

# Introducing k3s - a Lightweight Kubernetes Distribution Built for the Edge



 RANCHER®

#k3s • February 25, 2019



**Shannon Williams**  
Co-Founder/VP Sales & Marketing  
Rancher Labs  
 @smw355



**Darren Shepherd**  
Co-Founder/Chief Architect  
Rancher Labs  
 @ibuildthecloud

# Agenda

1. Increasing demand for an edge-optimized Kubernetes
2. Introducing k3s – a micro-distribution of Kubernetes
3. How we cut down the size and runtime-foot print of Kubernetes
4. Adding SQLite as an optional data store
5. Demo
6. Managing clusters on the edge

# In the last year we've seen a major increase in the demand for Kubernetes outside the datacenter

## Bare Metal K8s Clustering at Chick-fil-A Scale



Chick-fil-A Tech Blog [Follow](#)

Jun 25, 2018 · 6 min read

by [Brian Chambers](#), [Caleb Hurd](#), and Alex Crane



# kubernetes

<https://www.youtube.com/watch?v=8edDcy3oeUo>

At full scale Chick-fil-A will be running Kubernetes at the Edge in each of our 2000 restaurants. That means roughly 6000 devices at the Edge running Kubernetes.

One of the biggest challenges associated with this is **bare metal clustering** on-the-fly, in-restaurant.

While most Kubernetes deployments are in the cloud or benefit from skilled technicians that are physically located near their deployments (or at least equipped with remote access), our deployments are completed by installers who focus only on initial hardware installations. They never connect to the compute devices directly—rather they connect ethernet and power cords, and then look at an app to check the status of the cluster as it self-bootstraps. Replacements are completed by restaurant Owner/Operators or their teams, which are sometimes less technical.

On top of that, our Edge deployments are not exactly in a “datacenter environment”.



# In the last year we've seen a major increase in the demand for Kubernetes outside the datacenter

The screenshot shows a blog post from the Target tech blog. The header includes the Target logo and navigation links for events, open source, dojo, and careers. The main title of the post is "Meet Target's Stores Deployment Interface that Realizes Distributed Edge Computing at Retail Scale". It was posted by Dan Woods on June 20, 2018. The post features four tags: edge, iot, kubernetes, and docker. The first paragraph begins with "At the beginning of 2017..." and describes the engineering team's journey to facilitate rapid software delivery to Target stores.

