



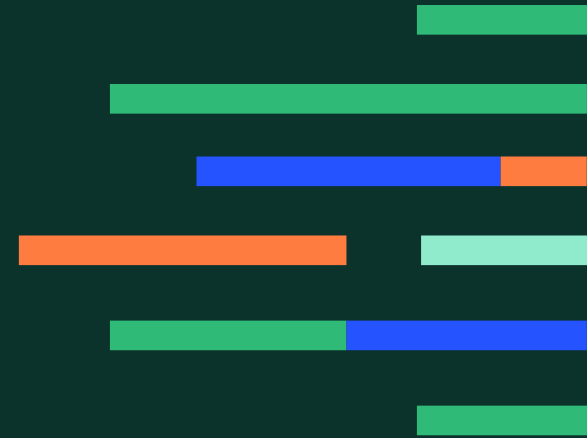
# Gestionar clusters de Kubernetes con Rancher y CAPI

Andrés Valero  
esLibre 2024

# Agenda

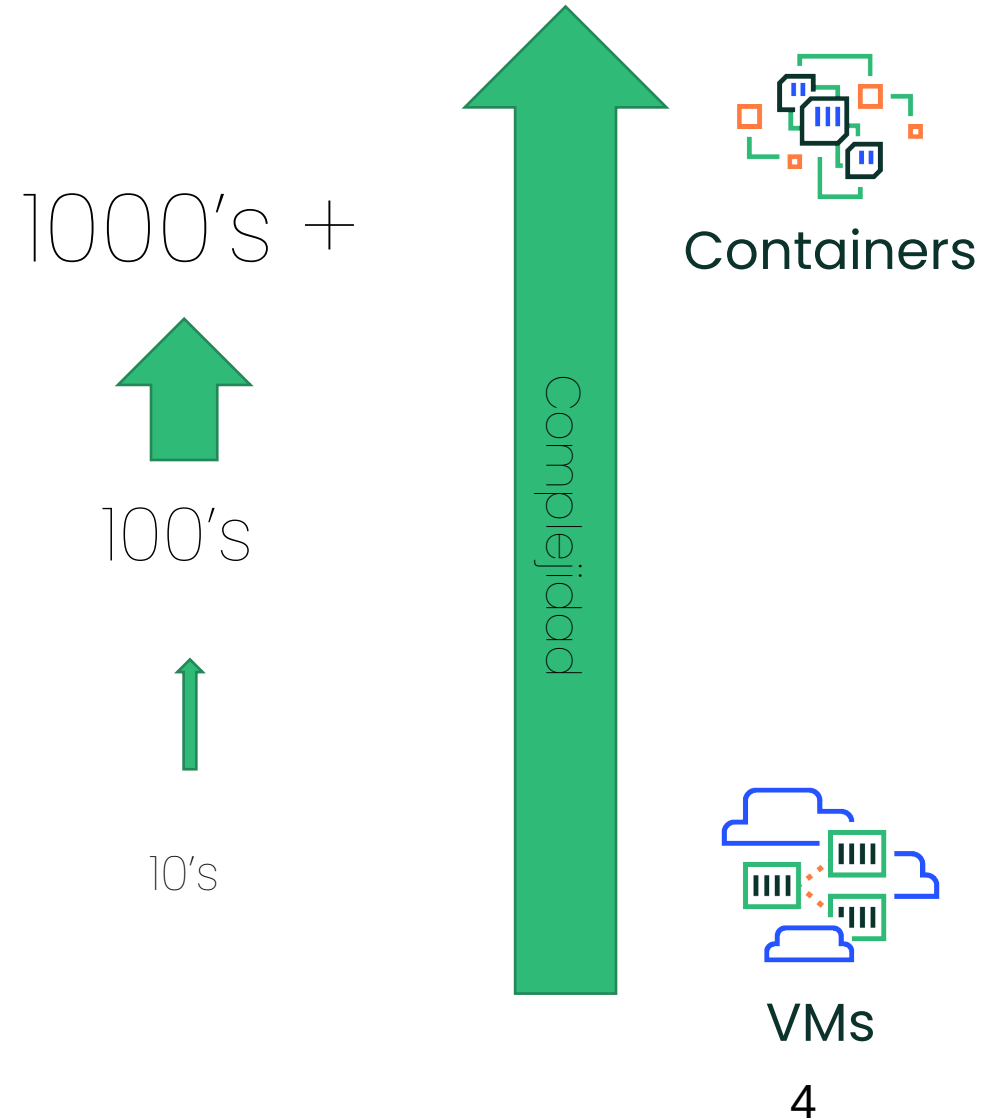
1. Gestionar Kubernetes es complejo
2. Introducción a Rancher
3. ¿Qué es CAPI?
4. ¿Por qué Rancher y CAPI?
5. Demo

