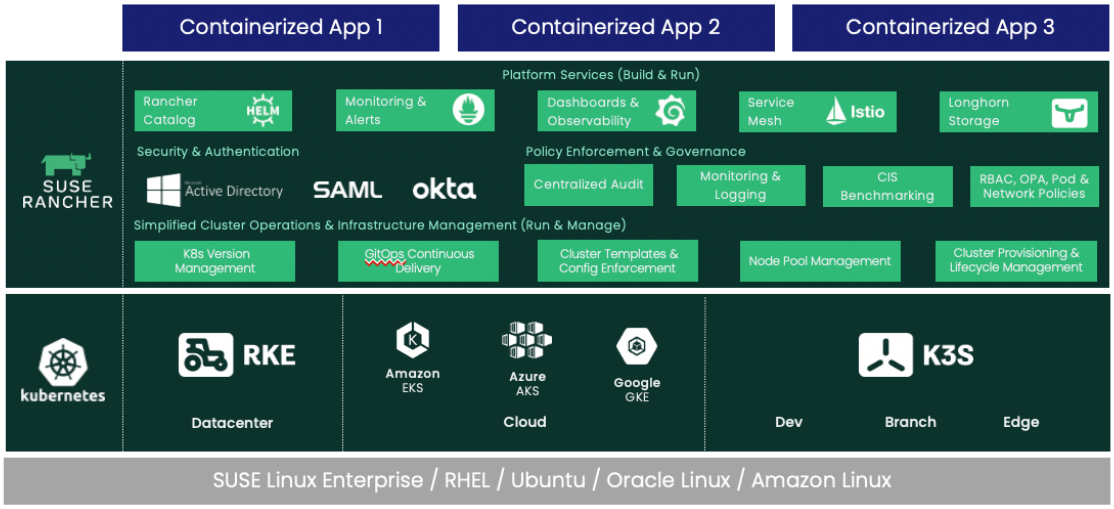


Figure 1: Overview of Rancher's recipe for production-quality Kubernetes at scale

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 SUSE Rancher 2.6, clusters can be provisioned on Linux x86_64 and Arm64 architectures and Windows 20 H2 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 http://suse.com/products/rke

2.1.2 Rancher Kubernetes Engine 2 (RKE2)

RKE2, also known as RKE Government, is a fully conformant certified Kubernetes distribution focused on security and compliance. RKE2 leverages the best components of RKE and K3s distributions to form its anatomy. RKE2 brings government grade security capabilities to the enterprise and cloud native community. It has been built to take advantage of changes across the ecosystem. RKE2 requires no dependency on the Docker container runtime and includes a supported containerd runtime. The distribution supports SELinux and has been compiled with FIPS certified golang libraries.

In SUSE Rancher 2.6, RKE2 provisioning is being introduced as a technical preview. This new provisioning system is built on-top of the community standard Cluster API specifications. Users will now be able to leverage GitOps tools to define their clusters as infrastructure as code out of the box. Additionally, when deploying through SUSE Rancher, RKE2 clusters will default to using the open-source Calico container networking interface (CNI) plugin, as well as options to deploy multiple network interfaces into their pods with Multus. With RKE2 users will be able to provision Windows nodes in custom clusters.

For more information on RKE2, visit https://docs.rke2.io/

2.1.3 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 SUSE Rancher 2.6, you can choose to manage your own existing Kubernetes clusters provisioned with existing tools or use Kubernetes clusters managed by a cloud. Kubernetes services and hosted clusters on EKS, GKE and AKS can easily be provisioned or imported into and managed within your SUSE Rancher installation. In addition, you can provision and operate RKE Kubernetes clusters on any cloud, virtualized or bare metal infrastructure.

