

ASYS

Por una humanidad sustentable

Dev Ops

Modulo 3

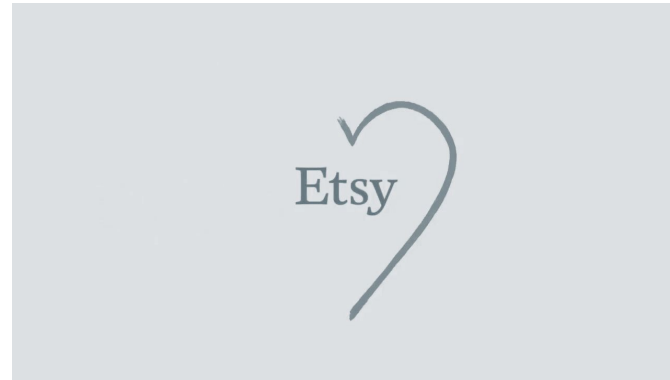
Sesión 2

Objetivos del Modulo

1. Hilo Rojo
2. Repaso de sesión 1
3. Mock de la Aplicación que vamos a desarrollar
4. Diagrama de Arquitectura
5. Docker
6. Kubernetes



Qué es DevOps ?



Desarrollar, Operar y Mejorar Sistemas de Excelente Calidad

Flujo

**Hacer el
trabajo**

**Lo más
eficiente
posible**

**Retro
Alimentación**

**Observar
Medir
Notificar**

**Mejora
Continua**

**Incrementar
el valor de
nuestro
trabajo**

Cultura

Flujo

Intro

Trabajo

Diseño

Calidad

Infraestructura

FrontEnd

BackEnd

DataEnd

Infraestructura de Producción on Demand

Pipeline - C.I. - C.D. - C.Deployment

Cultura

Retroalimentación

Telemetría FrontEnd

Telemetría BackEnd

Telemetría Negocio

Telemetría Infraestructura

Mejora Continua

Operaciones

Experimentación

Ingeniería del Caos

Objetivos del Modulo

1. Hilo Rojo
2. Repaso de sesión 1
3. Mock de la Aplicación que vamos a desarrollar
4. Diagrama de Arquitectura
5. Docker
6. Kubernetes

Monolito VS Microservicios

Aplicación distribuida

Escalamiento Horizontal VS Escalamiento Vertical

Diseño para Operaciones

Repaso

Objetivos del Modulo

1. Hilo Rojo
2. Repaso de sesión 1
3. Mock de la Aplicación que vamos a desarrollar
4. Diagrama de Arquitectura
5. Docker
6. Kubernetes

Link

Mock

Objetivos del Modulo

1. Hilo Rojo
2. Repaso de sesión 1
3. Mock de la Aplicación que vamos a desarrollar
4. Diagrama de Arquitectura
5. Docker
6. Kubernetes

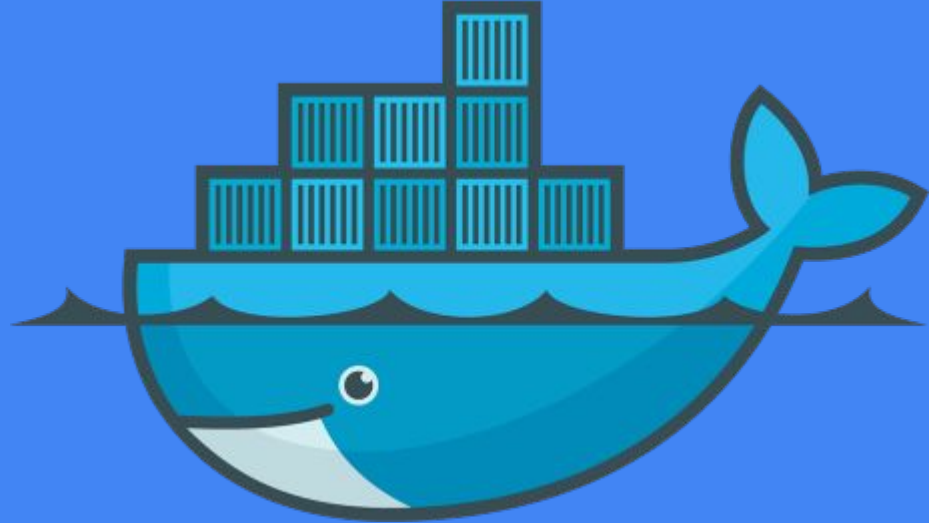
[Link](#)

Diagrama de Arquitectura

Objetivos del Modulo

1. Hilo Rojo
2. Repaso de sesión 1
3. Mock de la Aplicación que vamos a desarrollar
4. Diagrama de Arquitectura
5. Docker
6. Kubernetes

Docker



Que es ?

