

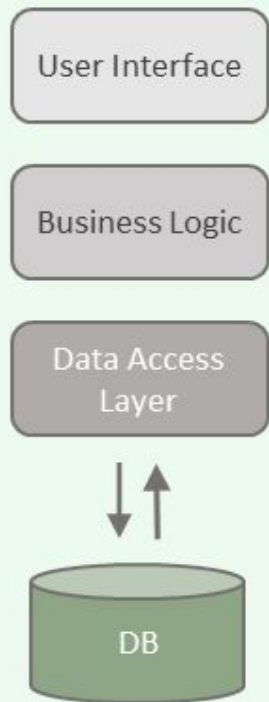
# Kubernetes vs DockerSwarm

Sebastián Guerra

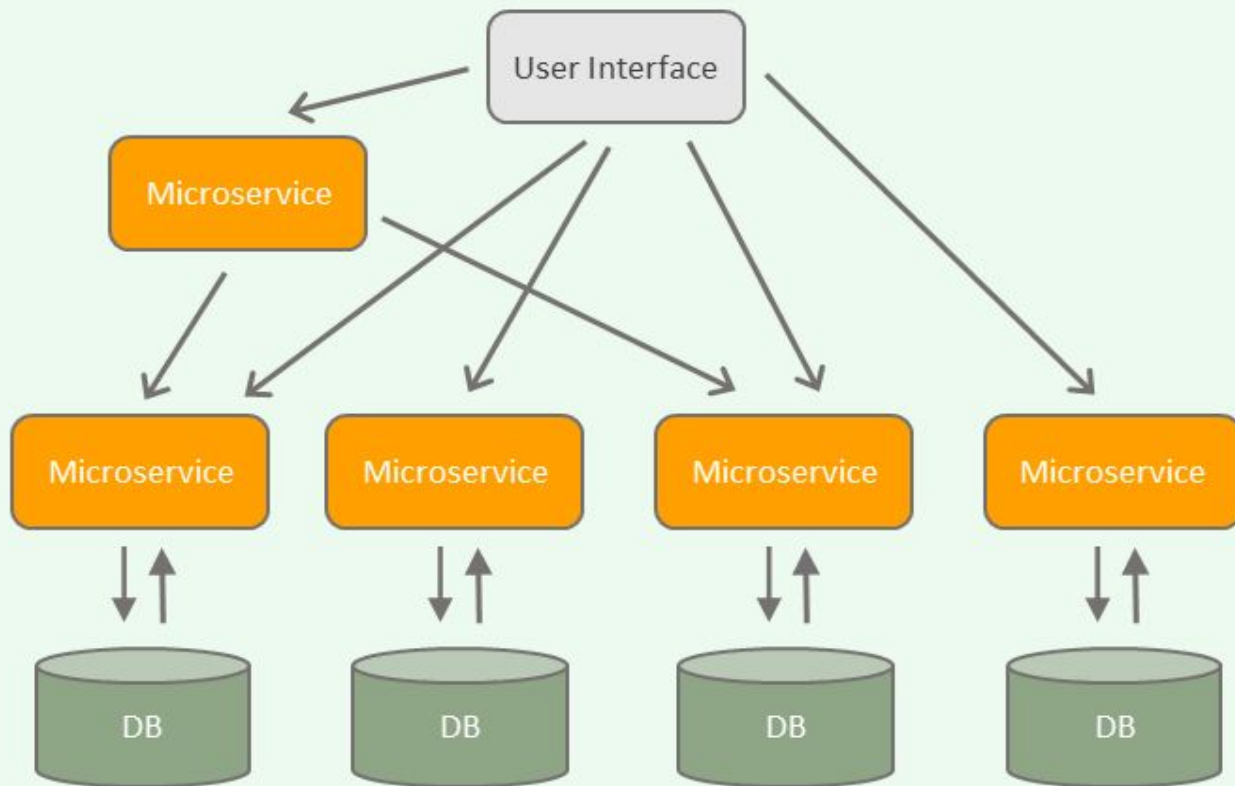
Nicolás Teare

# MICROSERVICIOS

## MONOLITHIC ARCHITECTURE



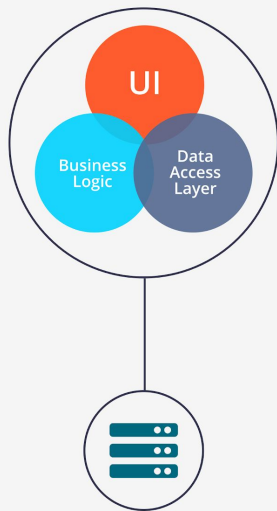
## MICROSERVICES ARCHITECTURE



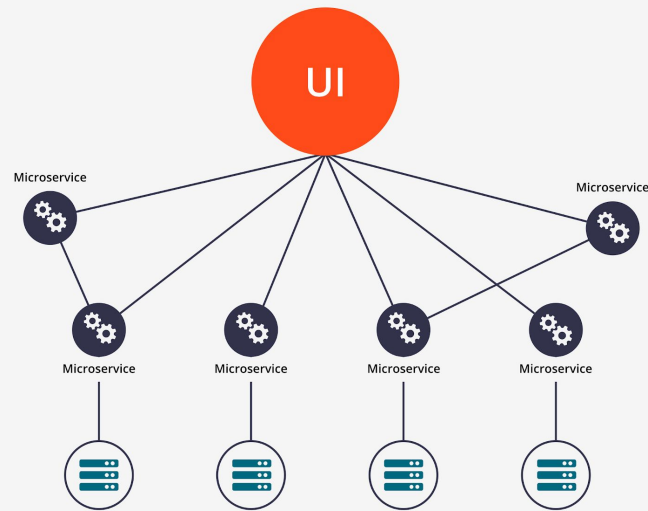
# Microservicios

## Ventajas:

- Independencia de Lenguajes
- Equipos pequeños
- Tolerante a las fallas
- Es muy compatible con contenedores
- Escalable



