# Rancher 2.7: Technical Architecture Guide

May 2023

Rancher Government Solutions

# 1 Background

Prior to COVID, many US Government agencies had begun implementing Kubernetes as part of their digital transformation taking advantage of the speed, flexibility, and stability that the orchestration platform provided. However, with the onset of the global pandemic, it forced many public and private organizations to rapidly pivot and accelerate their adoption of container and cloud native technology to meet the demands of the changing market. In the annual CNCF (Cloud Native Computing Foundation) survey released in February 2022, it stated that 96% of organizations were already either using or evaluating Kubernetes.

Yet the reality for teams implementing Kubernetes saw compounding challenges from increasingly distributed infrastructure environments to difficulty in the retention and hiring of Kubernetes talent to support their systems. To complicate things further, as container based ecosystem grew, more organizations undertook multi-cloud deployments, increasing the attack surface for data centers significantly and causing a seismic shift in the focus on security, compliance enforcement and monitoring across Kubernetes workloads.

However, Kubernetes still provides multiple benefits for enterprises including:

- A common platform to manage clusters across different infrastructures improving reliability
- Improve DevOps efficiency with standardized automation
- Ensure consistent security and compliance policies and regulation

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

- Without central visibility
- Without consistent security policies and configurations
- Without centralized management

Rancher is a Kubernetes management platform that addresses these challenges and more by delivering the following essential functions:

- **Consistent Cluster Operations** – simplified Kubernetes upgrades, backups, configurations, and deployments, anywhere from core to cloud and at the edge.
- **Multi-cluster Management** – single console to manage Kubernetes clusters anywhere at scale from on-premises and to the cloud.
- **Authentication & User Management** – Consistent RBAC (Role Based Access Control), security policies, and user management.
- **Shared Tools & Services** – Out-of-the-box access to tools and services. Such as unified monitoring, automation, etc.

- **Comprehensive Security** – A DISA (Defense Information Systems Agency) STIG (Security Technical Implementation Guides) [1] certified solution with CIS scans for Kubernetes nodes and clusters. Additional compliance capabilities including audit logging, Open Policy Agent, Gatekeeper, Kubewarden, and NeuVector integrations.

Rancher simplifies Kubernetes management at scale and the operations around it. It helps to improve automation and configuration management while enforcing and securing your clusters with the help of different open source projects and tools. Rancher helps remove the burden of repetitive and tedious tasks from your operations teams while giving more freedom to the developers to build and ship their applications.

# 2 Rancher 2.7: Built for Enterprise Production-Government Grade Kubernetes

Rancher 2.7 is a complete container management platform built on Kubernetes. As illustrated in Figure 1, Rancher 2.7 consists of four primary components: a certified Kubernetes distribution (including RKE2 and K3s), consistent cluster operations, security/authentication/policy management/governance and developer platform services.