Gestionar  
Kubernetes es  
complejo



# Contenedores & Kubernetes 101

- ✓ Gestionar imagenes
- ✓ Desplegar Contenedores (pods)
- ✓ Integrar servicios
- ✓ Gestionar recursos y hardware
- ✓ Balancear recursos y cargas
- ✓ Gestionar y orquestar servicios
- ✓ Infraestructura compleja
- ✓ Trabajar declarativamente



# Gestionar Kubernetes NO es fácil

**Complejidad:** Kubernetes tiene una curva de aprendizaje pronunciada debido a su complejidad, especialmente para aquellos nuevos en la orquestación de contenedores.

**Gestión de recursos:** Optimizar la asignación y utilización de recursos en los clústeres puede ser un desafío, lo que puede llevar a una subutilización o sobredimensionamiento.

**Escalabilidad:** Escalar clústeres de Kubernetes de manera eficiente manteniendo el rendimiento y la estabilidad es complicado.

**Redes:** Configurar y gestionar las redes dentro de los clústeres de Kubernetes, incluido el "service mesh" y los "load balancers", puede ser complejo, especialmente en entornos híbridos o multi-cloud.



# Gestionar Kubernetes NO es fácil

**Seguridad:** Asegurar la seguridad de los clústeres de Kubernetes, incluida la autenticación, autorización y políticas de red, es crucial pero complejo.

**Monitorización y logs:** La implementación de soluciones efectivas de monitorización y login para obtener información sobre la salud del clúster, el rendimiento y el uso de recursos es esencial.

**Actualizaciones y cambios de versión:** Gestionar actualizaciones y cambios en Kubernetes mismo, así como en las aplicaciones que se ejecutan en él, minimizando la interrupción de las operaciones puede ser un desafío.

**Diversidad de tecnologías:** Kubernetes significa gestionar diferentes tecnologías, distribuciones de Kubernetes, en diferentes plataformas en diferentes SOs. Esto añade una capa extra de complejidad.



# Intro a Rancher



# ¿Qué es Rancher?

Rancher es una plataforma de gestión de Kubernetes open source de grado empresarial. Provee un punto unificado de gestión para diferentes clusters corriendo en diferentes plataformas e infraestructuras.

Rancher ofrece una plataforma con todas las herramientas necesarias para gestionar Kubernetes a escala de una forma simplificada