Monolithic Architecture

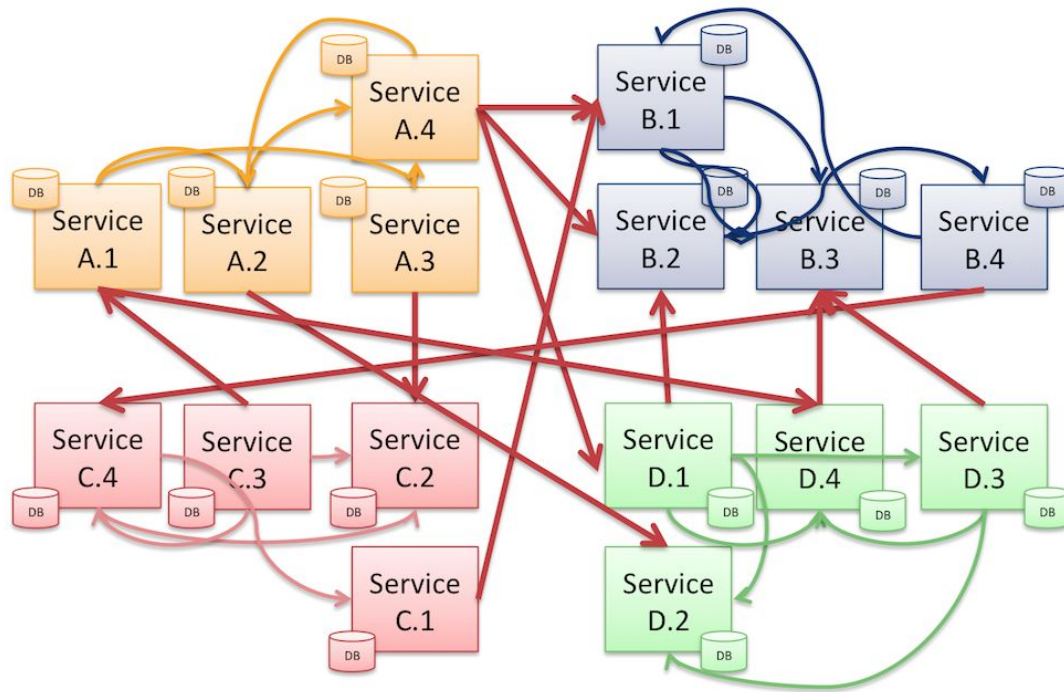


Microservice Architecture