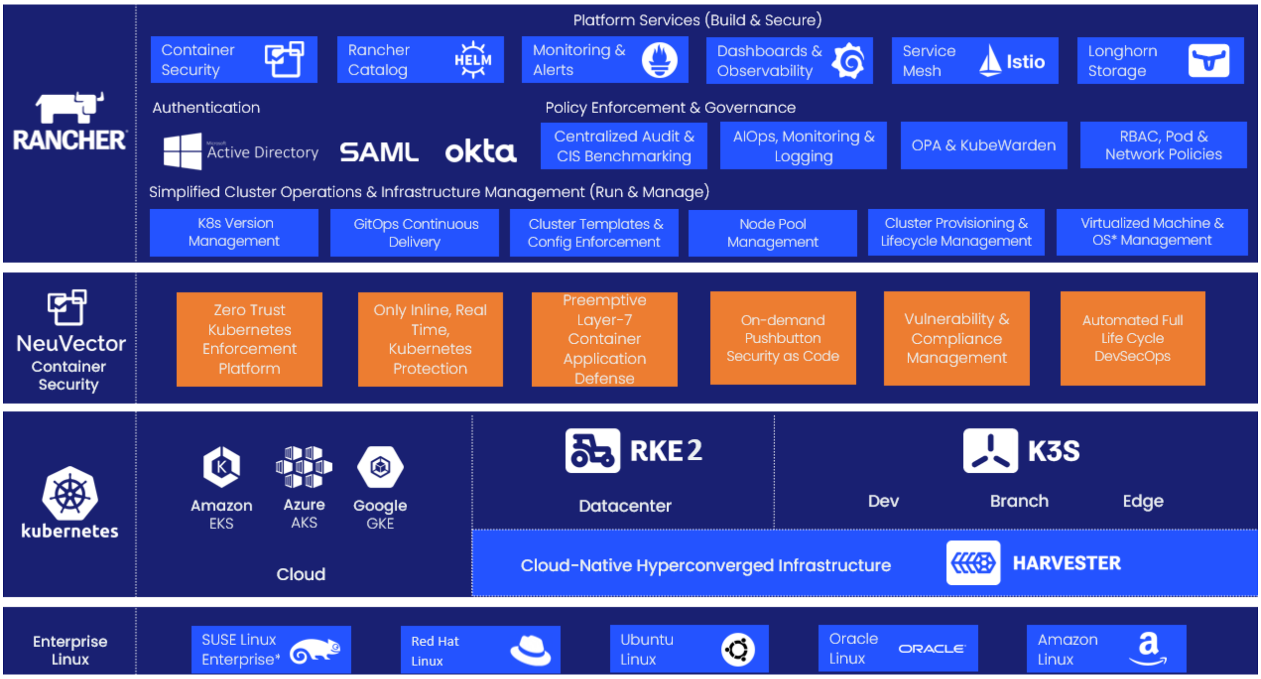

RANCHER
Platform Services (Build & Secure)
Container Security
Rancher Catalog
HELM
Monitoring & Alerts
Dashboards & Observability
Service Mesh
Istio
Longhorn Storage
Authentication
Active Directory
SAML
okta
Policy Enforcement & Governance
Centralized Audit & CIS Benchmarking
AIOps, Monitoring & Logging
OPA & KubeWarden
RBAC, Pod & Network Policies
Simplified Cluster Operations & Infrastructure Management (Run & Manage)
K8s Version Management
GitOps Continuous Delivery
Cluster Templates & Config Enforcement
Node Pool Management
Cluster Provisioning & Lifecycle Management
Virtualized Machine & OS* Management
NeuVector Container Security
Zero Trust Kubernetes Enforcement Platform
Only Inline, Real Time, Kubernetes Protection
Preemptive Layer-7 Container Application Defense
On-demand Pushbutton Security as Code
Vulnerability & Compliance Management
Automated Full Life Cycle DevSecOps
kubernetes
Amazon EKS
Azure AKS
Google GKE
Cloud
RKE2
Datacenter
K3S
Dev
Branch
Edge
Cloud-Native Hyperconverged Infrastructure
HARVESTER
Enterprise Linux
SUSE Linux Enterprise*
Red Hat Linux
Ubuntu Linux
Oracle Linux
ORACLE
Amazon Linux

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

## 2.1 Certified Kubernetes Distribution for Government

### 2.1.1 Rancher Kubernetes Engine 2 (RKE2)

RKE2 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, has been compiled with FIPS (Federal Information Processing Standards) certified golang libraries and it is the only DISA STIG certified Kubernetes distribution.

The provisioning system for RKE2 is built on-top of the community standard Cluster API specifications. Users will now be able to leverage GitOps tools easier to define their clusters as infrastructure as code out of the box. Additionally, when deploying through 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/

For more on Rancher DISA STIGS for RKE2 and Rancher MCM 2.7 visit:

https://dl.dod.cyber.mil/wp-content/uploads/stigs/zip/U_RGS_MCM_V1R2_STIG.zip

### 2.1.2 K3s – Lightweight Kubernetes Distribution Built for IoT (internet of things) & the Tactical 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 and socat. The only OS (Operating System) 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 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.