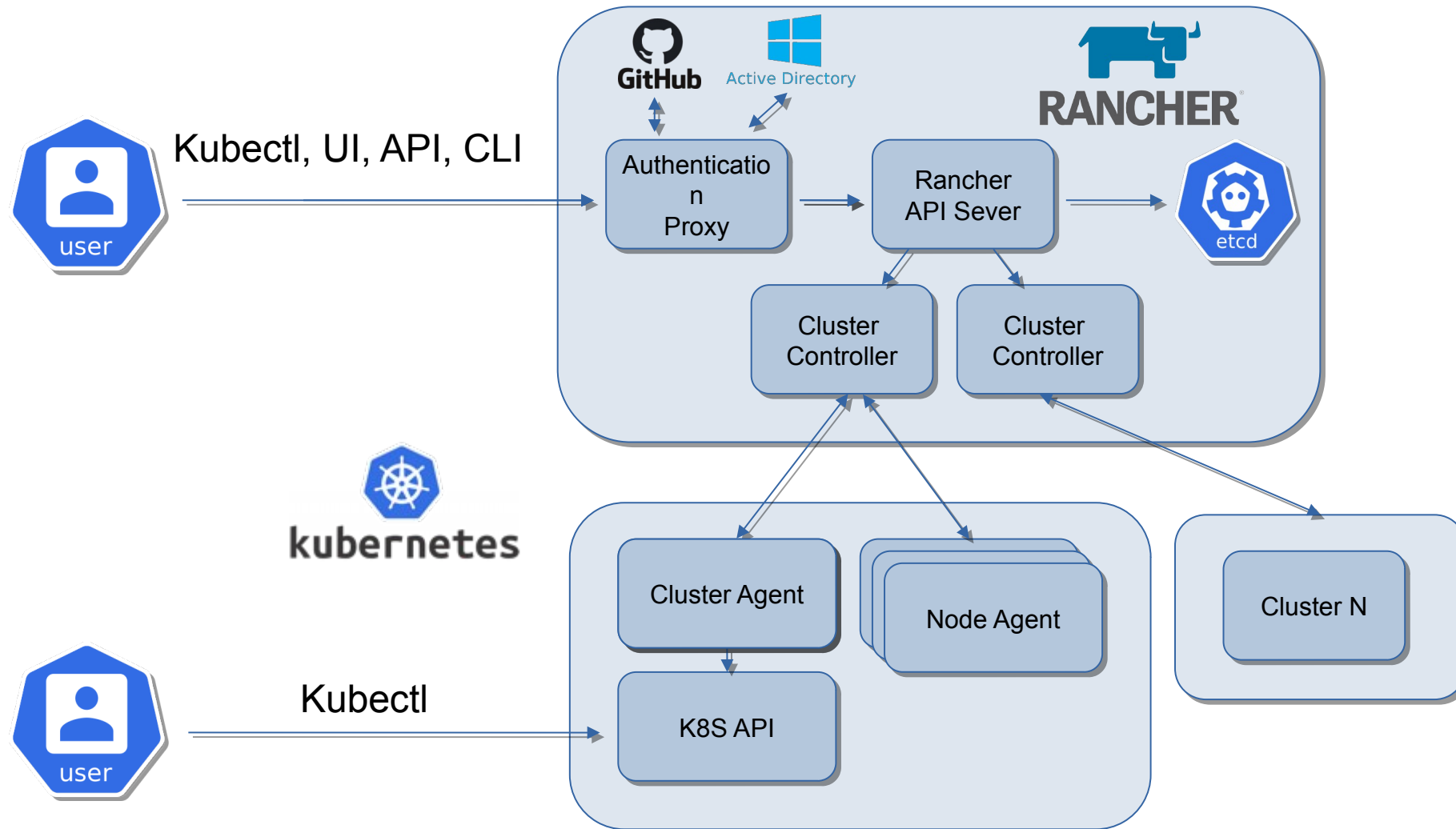
Soporte para diferentes soluciones de infraestructura, distribuciones de Kubernetes y sistemas operativos, ofreciendo la flexibilidad necesaria en el mundo cloud native.

Gestión de roles, permisos y cuotas simplificado.