tech blog

events open source dojo careers

## Meet Target's Stores Deployment Interface that Realizes Distributed Edge Computing at Retail Scale

Posted by Dan Woods  
Jun 20, 2018

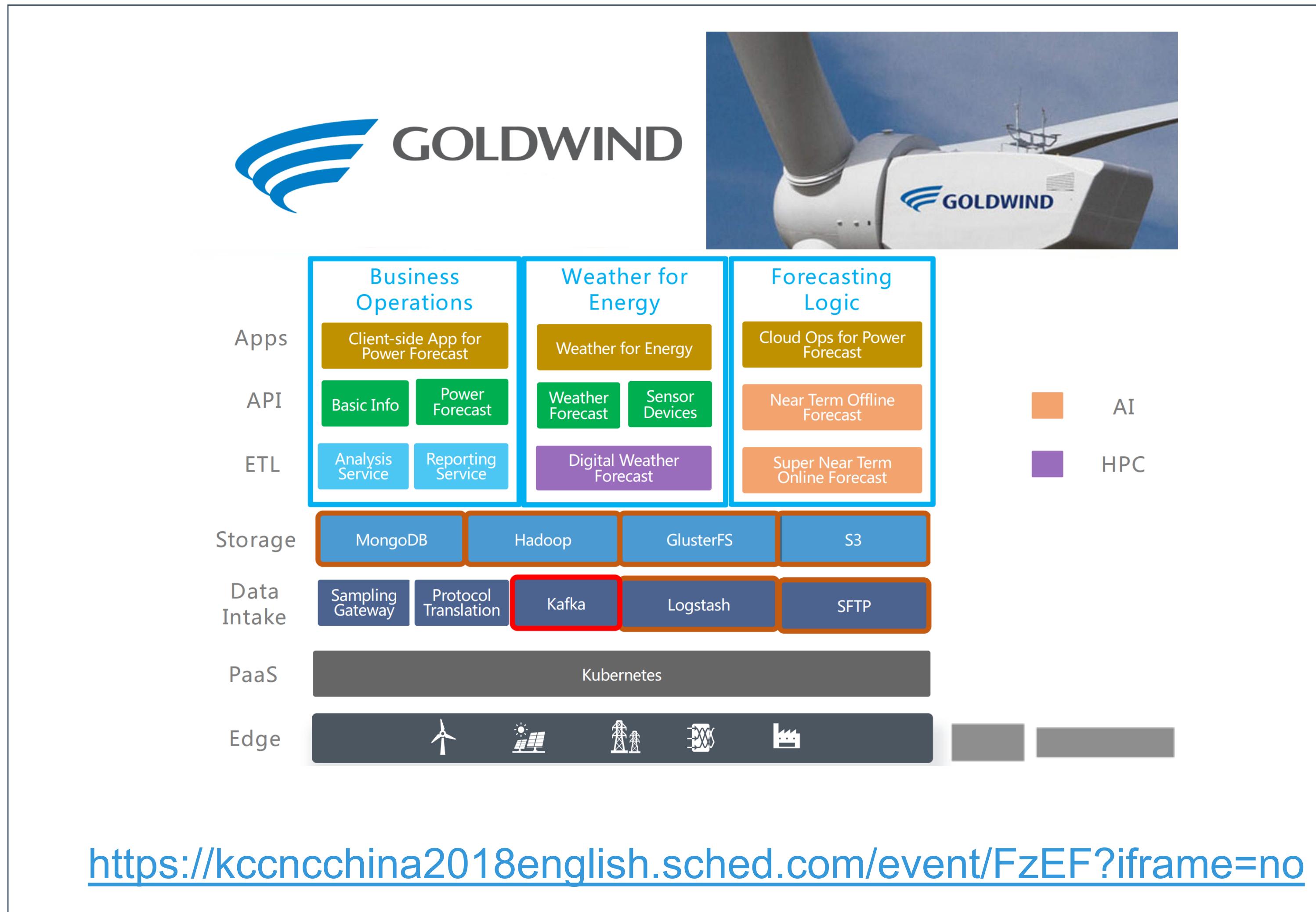
edge iot kubernetes docker

**At the beginning of 2017...**

... our engineering team embarked on a journey to facilitate rapid software delivery to Target stores to better enable innovation and more quickly respond to ever-evolving business needs. The destination seemed clear and simple, but we quickly discovered the path to get there was not – not well-defined for creating a distributed software runtime platform to scale to more than 1,800 edge locations. Along the way, however, we forged a path that will forever change how we build, deploy and manage software at Target.

<https://tech.target.com/infrastructure/2018/06/20/enter-unimatrix.html>

# In the last year we've seen a major increase in the demand for Kubernetes outside the datacenter



# Containers and Kubernetes are an excellent choice for deploying complex software to the edge

- Containers are awesome
- Consistent across a wide variety of infrastructure
- Capable of standalone or clustered operations
- Easy to upgrade and/or replace containers
- Support for different infrastructure configs (storage, cpu, etc)
- Strong ecosystem (monitoring, logging, CI, management, etc)

# Our journey to building an edge-optimized k8s distro started with Rancher



- Open source
- Multi-cluster management
- Deploy k8s anywhere
- Manage any k8s cluster

A screenshot of the Rancher web interface. At the top, there's a navigation bar with links for Global, Clusters (which is the active tab), Node Drivers, Catalogs, Users, Settings, and Security. Below the navigation is a section titled "Add Cluster".

The "In a hosted Kubernetes provider" section contains three options: Google Container Engine (selected, indicated by a checked checkbox icon), Amazon EKS, and Azure Kubernetes Service. To the right of this section is a button labeled "Import existing cluster" with an "IMPORT" icon.

The "From nodes in an infrastructure provider" section contains five options: Amazon EC2, Microsoft Azure, DigitalOcean, openstack, and others like Open Telekom Cloud, Otcdev, and vSphere.

The "From my own existing nodes" section contains a single option: CUSTOM, represented by a server icon with a gear.