With the release of Rancher 2.7 deploying K3s is now GA (General Availability) across x86 platforms.

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

## 2.2 Consistent Cluster Operations

With Rancher, you can manage your Kubernetes clusters provisioned with existing tools or use Kubernetes clusters managed by a cloud. Kubernetes services like EKS (Elastic

Kubernetes Service), GKE, and AKS can easily be provisioned or imported into and managed within your Rancher installation. Rancher is the only solution in the market that offers full lifecycle management across these three public hosted solutions. In addition, you can provision and operate RKE2 and K3s clusters on any cloud, virtualized, or bare metal infrastructure.

Rancher can easily be managed as Infrastructure-as-code with the Rancher Terraform provider, using CI/CD pipelines and Rancher's API or using GitOps projects like Fleet. You can easily store your configurations for clusters, cluster templates, policies, and apps in Git and use any strategy that fits your organization's needs.

### 2.2.1 Fleet

Fleet, an open-source project developed by the Rancher team. It addresses the challenges of both GitOps based application delivery and cluster configuration management at scale. Whilst it has been designed for use at large scale, its 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 application deployments and configurations are consistent across clusters. You can deploy applications or easily enforce standards across hundreds of clusters for a true policy as code global strategy.

#### 2.2.1.1 New in Fleet

In the latest Rancher 2.7 release, authentication for **OCI-based registries are now supported**. Note that the structure of the fleet.yaml is the same, and the credentials are provided as a Kubernetes secret which is described in the Private Helm Repo box in the Repo Structure docs.

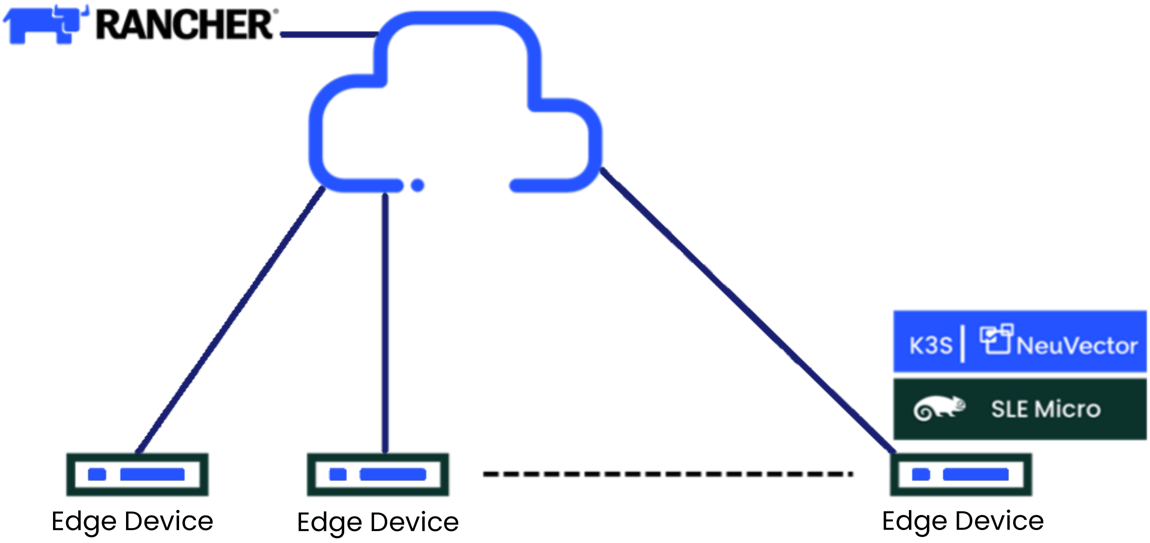
### 2.2.2 Extensions and UI (User Interface) improvements

Rancher 2.7 introduces **extensions**. Admins may now make changes and enhancements to their UI functionality as desired, independent of Rancher Manager releases. Using the Extensions catalog, the admins can view the list of installed extensions, update or roll back existing extensions, and install new extensions as desired. With extensions, operators can extend Rancher's UI and capabilities. Please read the docs for more information on how to use Rancher extensions.