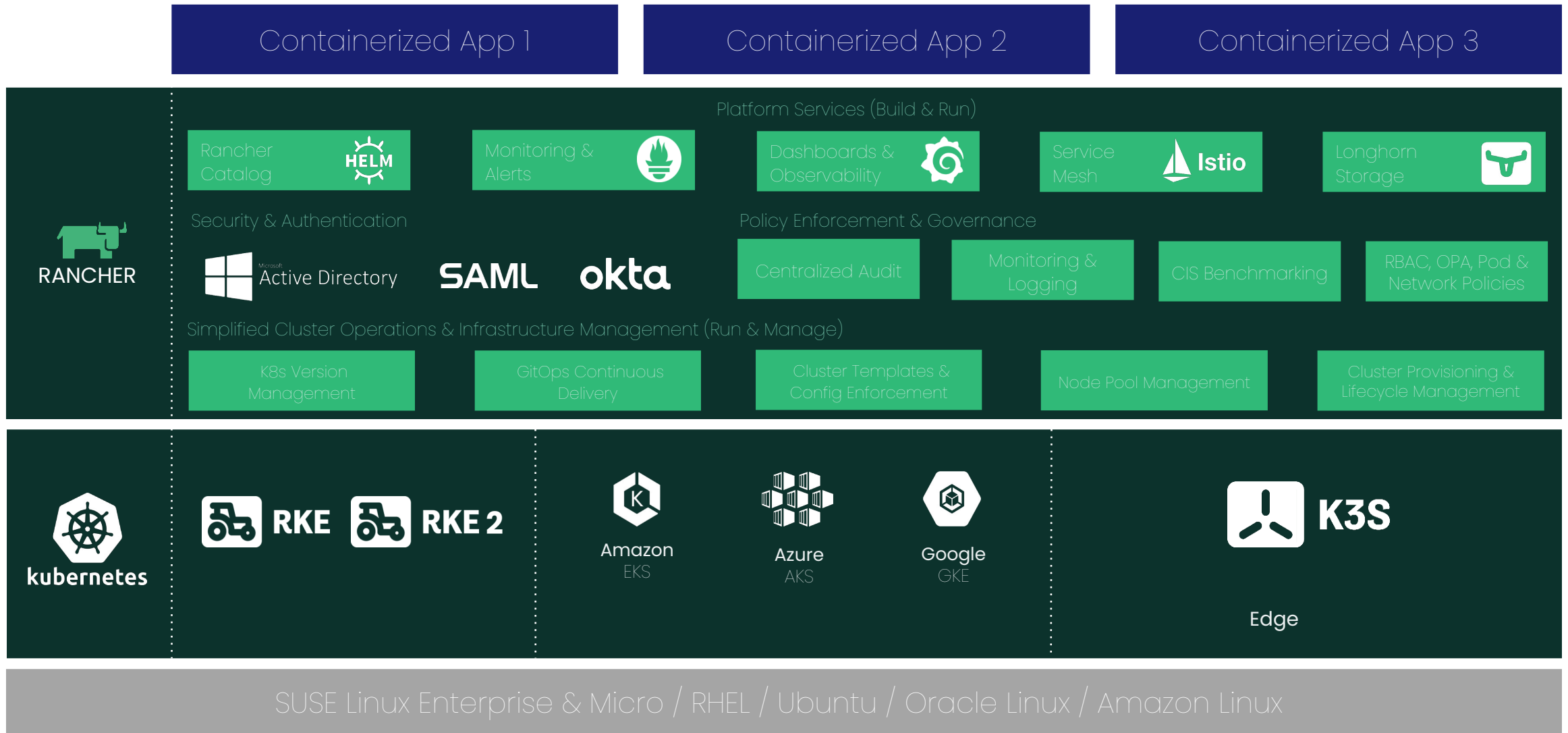




# Arquitectura de Rancher a alto nivel



# Principales características de Rancher

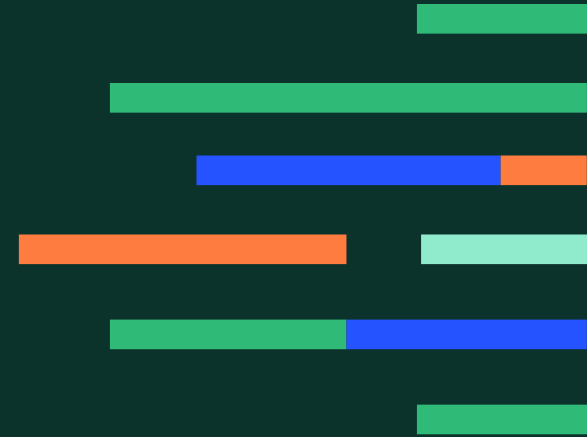
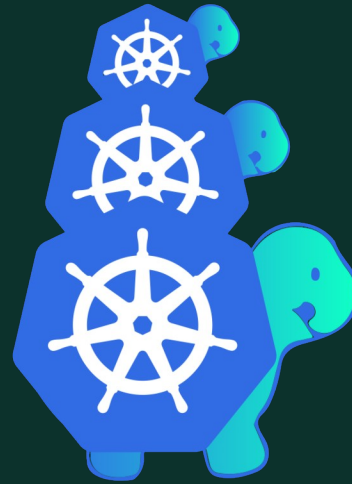