At the bottom of the form, there's a "Cluster Name" field with placeholder text "e.g. sandbox", an "Add a Description" link, and a "Member Roles" section with a note: "Control who has access to the cluster and what permission they have to change it."

# In late 2017, we pulled out RKE our Kubernetes installer



- Open-source k8s installer and distribution
- Externally define cluster config
- Launch services and add-ons as part of deployment

```
./rke --version
rke version v0.0.6-dev

./rke --help
NAME:
  rke - Rancher Kubernetes Engine, Running kubernetes cluster in the cloud

USAGE:
  rke [global options] command [comm
VERSION:
  v0.0.6-dev

AUTHOR(S):
  Rancher Labs, Inc.

COMMANDS:
  up           Bring the cluster up
  remove       Teardown the cluster
  version      Show cluster Kubernetes version
  config, config  Setup cluster configuration
  help, h       Shows a list of commands and their descriptions

GLOBAL OPTIONS:
  --debug, -d   Debug logging
  --help, -h    show help
  --version, -v print the version

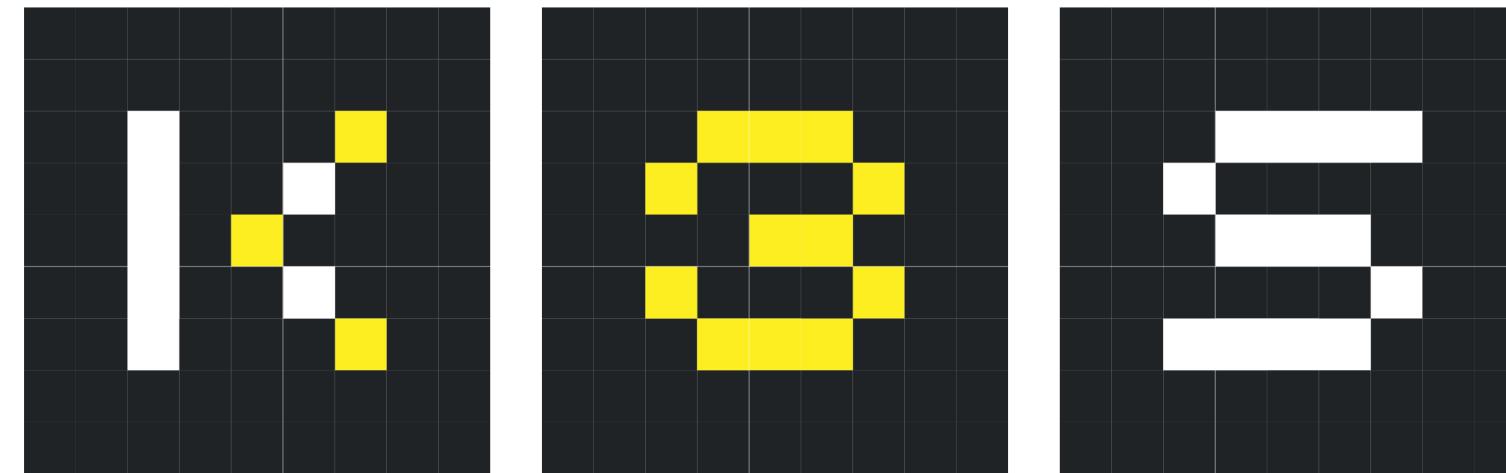
  ---[redacted]
  nodes:
    - address: 192.168.1.5
      user: ubuntu
      role: [controlplane]
    - address: 192.168.1.6
      user: ubuntu
      role: [worker]
    - address: 192.168.1.7
      user: ubuntu
      role: [etcd]

  services:
    etcd:
      image: quay.io/coreos/etcd:latest
    kube-api:
      image: rancher/k8s:v1.8.3-rancher2
    kube-controller:
      image: rancher/k8s:v1.8.3-rancher2
    scheduler:
      image: rancher/k8s:v1.8.3-rancher2
    kubelet:
      image: rancher/k8s:v1.8.3-rancher2
    kubeproxy:
      image: rancher/k8s:v1.8.3-rancher2
```

# However, running Kubernetes on the edge left us with some challenges

- Most Kubernetes distributions don't support ARM
- Kubernetes could easily consume up to 4gb of RAM
- Kubernetes wasn't built for embedded or offline management