Cluster events have now been moved to a more prominent area, making it easier to find what is happening in the managed clusters. Also, the diagnostics page now shows more detailed information for troubleshooting performance issues. Plus, admins can also add links to documentation from inside the Rancher UI.

### 2.2.3 Edge management with Elemental

In Rancher 2.7, project Elemental can be enabled in the UI via extensions. Elemental is a fully integrated solution for managing the full lifecycle of Edge devices at scale. By leveraging the latest versions of Rancher, SLE Micro and NeuVector, it brings a muchneeded integrated platform that simplifies, centralizes, and automates Kubernetes and Linux OS lifecycle management across distributed edge locations.



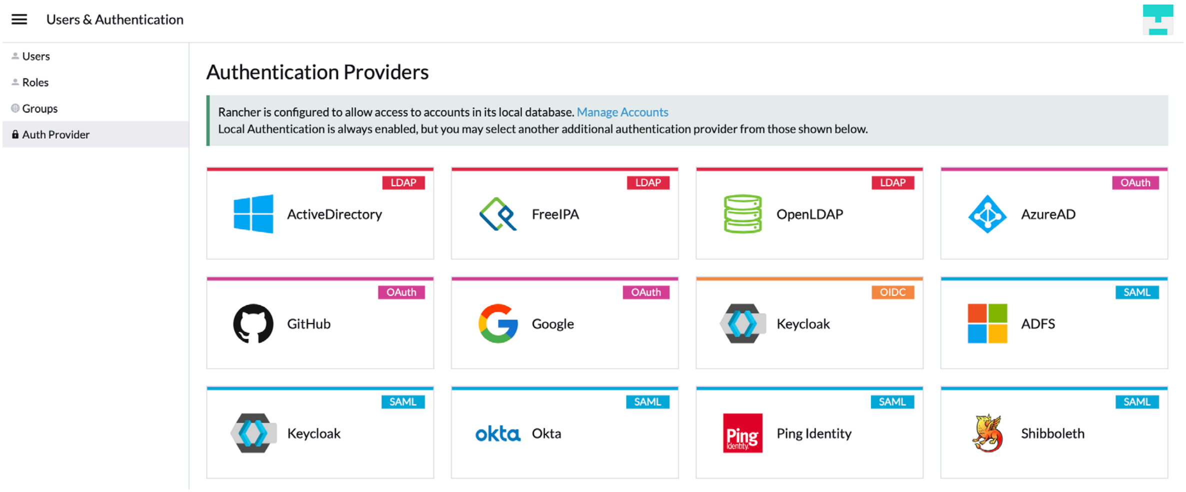
2: Rancher + Elemental EDGE stack

## 2.3 Authentication 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 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

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 (Lightweight Directory Access Protocol), SAML, GitHub and more. When connected in this way, Rancher enables you to extend your existing corporate authentication out to all the Kubernetes clusters under Rancher's umbrella, no matter where they are running.



Authentication Providers

Rancher is configured to allow access to accounts in its local database. Manage Accounts
Local Authentication is always enabled, but you may select another additional authentication provider from those shown below.

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 import it, that cluster is secure.

## 2.4 Shared Tools & Services

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

The app catalog experience in Rancher is based on Helm charts. Helm is a powerful templating mechanism for deploying applications on Kubernetes. But users still need to read 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 works with any CI/CD systems that integrate with Kubernetes. For example, Jenkins, Drone, and GitLab will continue to work with Rancher as they do with any other Kubernetes distribution. If the CI/CD system does not natively support Kubernetes, the Rancher CLI can be embedded to allow deployments to Rancher-managed Kubernetes clusters.