# Principales beneficios de Rancher

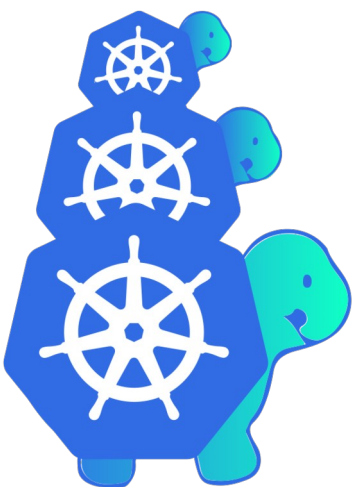
1. Interfaz intuitiva, fácil de usar y fácil de extender usando "Rancher Extensions".
2. Soporte de gestión de clusters y aplicaciones a escala, incluyendo casos de uso como Edge Computing usando metodologías GitOps. Integración y soporte de CI/CD y prácticas DevOps.
3. Automatización de despliegues de clusters vía interfaz o vía GitOps usando templates.
4. Soporte para los principales proveedores de Servicios de cloud computing, infraestructura y bare metal.
5. Marketplace de aplicaciones basado en Helm.
6. Elemental para la gestión de sistemas operativos y onboarding.
7. Gran comunidad soportando y utilizando el proyecto.



# ¿Qué es CAPI?



# Cluster API (CAPI)

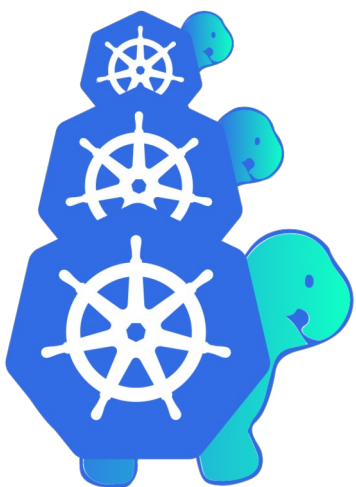


Cluster API es un subproyecto de Kubernetes centrado en proporcionar APIs declarativas y herramientas para simplificar el aprovisionamiento, la actualización y la operación de múltiples clusters de Kubernetes.

El proyecto Cluster API utiliza APIs y patrones al estilo de Kubernetes para automatizar la gestión del ciclo de vida de los clusters. La infraestructura de soporte, así como la configuración del cluster de Kubernetes, se definen de la misma manera en que los desarrolladores de aplicaciones operan al desplegar y gestionar sus cargas de trabajo.



# Beneficios de CAPI



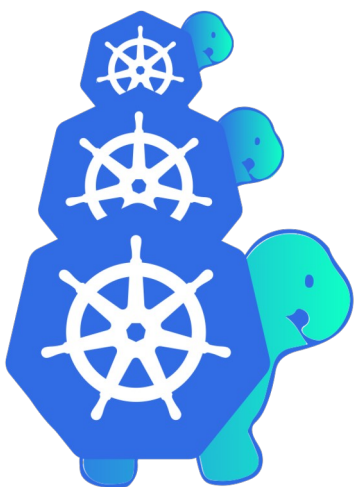
**Automatización de la Gestión del Ciclo de Vida de los Clusters:** Cluster API automatiza las tareas repetitivas y propensas a errores asociadas con la gestión del ciclo de vida de los clusters, como el aprovisionamiento, la actualización y la eliminación de clusters de Kubernetes.

**APIs Declarativas:** Al utilizar APIs declarativas, Cluster API permite a los usuarios definir el estado deseado de los clusters. Kubernetes se encarga de reconciliar el estado actual con el estado deseado, facilitando una gestión más sencilla y predecible.

**Consistencia y Repetibilidad:** Cluster API permite definir la infraestructura y la configuración del cluster de manera declarativa, lo que asegura que los despliegues de clusters sean consistentes y repetibles.



# Beneficios de CAPI



**Soporte Multi-cloud y On-premise:** Cluster API es compatible con diferentes proveedores de infraestructura, incluidos AWS, Azure, Google Cloud y entornos on-premise.

**Escalabilidad:** Cluster API permite escalar tanto horizontal como verticalmente los clusters de Kubernetes de manera eficiente. Esto asegura que los recursos se pueden ajustar según la demanda, optimizando el uso de la infraestructura.

**Extensibilidad:** Cluster API está diseñado para ser extensible, lo que permite a los usuarios y proveedores de infraestructura crear controladores personalizados que se integran con sus sistemas y herramientas específicas.

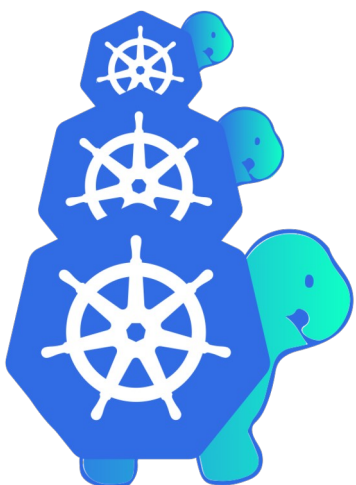
**Mejora Continua y Comunidad Activa:** Cluster API es un proyecto de código abierto con una comunidad activa que contribuye constantemente a su mejora.