Docker

Aislar

APLICACIONES

Estandarizar

ADMINISTRACION

Herramienta

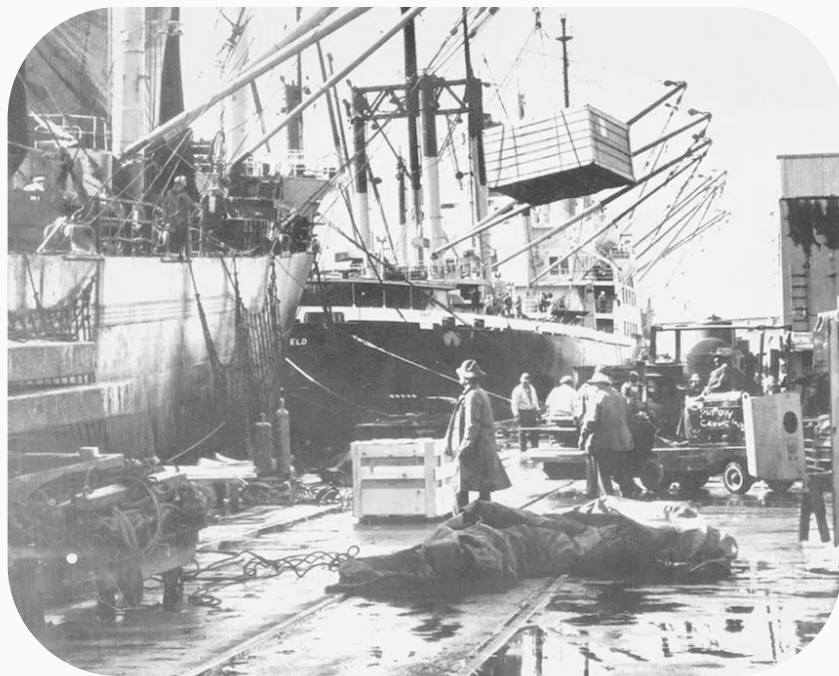
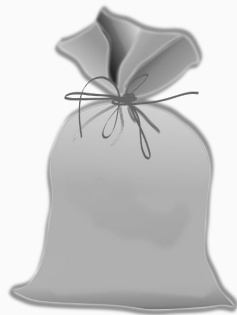