# Introducing k3s



- Lightweight certified Kubernetes distro
- Built for production operations
- 40MB binary, 512MB memory consumption
- Single process w/ integrated Kubernetes master, Kubelet, and containerd
- SQLite in addition to etcd
- Simultaneously released for x86\_64, ARM64, and ARMv7
- Open source project, not yet a Rancher product

To build k3s we removed unnecessary code and made a few enhancements

# Removes

Legacy and non-default features

Alpha features

In-tree cloud providers

In-tree storage drivers

Docker (optional)

# Adds

Simplified installation

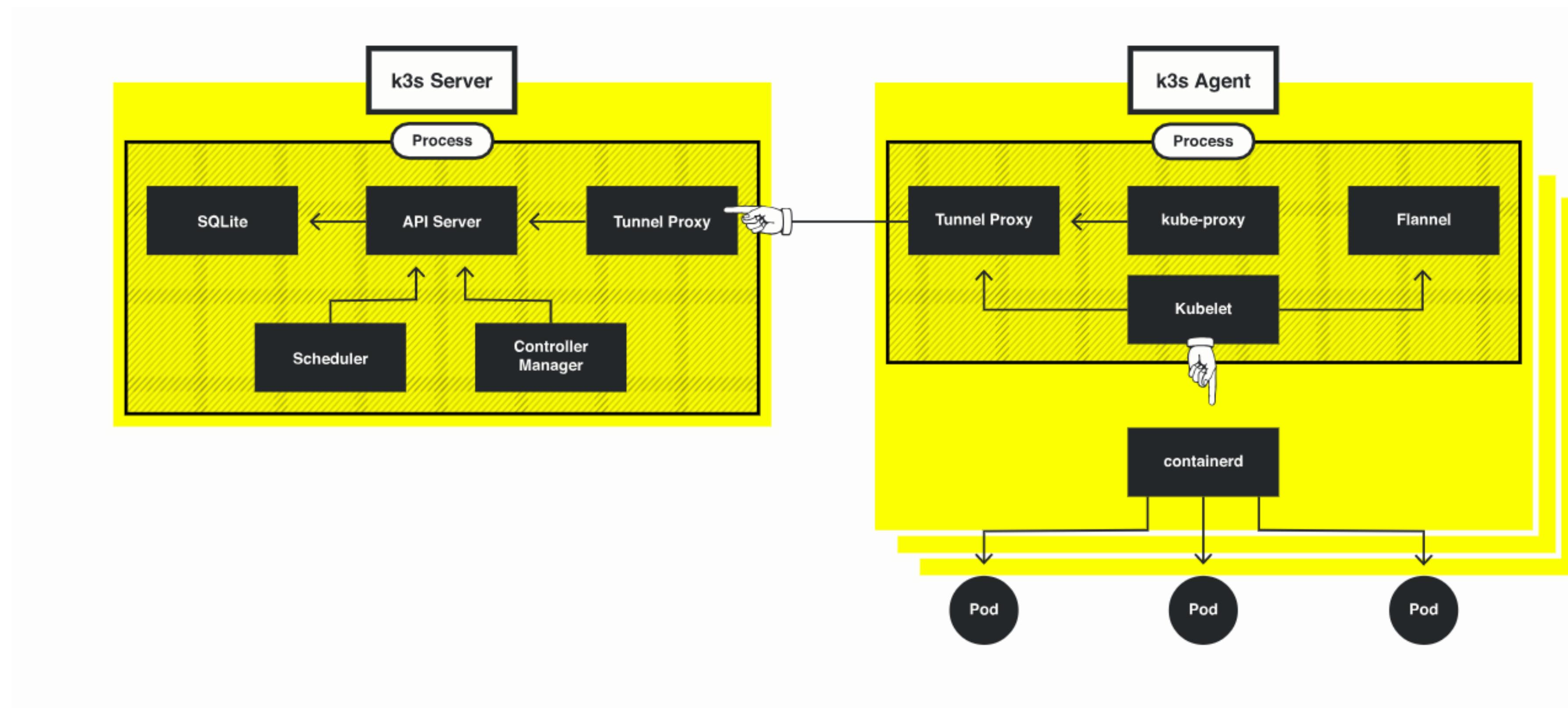
SQLite3 support in addition to etcd

TLS management