# Cluster API (CAPI)

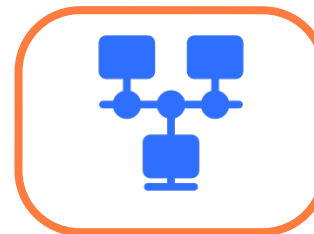
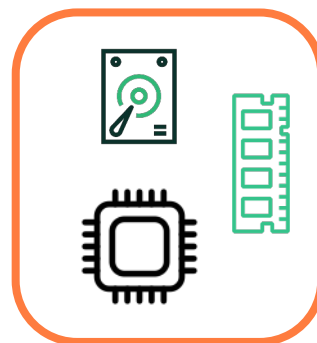
## Core CAPI Controller

Definición estándar de Cluster



## Infrastructure Provider

Crea VMs  
Crea Redes  
Cliente API para Infraestructura



## Bootstrap Provider

Instala y configura  
Kubernetes en los nodos





# Conceptos

**Bootstrap**: componente responsable de convertir un servidor en un nodo de Kubernetes y de:

- Generar los certificados del clúster, si no se especifica de otra manera.
- Inicializar el plano de control y controlar la creación de otros nodos hasta que esté completo.
- Unir nodos del plano de control y nodos de trabajo al clúster.

**Control Plane**: conjunto de componentes que sirven la API de Kubernetes y reconcilian continuamente el estado deseado utilizando bucles de control. Usa kubeadm por defecto, sin embargo hay otros proveedores como K3s, MicroK8s o K0s.

**Providers**: componente responsable del aprovisionamiento de los recursos de infraestructura/computacionales requeridos por el clúster o por las máquinas (por ejemplo, máquinas virtuales, redes, etc.). Por ejemplo, los proveedores de infraestructura en la nube incluyen AWS, Azure y Google, y los proveedores de infraestructura en hardware físico incluyen VMware, MAAS y metal3.io. En total CAPI soporta unos 30 proveedores de infra diferentes.



# Conceptos

**Clusters:** Definición básica del cluster que se va a desplegar.

**Machine Deployments:** Provee definiciones para provisionar el cluster cómo el proveedor de bootstrap y de infraestructura que se van a utilizar para desplegar el cluster.

**Machine Templates:** Define el tipo de máquina que se van a provisionar como nodo de Kubernetes en el proveedor de infraestructura.

**Machine Sets:** Información de las máquinas a desplegar como nodos de Kubernetes

**Cluster Classes:** Define y enlaza todas las piezas que intervienen en el despliegue del cluster así como configuraciones para el cluster.

<https://github.com/k3s-io/cluster-api-k3s/blob/main/samples/aws/k3s-template.yaml>



# ¿Por qué Rancher & CAPI?

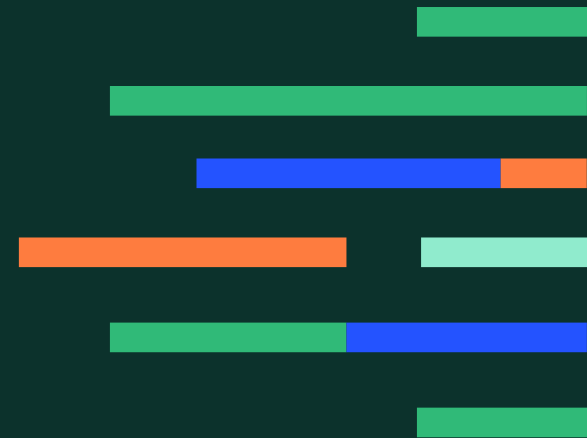


## Rancher provisiona clusters utilizando CAPI o Docker Machine dependiendo de la distribución de Kubernetes. ¿Por qué cambiar?