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

Rancher now also uses the SLE BCI-Micro image for audit logging sidecar containers instead of BusyBox. Deploying SLE BCI-Micro enforces security since it is based on SUSE Linux Enterprise Server, one of the most secure operating systems on the market. It also provides the necessary tooling to do what it needs, avoiding misconfigurations and minimizing security exposure.

## 2.5 Comprehensive Security

Security incidents are becoming increasingly widespread across container environments with misconfigurations now the most significant security exposure for Kubernetes environments.

Having the right tools to ensure that your clusters and the platform have the proper setup is crucial. Using admission controllers and regulated policies (as recommended by the Center for Internet Security) to avoid the improper deployment of containers with problematic configurations is now a high priority for organizations.

Rancher's security capabilities are not restricted to just tackling misconfigurations. Rancher includes multiple features to address vulnerabilities and help operators manage risk, including:

- ‘User ID Tracking’ has been added to audit logs to help users trace events. Rancher now includes the Identity Provider name in both Rancher and Kubernetes audit logs. This helps promote the self-service model of Rancher giving users clarity to identify different owners of clusters.
- ‘Image Scanning’ for CVEs (Common Vulnerabilities and Exposures) 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, Rancher has pre-determined actions to help identify, fix and/or mitigate issues.
- ‘SLE Base Container Image (SLE BCI)’ Rancher begun adopting SLE BCI as a base image for microservices and allows users access to a secure, open image that helps avoiding CVEs and misconfigurations.
- '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 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.
- ‘Integration with OPA/Gatekeeper’ Kubewarden and NeuVector enable admission controllers to manage clusters, and policies amongst other security features. Pod security policies were deprecated in Kubernetes v1.21 and removed in v1.25 the security capabilities provided by these two solutions should be crucial components to keep your clusters secure by blocking images affected by CVEs or dangerous configurations.
- NeuVector’ can be implemented with Rancher and extend Rancher’s fortified stance with its full lifecycle container security capabilities. It secures Runtime and adds compliance through its container segmentation and Layer 7 deep packet inspection which helps detect and block potential threats. Learn more about NeuVector here.
- With the addition of ‘Extensions’ in Rancher 2.7, new tool integrations are now available in the Rancher's UI. Operators can now access to more security-focused solutions, including the new CNCF sandbox project Kubewarden.
- For customers that require additional fortification across their container environment, Rancher Government Solutions is offering Carbide.
- Carbide delivers cutting-edge capabilities to enable software supply chain security and support federal security and compliance requirements.

- Rancher Government Carbide is an add-on support capability to the existing Rancher products suite.