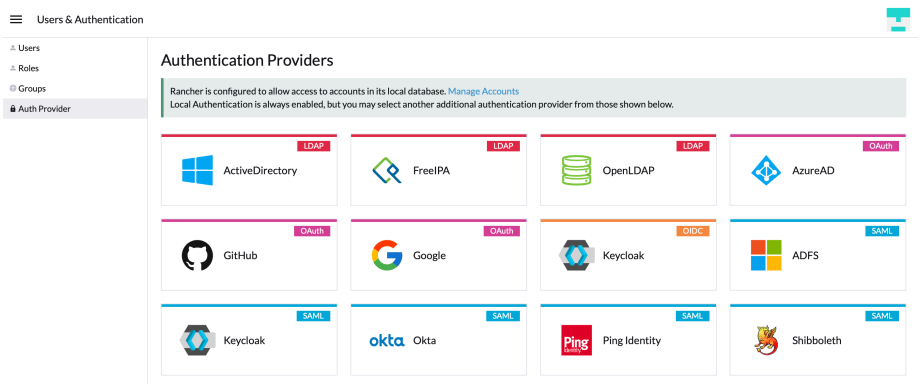
SUSE Rancher can easily be managed with Infrastructure-as-code with the Rancher Terraform provider. From this you can easily store your configurations for clusters, namespaces, secrets and catalog apps in Git. SUSE Rancher has a robust API that can be scripted against to perform routine tasks.

2.3 Security, Authentication, Policy and User Management

SUSE 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 hosted clusters within managed cloud providers like AKS, EKS and GKE. Once centralized policies are defined, assigning them to any Kubernetes cluster is instantaneous.

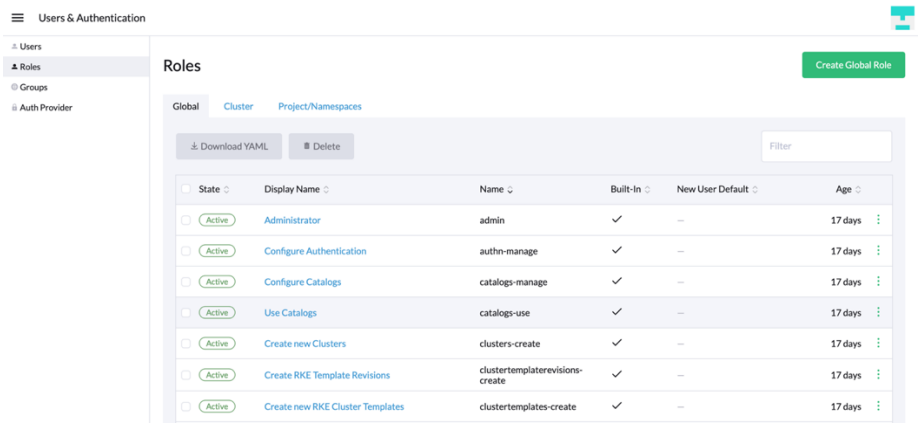
2.3.1 Authentication & RBAC

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



SUSE 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 SUSE Rancher or RKE, that cluster is secure.



2.3.2 SUSE Rancher's Latest Security Features

A recent StackRox report[3] 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. SUSE Rancher's recent releases include features to address this key vulnerability and help operators manage risk including:

- 'User ID Tracking' has been added to audit logs to help users trace events. SUSE Rancher now includes the Identity Provider name in both SUSE Rancher and Kubernetes audit logs. This helps promote the self-service model of SUSE Rancher giving users clarity to identify different owners of clusters.
- 'Image Scanning' for CVEs is now automated across all images as part of releases helping users easily determine if there are any major vulnerabilities across images in their cluster. If any critical vulnerabilities are found, SUSE Rancher has pre-determined actions to help identify, fix and/or mitigate issues.
- 'SLE Base Container Image (SLE BCI)' SUSE Rancher begun adopting SLE BCI as a base image for microservices and allows users access to a secure, open repository.
- 'Cluster Templates' allow operators to create, save and confidently reuse well-tested Kubernetes configurations across 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.
- '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 SUSE 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

SUSE Rancher 2.6 UI does not attempt to hide the underlying Kubernetes concepts and introduces an application deployment framework different from Kubernetes. SUSE Rancher provides an updated crisp UI for native Kubernetes resources like pods and deployments.