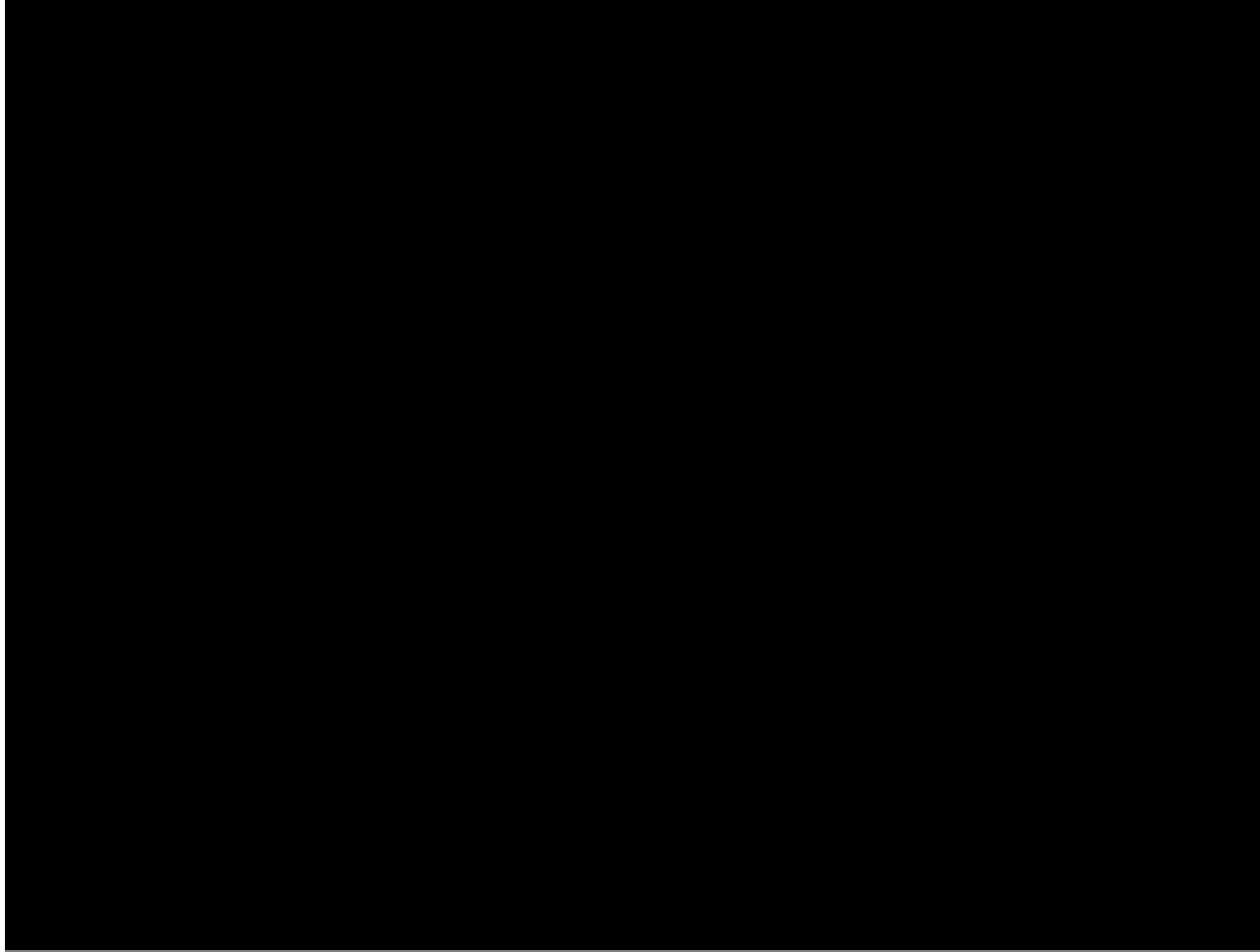
1. CAPI con clusterctl está siendo adoptado mayoritariamente para desplegar Kubernetes.
2. Adoptando CAPI a través de una extensión los usuarios tienen una experiencia más “pura” y facilita la adopción de GitOps para la gestión de clusters usando “The CAPI way” no “The Rancher way”.
3. CAPI no ofrece una interfaz gráfica para los usuarios, integrándose con Rancher los usuarios pueden usar la interfaz de Rancher para gestionar los clusters de Kubernetes.
4. Los clusters creados con CAPI a través de Rancher se auto-importan facilitando la gestión, así como el despliegue de aplicaciones.
5. Posibilidad de desplegar más distribuciones de Kubernetes desde Rancher.



# Demo



# Demo





# ¡Gracias!