# Microservicios

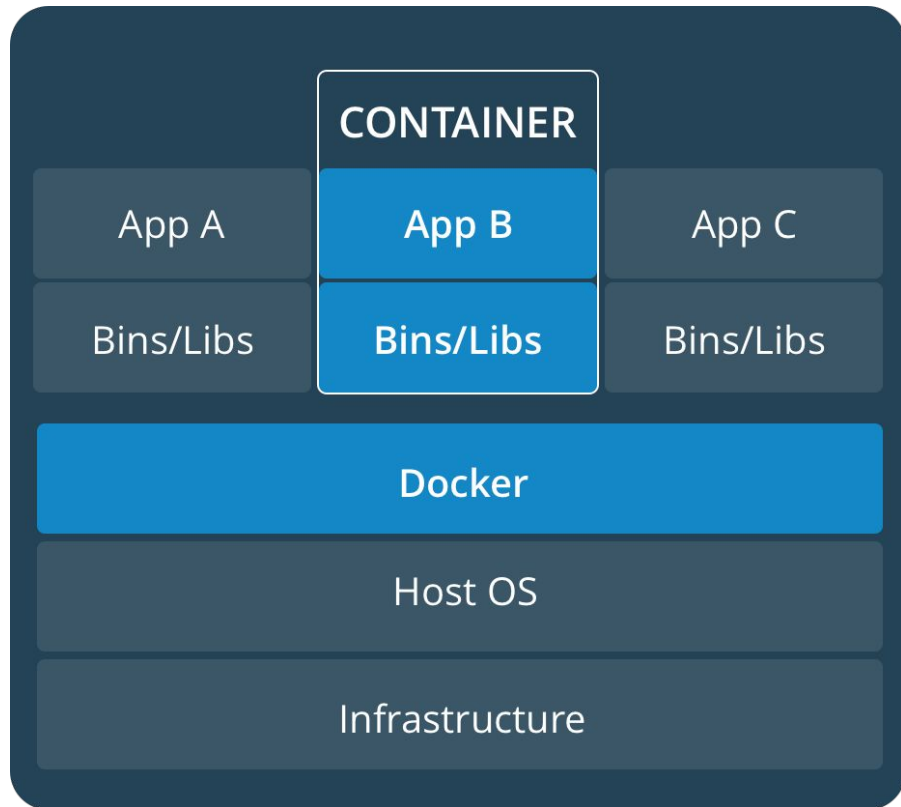
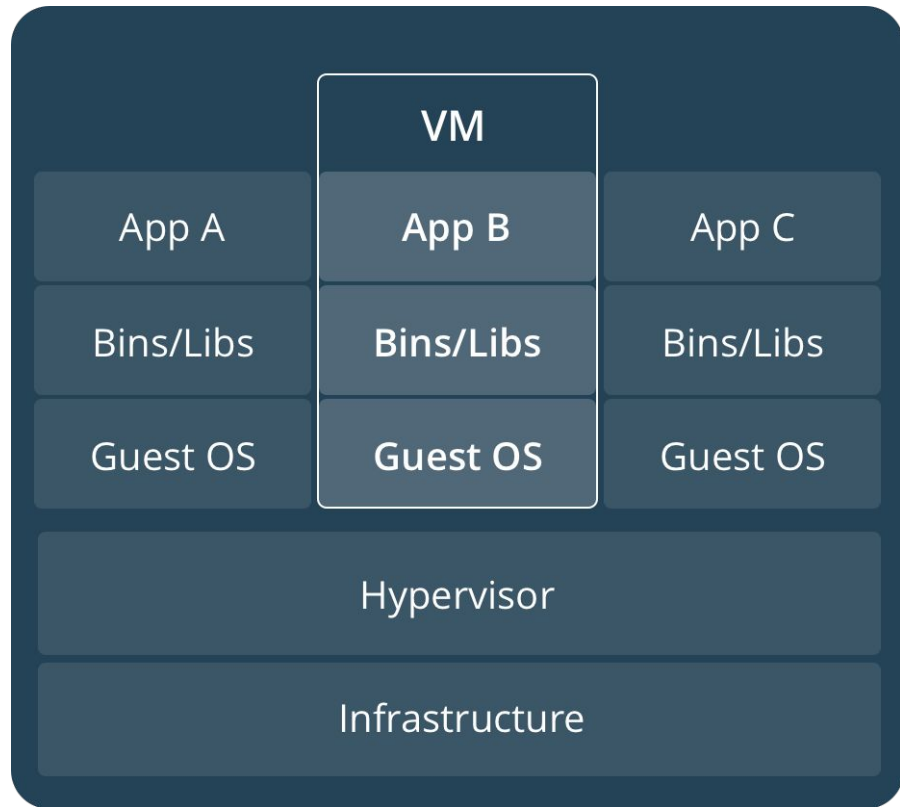
## Desventajas:

- Networking complejo
- Manejo de transacciones en base de datos
- Testing del sistema puede volverse complejo
- Deployment requiere mucho más esfuerzo

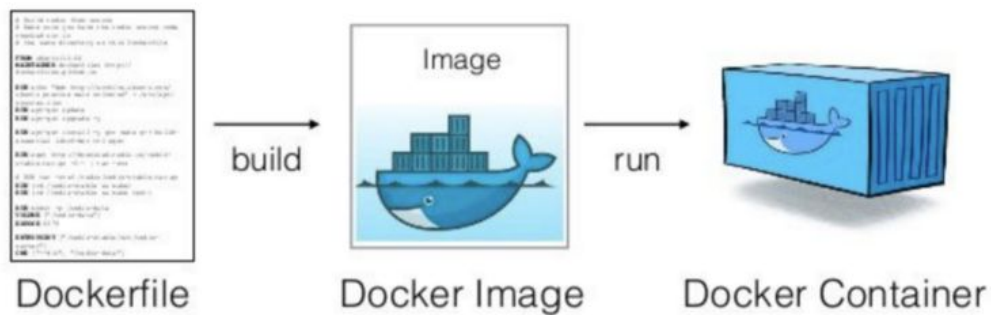


# Contenedores

# Docker



# Docker





**65%**  
of orgs have  
challenges  
maintaining  
legacy apps

**59%**  
of orgs have  
challenges from  
inertia of legacy apps  
and infrastructure

**39%**  
of orgs are modernizing  
legacy apps

**44%**  
of orgs are adopting  
microservices



**78%**  
are using, or planning to  
use, Docker to build new  
microservices applications.



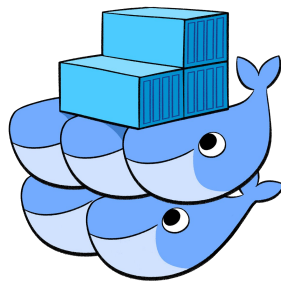
**71%**  
are using, or planning to  
use, Docker to containerize  
a legacy app.