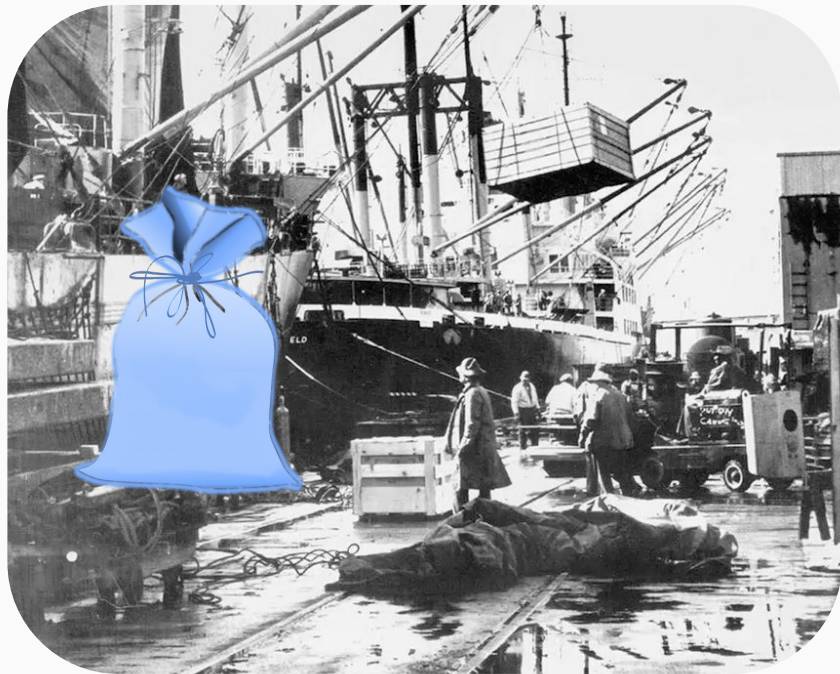
Y eso

Para qué sirve

?

Juguemos





Breve

Historia



“Transporte Mercancías”



Antes y Después

Transporte

ANTES





Transporte

DESPUÉS

Ventajas

De trabajar con

- Aislar las mercancías
- Estandarizar el transporte
- Evitar riesgos a los humanos
- Evitar riesgos a las mercancías

CONTENEDORES

**El concepto del
Contenedor
es para las**

APLICACIONES

LO QUE

**El concepto del
Contenedor
es para las**

MERCANCÍAS

Ventajas

De trabajar con

- Aislar las aplicaciones
- Estandarizar el transporte
- Evitar riesgos a los humanos
- Evitar riesgos a las aplicaciones

CONTENEDORES

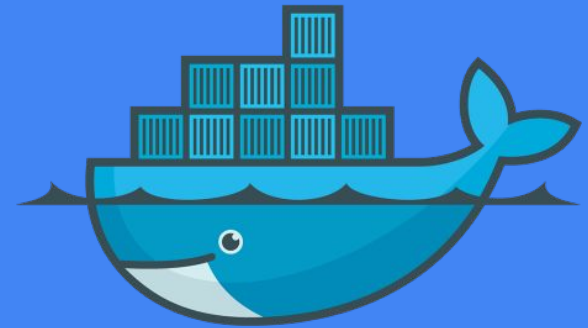
Aislar

APLICACIONES

Estandarizar

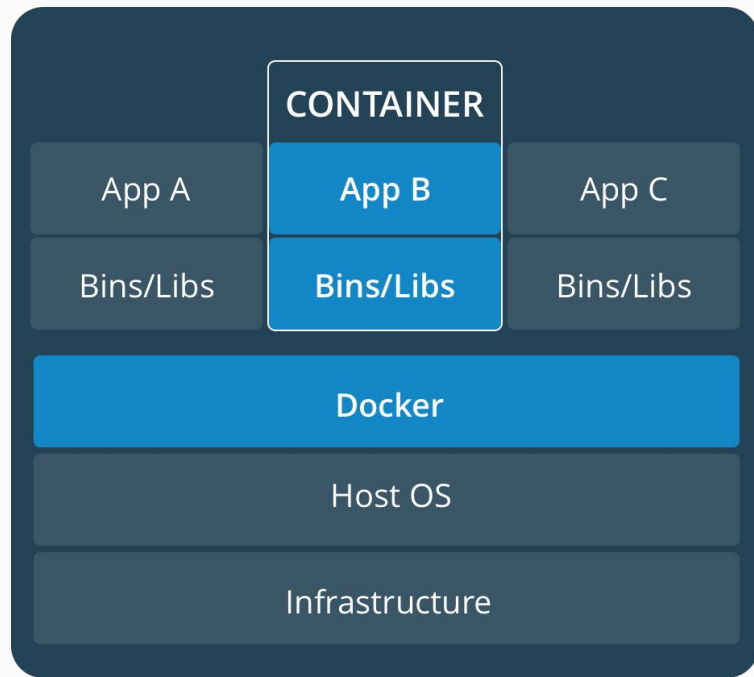
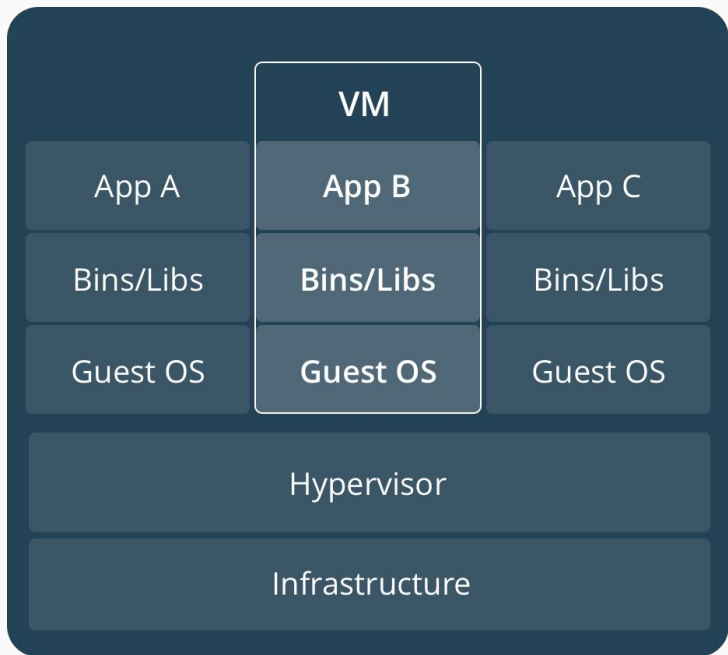
ADMINISTRACION