The app catalog experience in SUSE Rancher 2.6 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. SUSE Rancher simplifies Helm chart deployment by exposing just the right set of variables and guiding the user through the process. SUSE Rancher catalog shows the user by asking the right

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

questions and presenting sensible defaults and multiple-choice values. SUSE Rancher supports Helm 3 catalogs as well as git-based catalog repos.

SUSE Rancher 2.6 works with any CI/CD systems that integrate with Kubernetes. For example, Jenkins, Drone, and GitLab will continue to work with SUSE Rancher 2.6 as they do with any other Kubernetes distribution. If the CI/CD system doesn't natively support Kubernetes, then the SUSE Rancher CLI can be embedded as a task to allow for deployments to SUSE Rancher managed Kubernetes clusters.

SUSE Rancher 2.6 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 SUSE Rancher 2.6. For log aggregation, SUSE Rancher provides simple click deployment of Fluentd and Fluent Bit that will ship logs from the hosts.

2.5 Fleet

Fleet, an open-source project developed by the SUSE Rancher team, is a Kubernetes cluster controller. It has been 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 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.

3.2 Rancher Server

SUSE Rancher 2.6 has server components that manage the entire SUSE Rancher deployment and deploys agent components into Kubernetes clusters.

Figure 2 illustrates the high-level architecture of SUSE Rancher 2.6. 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.