# 3 High-level Architecture

## 3.1 Using 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.

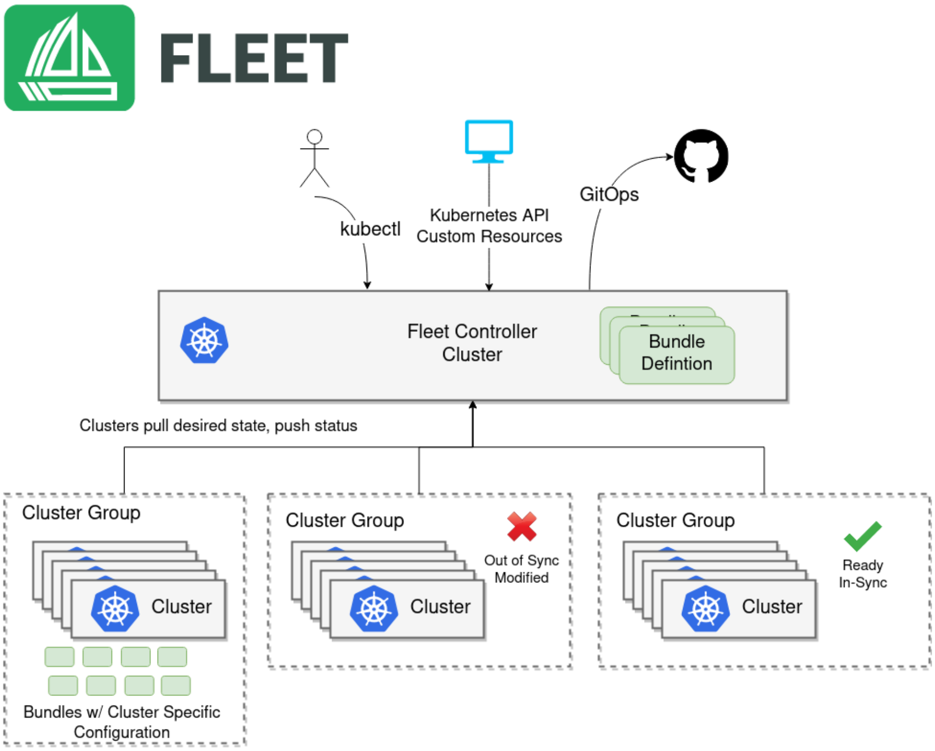


Figure 3: Fleet High-Level Architecture

## 3.2 Rancher Server

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

Figure 3 illustrates the high-level architecture of Rancher. It depicts a Rancher server installation that manages two Kubernetes clusters: one Kubernetes cluster created by RKE2 and another non-RKE2 Kubernetes cluster that could be EKS, AKS, GKE or any other Kubernetes cluster.