Herramienta



Explicame la Magia

Las máquinas virtuales no hacen lo mismo ??



Componentes de **Docker**

Docker

FILES



Son el diseño escrito para crear las imágenes de docker

Docker

IMAGES



Son los moldes para crear los
contenedores

Docker

CONTAINER



Son las instancias vivas de una imagen de Docker

Es lo que contiene en definitiva a nuestra aplicación y sus dependencias

Docker

ENGINE



Maneja a las imágenes y contenedores

- Crea
- Elimina
- Inspecciona
- Administra

Docker

REGISTRY



Es un repositorio de images de Docker

Un ejemplo de Docker registry es Docker hub

Docker

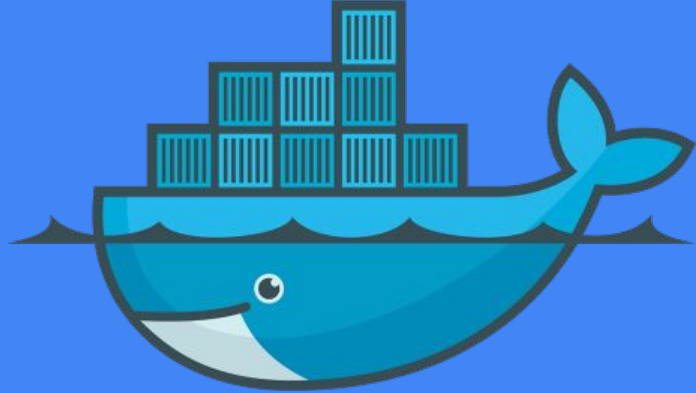
SWARM



Administra Clusters (Grupos) de
Docker Engines

Objetivos del Modulo

1. Hilo Rojo
2. Repaso de sesión 1
3. Mock de la Aplicación que vamos a desarrollar
4. Diagrama de Arquitectura
5. Docker
6. Kubernetes