Automatic Manifest and Helm Chart management

containerd, CoreDNS, Flannel

# How k3s works



# K3s use cases

1. Edge computing
2. Bundled into applications
3. Embedded appliances
4. CI environments

# Get started with k3s in two easy steps

## Quick Start

1. Download k3s - [latest release](#), x86\_64, ARMv7, and ARM64 are supported
2. Run server

```
sudo k3s server &
# Kubeconfig is written to /etc/rancher/k3s/k3s.yaml
sudo k3s kubectl get node

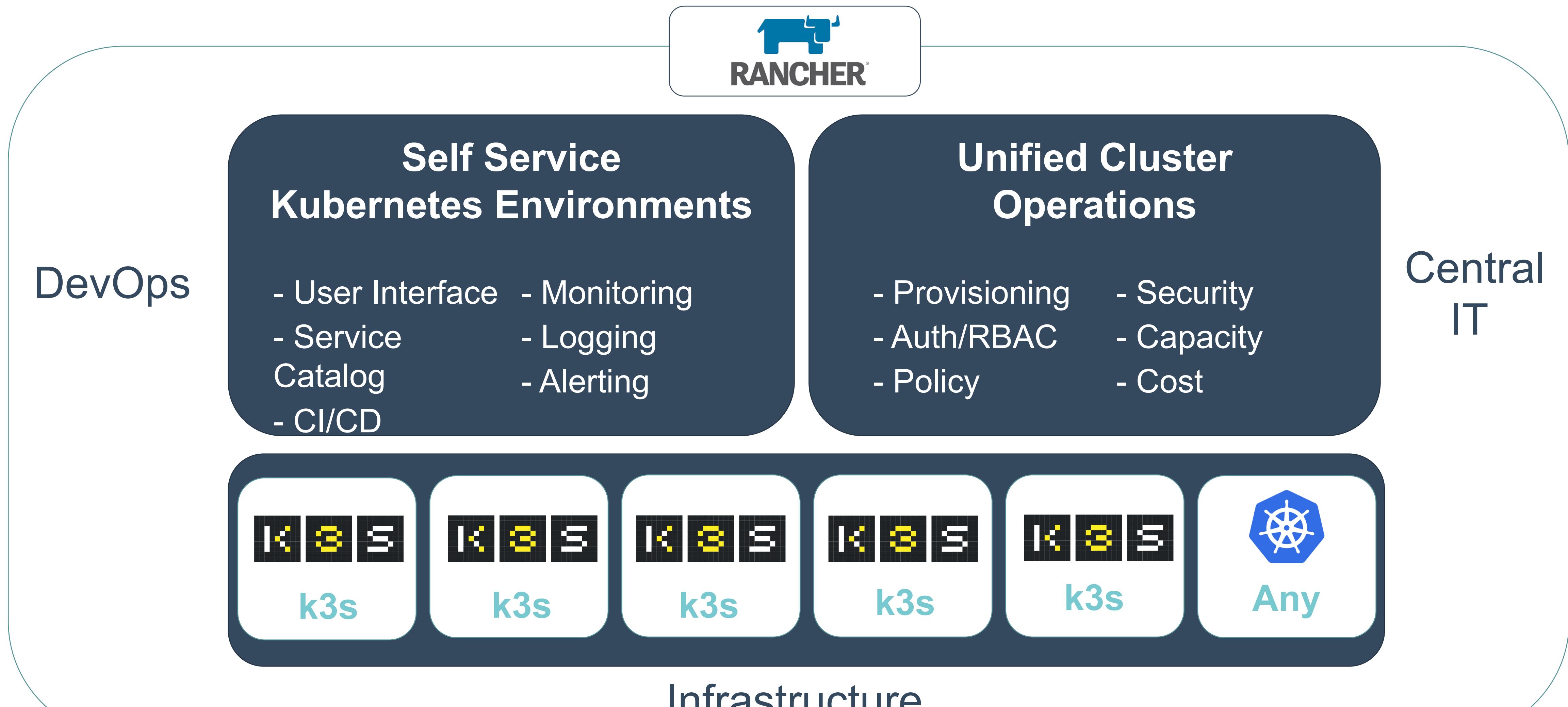
# On a different node run the below. NODE_TOKEN comes from /var/lib/rancher/k3s/server/node-token
# on your server
sudo k3s agent --server https://myserver:6443 --token ${NODE_TOKEN}
```