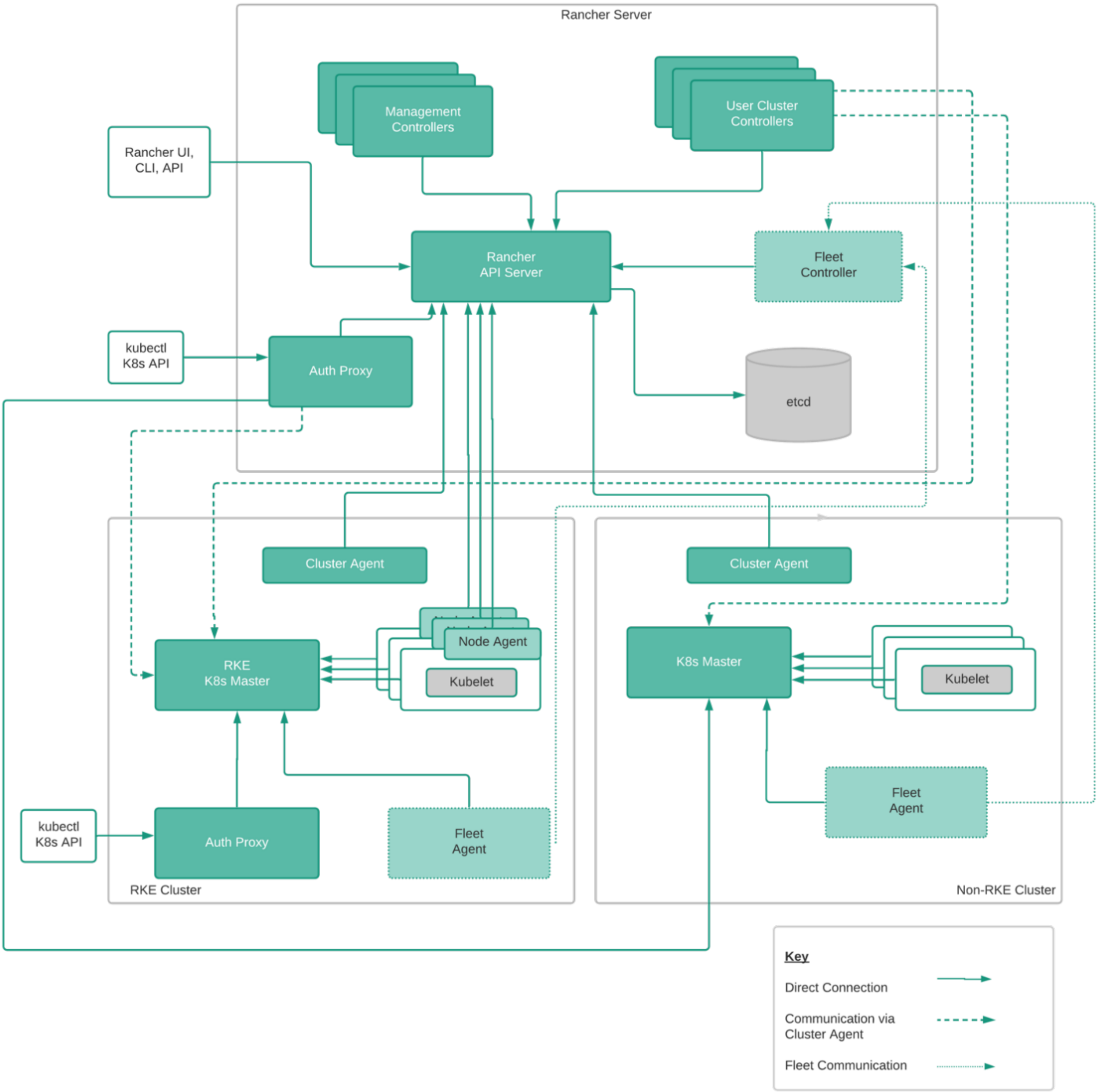


Figure 4: Rancher High-Level Architecture

# 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. 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 (custom resource definitions) 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, Okta and GitHub. The authentication proxy forwards all Kubernetes API calls to downstream clusters. 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, which provides an identity for processes that run in a pod.

The authentication cluster endpoint was introduced into RKE2 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.