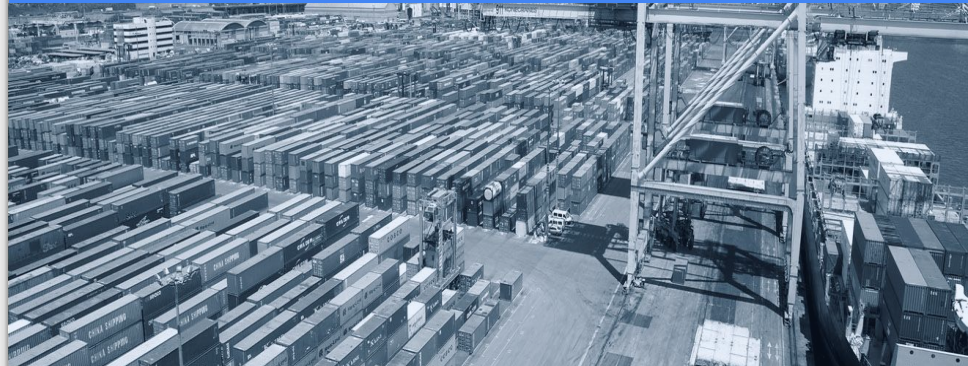
Kubernetes

Setup

Kubernetes



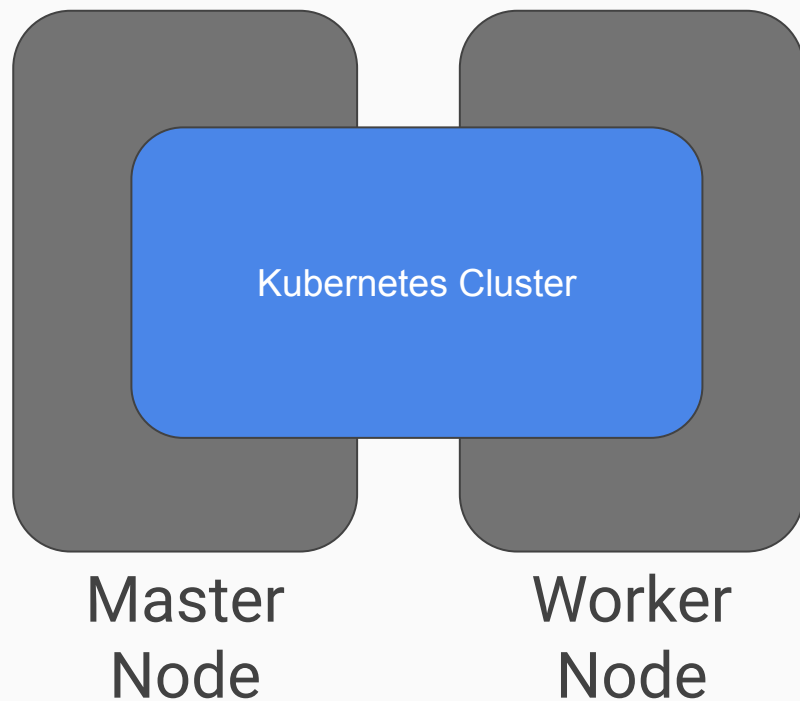
Manager docker containers



Infrastructure Components

Kubernetes

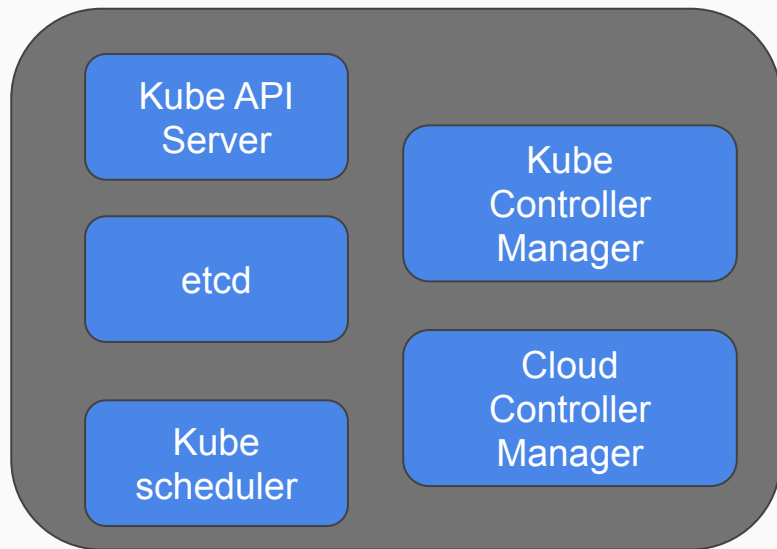
Nodes



Virtual Machines or Bare Metal machines

That actually run kubernetes

Master Node



Kube API Server : Control Panel

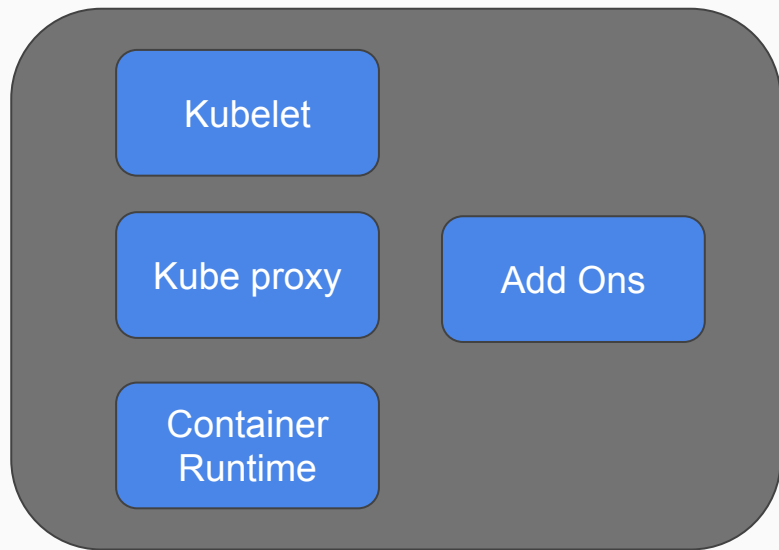
Etcd : Database

Kube-scheduler : Assigns pods to nodes