# Orquestadores

# Orquestadores

Sistemas que agrupan contenedores

- Configuración Automática
- Networking
- Controller/Management unit
- Monitoreo de estado



**Amazon ECS**



**kubernetes**



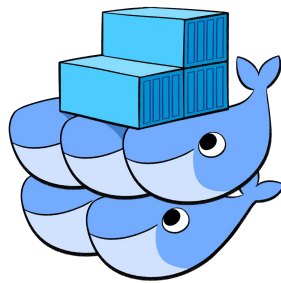
HashiCorp

**Nomad**

# Orquestadores

Principios fundamentales:

- Tolerancia a fallas
- Escalamiento on-demand
- Uso óptimo de recursos
- Apps accesibles desde el exterior
- *Update/rollback sin downtime*



Amazon ECS



kubernetes



HashiCorp

Nomad

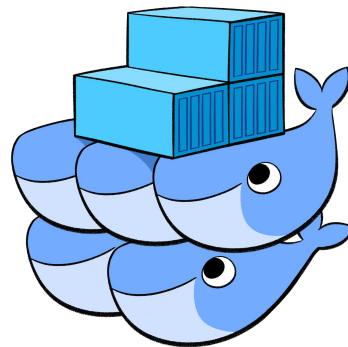
# Docker Swarm

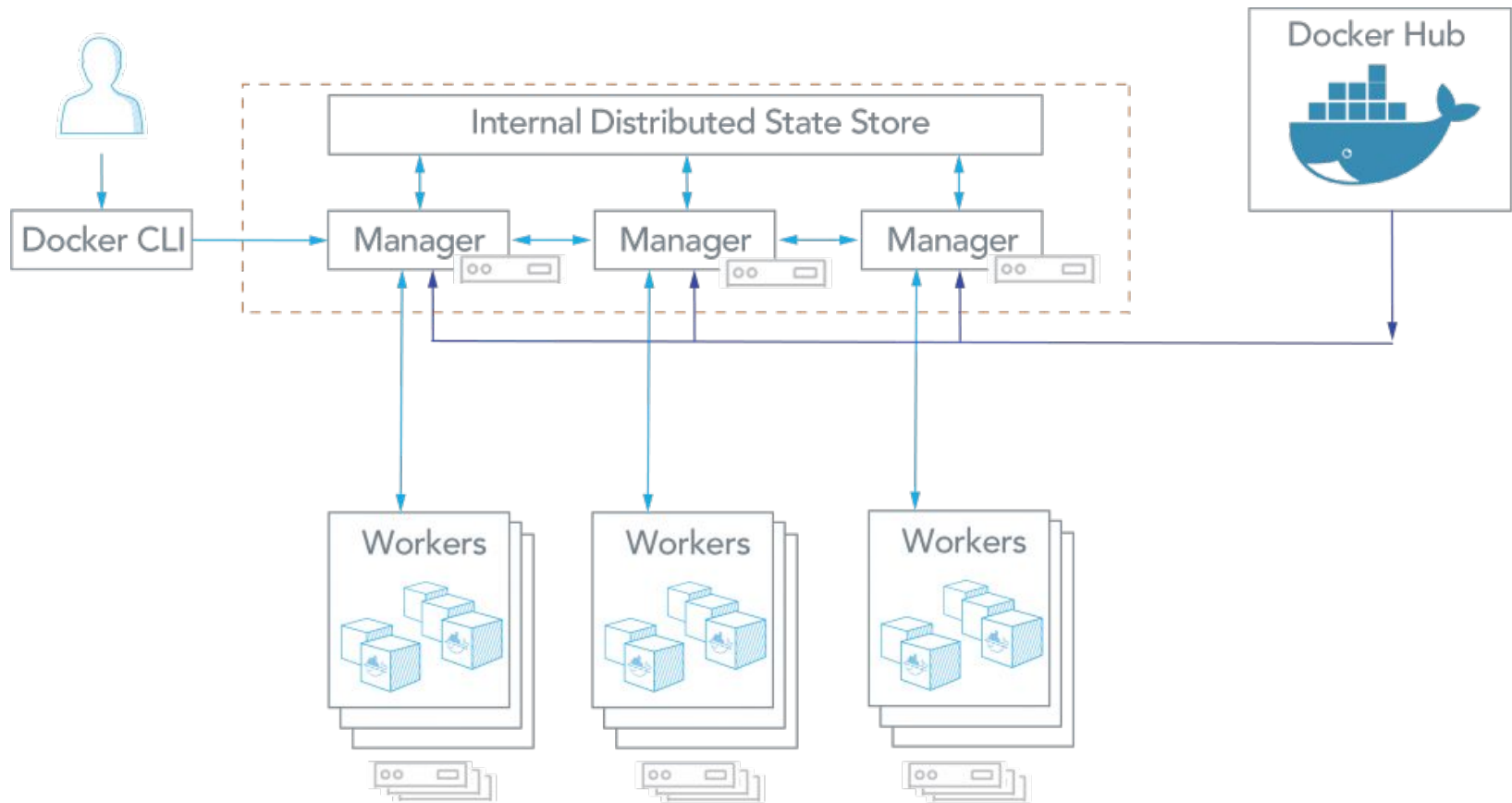
# DockerSwarm

“Docker swarm mode allows you to manage a cluster of Docker Engines, natively within the Docker platform”

- Arquitectura:

- **Manager:** Se encargan de mantener el estado del cluster. Hacer el scheduling de servicios y además de asignar tareas por protocolos HTTP a los workers.
- **Leader Node:** Es el nodo master, encargado de la toma de decisiones (orquestador). Se elige mediante votación utilizando *Raft Consensus Algorithm*.
- **Worker Node:** Reciben y ejecutan los tasks recibidos. Son instancias de Docker Engine.
- **Docker Hub:** Repositorio en la nube de contenedores. Se utiliza para actualizar los contenedores en los workers a través de la API.
- **Swarm:** Modo en el que se configuran múltiples Docker hosts para trabajar en conjunto utilizando Managers.





Manager: a node that dispatches tasks

Worker: a node that executes tasks provided by a Manager

Internal Distributed Store: used to maintain cluster state

Docker CLI: User interacts with the swarm using Docker CLI, for example "docker service"

Docker Hub: contains repositories for downloading and sharing container images

# Kubernetes



# Kubernetes

“Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.”