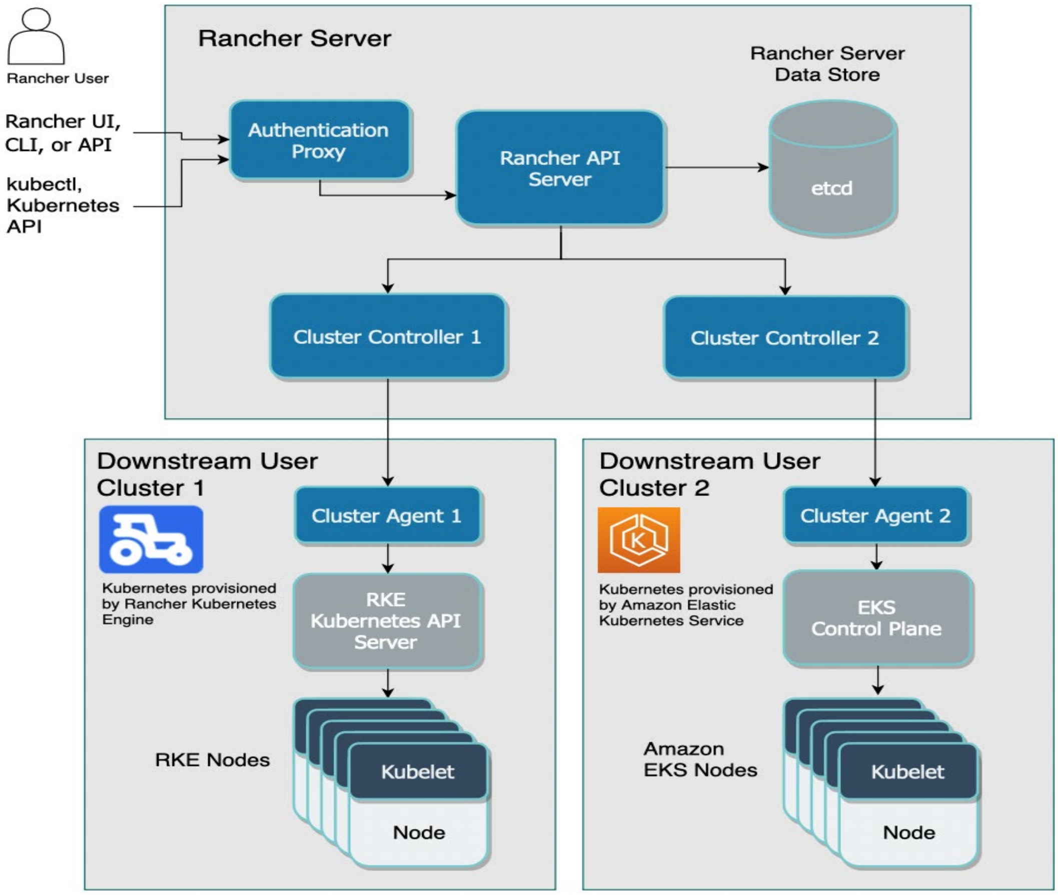


Figure 5: Rancher 2.7 Auth proxy communication diagram

### 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 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 Bundle Deployment.

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 2.7 by upgrading the Rancher server via a Helm upgrade. Rancher 2.7 will automatically upgrade Rancher agents in child clusters. Users will then have the option to upgrade the underlying Kubernetes versions of RKE2 and K3s clusters to take advantage of new functionality. This process also works in air-gapped environments, for more information review the docs.

# 7 High Availability

Users may use a dedicated RKE2/K3s cluster to run the Rancher server. The standard Rancher installation guide, for example, creates a cluster 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 for Rancher 2.7 can be found here.

# 8 About Rancher Government Solutions

Rancher Government Solutions is specifically designed to address the unique security and operational needs of the U.S. Government and military as it relates to application modernization, containers and Kubernetes.

Rancher is a complete open-source software stack for teams adopting containers. It addresses the operational and security challenges of managing multiple Kubernetes clusters at scale, while providing DevOps teams with integrated tools for running containerized workloads.

RGS supports all Rancher products with U.S. based American citizens with the highest security clearances who are currently supporting programs across the Department of Defense, Intelligence Community and civilian agencies.

**RGS Product Suite Includes:**

- **Rancher** - the world's most popular enterprise-grade Kubernetes management platform with over +40,000 active users and +120 million downloads.
- **RKE2** – is a fully conformant Kubernetes distribution focused on security and compliance.
- **Fleet** – an open source project built to help manage Kubernetes clusters at scale.
- **K3s** – a lightweight production-grade Kubernetes distribution built for embedded systems and the edge. In August 2020, K3s was donated to the CNCF as a sandbox project.
- **Longhorn** - a powerful cloud-native distributed storage platform for Kubernetes. In October 2019, Longhorn was donated to the CNCF as a sandbox project. In November 2021, Longhorn was promoted to an 'Incubating' project with the CNCF.

- **Harvester** - a modern open source hyperconverged infrastructure solution built on Kubernetes.
- **NeuVector –** an open source Zero Trust Kubernetes-native security platform.
- **Kubewarden** - a policy engine for Kubernetes. In June 2022, Kubewarden was donated to the CNCF as a sandbox project. Kubewarden enforces policy-as-code model allowing you to write policies in your favourite programming language.

RGS is committed to helping the US Government run Kubernetes securely everywhere. To learn more about Rancher Government Solutions, please contact us at:

info@rancherfederal.com
844-RGS-7779
ranchergovernment.com
Cage Code: 8GLZ3

Disclaimer: The author and Rancher Government Solutions assume no responsibility or liability for any errors or omissions, or for the results obtained from the use of this information. All information in this article is provided "as is." Rancher Government Solutions is available to help with any implementation roadblocks. To learn more about Rancher please visit: www.ranchergovernment.com