Kube Controller Manager :

- Node Controller
- Replication Controller
- Endpoints Controller
- Service Account and Tokens Controllers

Worker Node



Kubelet : Containers supervisor

Kube proxy : Handle cluster networking and service abstraction

Container Runtime : Containers manager like Docker Engine, RKT Engine

Add Ons : Extend functionality of kubernetes some of them

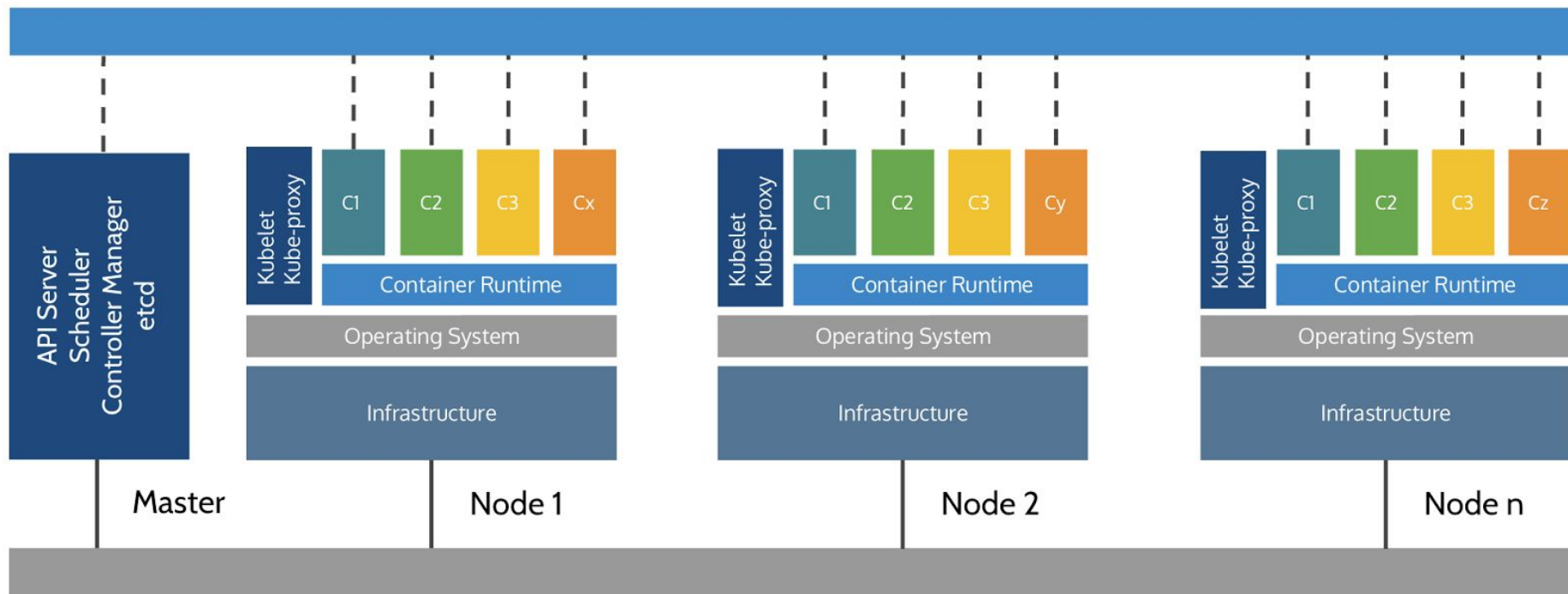
- Networking
- Visualization and Control