# Demo

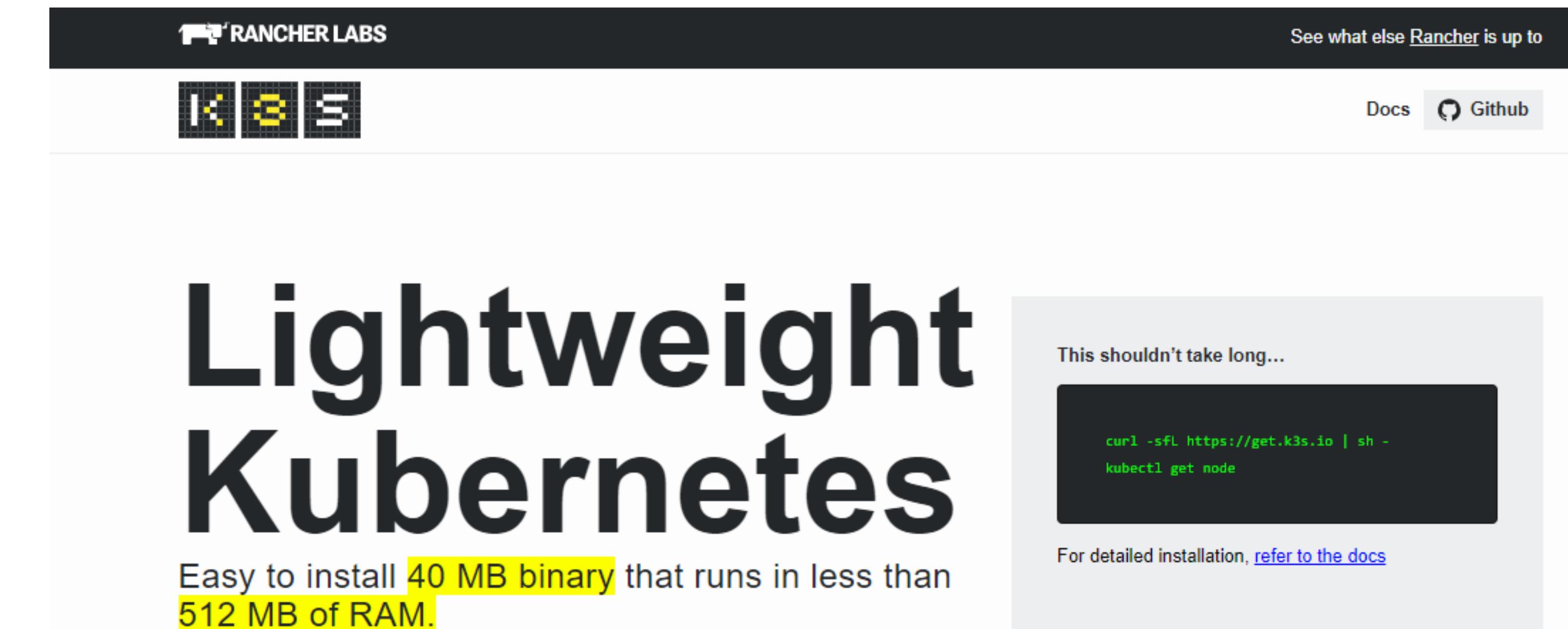
# Fleet Management for Edge Clusters

1. Multi-cluster operations (monitoring, logging, etc.)
2. Centralized access control and user management
3. Centralized security policies
4. Multi-cluster application templates
5. CI environments

# Using Rancher for unified cluster operations



<http://k3s.io>



The screenshot shows the official K3S website. At the top, there's a dark header with the Rancher Labs logo and a link to "See what else Rancher is up to". Below the header is the K3S logo, which consists of three squares containing the letters K, 3, and S. To the right of the logo are links for "Docs" and "Github". The main title "Lightweight Kubernetes" is prominently displayed in large, bold, black font. Below it, a subtitle reads "Easy to install 40 MB binary that runs in less than 512 MB of RAM." To the right, a light gray sidebar contains the text "This shouldn't take long..." above a terminal-style code block showing the command: "curl -sSL https://get.k3s.io | sh - & kubectl get node". At the bottom of the sidebar, it says "For detailed installation, [refer to the docs](#)".