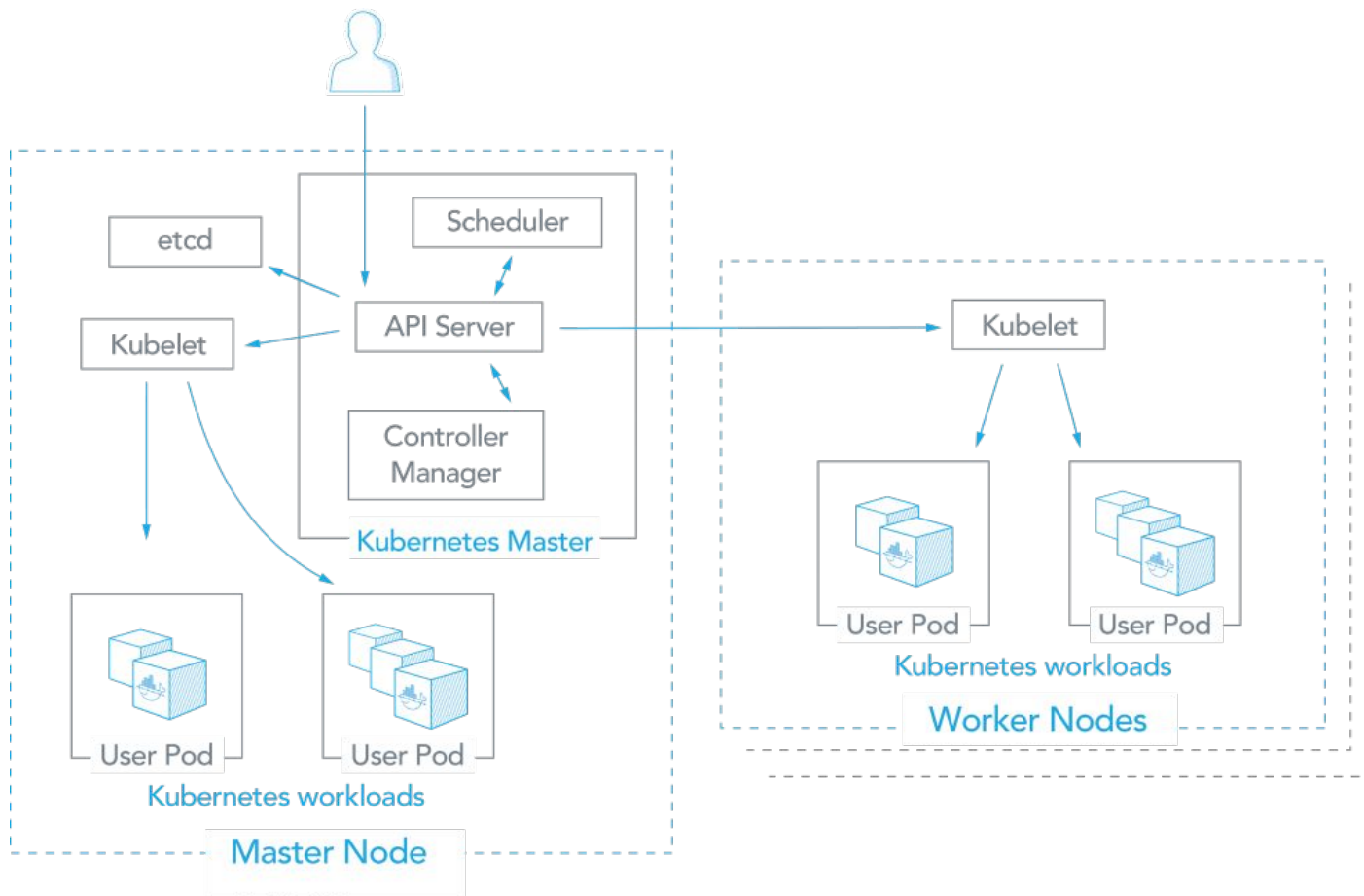
- Originalmente es un proyecto de Google que luego fue *donado* a la comunidad Open Source, Cloud Native Computing Foundation.

- Arquitectura:

- **etcd**: Guarda la configuración de los datos para el uso de los Master Nodes.
- **API Server**: Es el servidor que facilita la comunicación entre los distintos servicios de los *workers*.
- **Controller Manager**: Se encarga de monitorear el estado del sistema y escalar los *workers*.
- **Scheduler**: Se encarga de conectarse a los workers y asignarle tareas.
- **Kubelet**: Se encarga de levantar los *pods* y los *containers*.