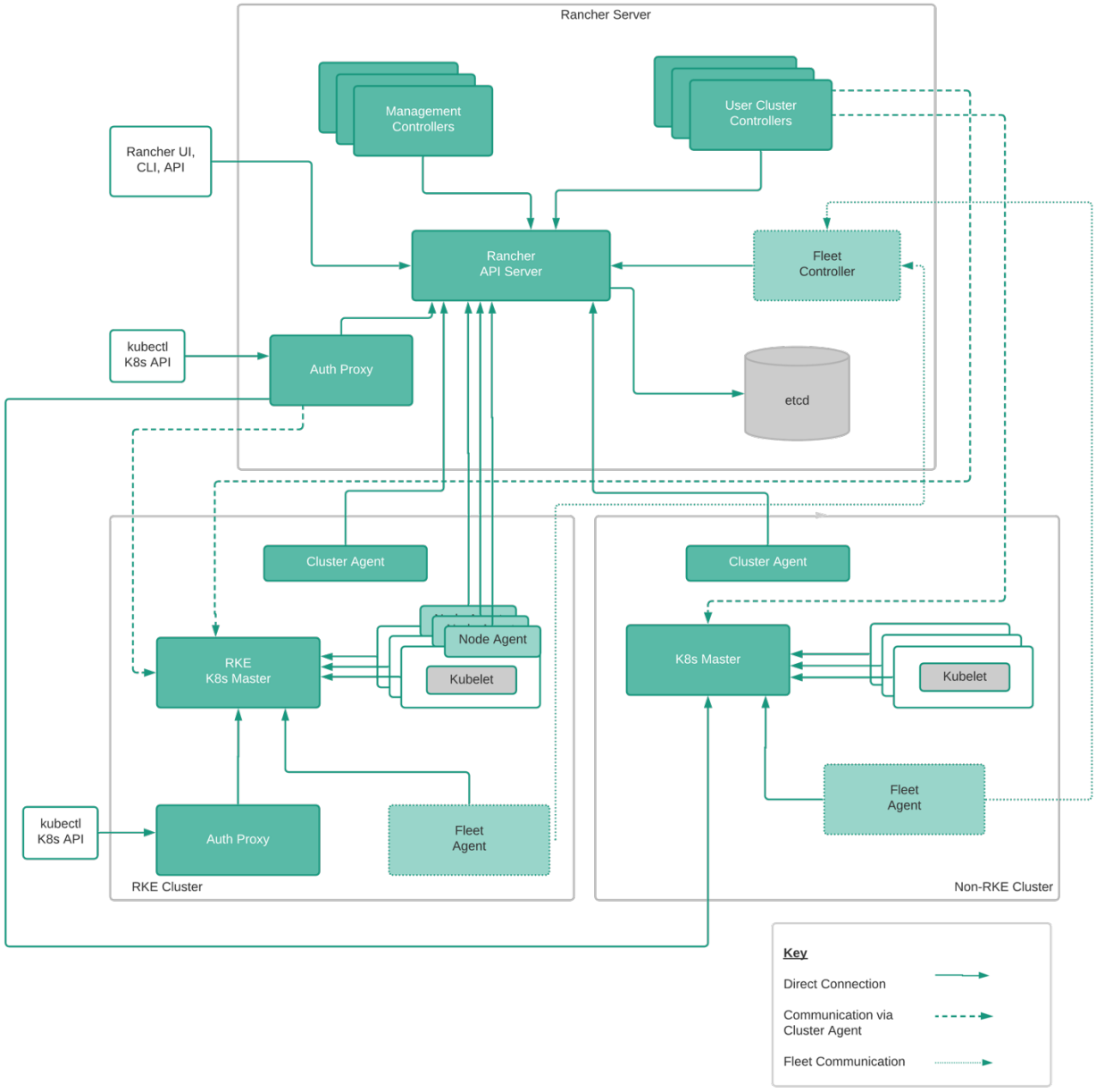


Figure 2: SUSE Rancher 2.6 High-Level Architecture

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. SUSE Rancher uses the persistent datastore of the underlying Kubernetes instance that it runs on, typically etcd, to store all configuration data. All SUSE 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 SUSE 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, Okta 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. SUSE 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.

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 server 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 server installation.

The Fleet Manager fulfills requests from the Fleet Agents.

5 SUSE Rancher Agent Components

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

5.1 Cluster Agents

SUSE 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 SUSE 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 SUSE Rancher to SUSE Rancher 2.6 by upgrading the Rancher server via a Helm upgrade. SUSE Rancher 2.6 will automatically upgrade SUSE 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 SUSE Rancher 2.6, the integration of hosted clusters across EKS, AKS and GKE has been improved, allowing for full lifecycle management of these clusters including more granular control over instantiation, the ability upgrade the underlying Kubernetes version when a new one is available and the destruction of the clusters all within the SUSE Rancher platform. This integration also treats existing hosted clusters same as those deployed in SUSE Rancher minimizing the need for operators to manage multiple platforms across hybrid and multi-cloud environments.

7 High Availability

Users may use a dedicated RKE cluster to run the Rancher server. The standard SUSE Rancher 2.6 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." SUSE 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 SUSE Rancher 2.6 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 SUSE Rancher 2.6 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 up to one million+ clusters.

9 About SUSE

SUSE is a global leader in innovative, reliable, and enterprise-grade open source solutions. SUSE specializes in Enterprise Linux, Kubernetes management, and edge solutions, and the company collaborates with partners and communities around the globe, empowering them to innovate everywhere – from the data center, to the cloud, to the edge and beyond. In 2020, SUSE acquired Rancher Labs, the team behind successful open source products including:

- **Rancher** - the world's most popular enterprise-grade Kubernetes management platform with over +40,000 active users and +120 million downloads.
- **RKE** - a simple, lightning-fast Kubernetes installer that works everywhere;
- **RKE2** – is a fully conformant Kubernetes distribution focused on security and compliance
- **Fleet** – an open source project built to help manage millions of Kubernetes clusters at scale
- **K3s** – a lightweight production-grade Kubernetes distribution built for embedded systems and the edge. Rancher invented K3s and donated it to the CNCF in August, 2020.
- **Longhorn** - a powerful cloud-native distributed storage platform for Kubernetes that can run anywhere. Rancher invented Longhorn and donated it to the CNCF in October, 2019.

All of Rancher's solutions remain open source after the acquisition, with support from a vibrant, active community. SUSE offers an enterprise subscription for some solutions, and those are differentiated by prefixing them with "SUSE," such as in "SUSE Rancher."

Together, these products help IT operators, DevOps, and technology leaders' teams address the operational and security challenges of managing certified Kubernetes clusters across any infrastructure. They also provide developers with an integrated stack of tools to build and run containerized workloads at scale.

To learn more about SUSE Rancher please visit: suse.com/products/suse-rancher