Overview

Architecture

Architecture

Overlay Network (Flannel/OpenVSwitch/Weave)

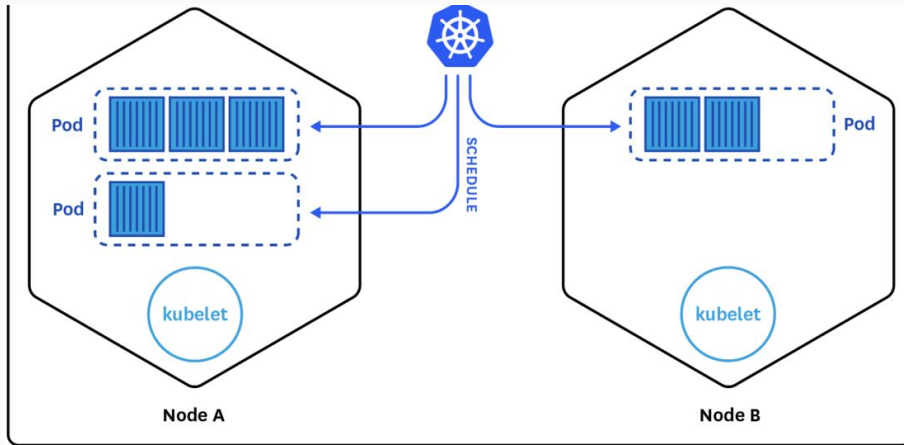


Physical Network

Components

Kubernetes

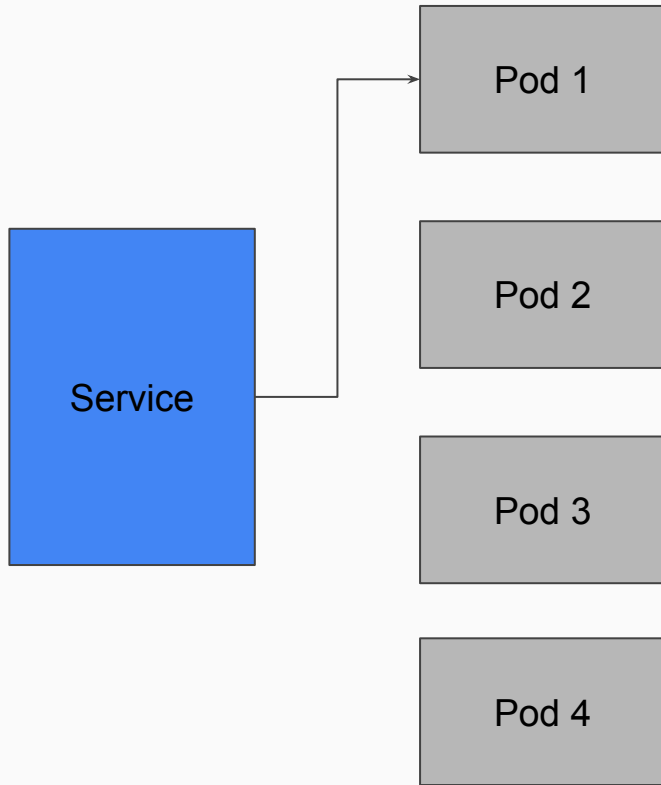
Pods



IT's a container of docker containers

All the containers within the pod share a single IP address

Services



IT's our

LOAD BALANCER for pods and
works together with the SERVICE
DISCOVERY mechanism

Volumes



IT's our

PERSISTENCE LAYER

These components can be mapped to defined file system to manage data

Objetivos del Modulo

1. Hilo Rojo
2. Repaso de sesión 1
3. Mock de la Aplicación que vamos a desarrollar
4. Diagrama de Arquitectura
5. Docker
6. Kubernetes