kubernetes



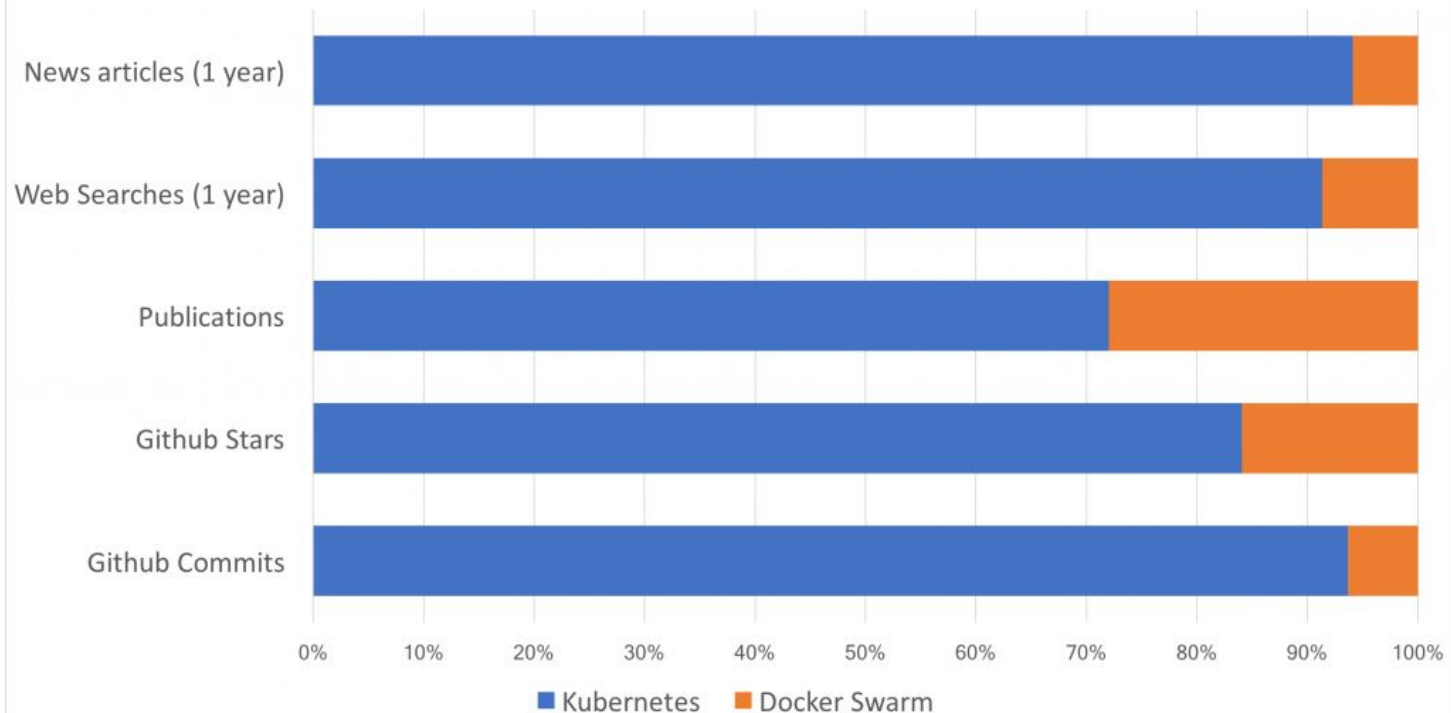
API Server: management hub for Kubernetes  
Scheduler: places a workload on the appropriate Node  
Controller Manager: scales workloads up/down  
etcd: stores configuration data which can be accessed by API Server

Kubelet: Receives pod specifications from API Server, updates Nodes  
Master Node: places workloads on Nodes  
Worker Nodes: receives requests from Master Nodes and dispatches them  
User Pod: a group of containers with shared resources

# Comparación

		
Escalamiento	Automático, basado en métricas configurables	Manual
Instalación	Compleja (pero flexible)	Muy simple, incluido como herramienta en Docker
Integración Docker	Requiere herramientas extra (ej: kubectl)	Comandos comunes a Docker
Logging	Múltiples versiones incluidas	No incluido
Comunidad	50.000 commits de 1200 miembros	3.000 commits de 160 miembros

## Kubernetes vs. Docker Swarm Mindshare



# In a nutshell

¿Con cuál nos quedamos?

- Ambos permiten asegurar criterios similares
- Deployment rápido y simple: Docker Swarm
- Alta complejidad: Kubernetes

# ¿Cuándo puede ser útil un orquestador?



*“The traffic grew to 50x the original estimation - it was even 10x the estimated worst case scenario (!) within a very short time.”*



**kubernetes**