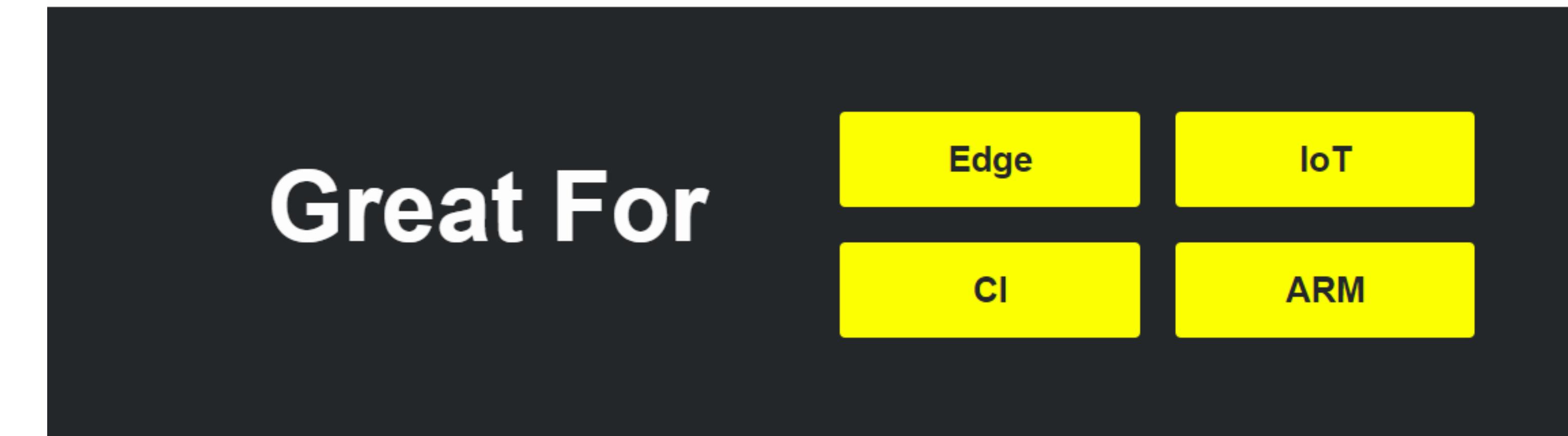
# Lightweight Kubernetes

Easy to install 40 MB binary that runs in less than 512 MB of RAM.

This shouldn't take long...

```
curl -sSL https://get.k3s.io | sh - & kubectl get node
```

For detailed installation, [refer to the docs](#)



The screenshot shows a dark rectangular area with the text "Great For" in white. To the right of this text are four yellow rectangular boxes arranged in a 2x2 grid, each containing a word: "Edge", "IoT", "CI", and "ARM".

## Great For

Edge	IoT
CI	ARM

<https://github.com/rancher/k3s>

The screenshot shows the GitHub repository page for the project `rancher / k3s`. The repository is described as "Lightweight Kubernetes. 5 less than k8s. <https://k3s.io>". It features 57 commits, 3 branches, 42 releases, and 3 contributors. The latest commit was made 14 hours ago. The repository uses the Apache-2.0 license. The code tab is selected, showing recent commits from `erikwilson` and `ibuildthecloud`, such as fixing asset lookup of HOME directory and setting up ARM CI. Other commits include fixes for ingress, package management, and agent startup.

Code Issues Pull requests Projects Wiki Insights

kubernetes k8s

57 commits 3 branches 42 releases 3 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

erikwilson and ibuildthecloud Fix asset lookup of HOME directory ... Latest commit 91251aa 14 hours ago

cmd Fix asset lookup of HOME directory 9 hours ago

manifests Fix ingress 12 days ago

package Get ARM CI working 21 days ago

pkg Set /proc/sys/net/ipv4/ip\_forward on agent start 2 days ago

scripts Fix version printing on startup 18 days ago

types Update generated code 18 days ago



# Thank you

@Rancher\_Labs · #RancherMeetup