

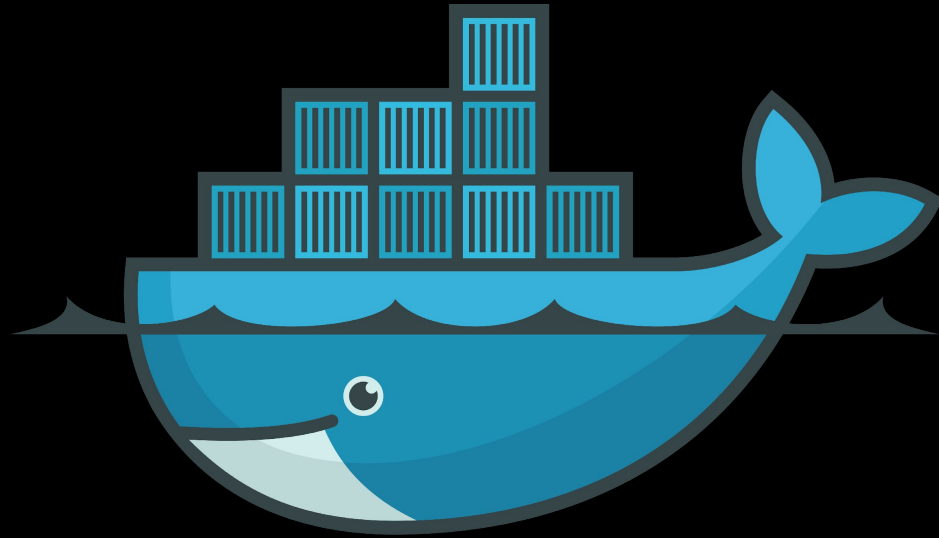
# Sistemas Operativos

---

Contenerización Docker

**Lic. R. Alejandro Mansilla**

Licenciatura en Ciencias de la Computación  
Facultad de Ingeniería  
Universidad Nacional de Cuyo



docker




# Docker: como nació?



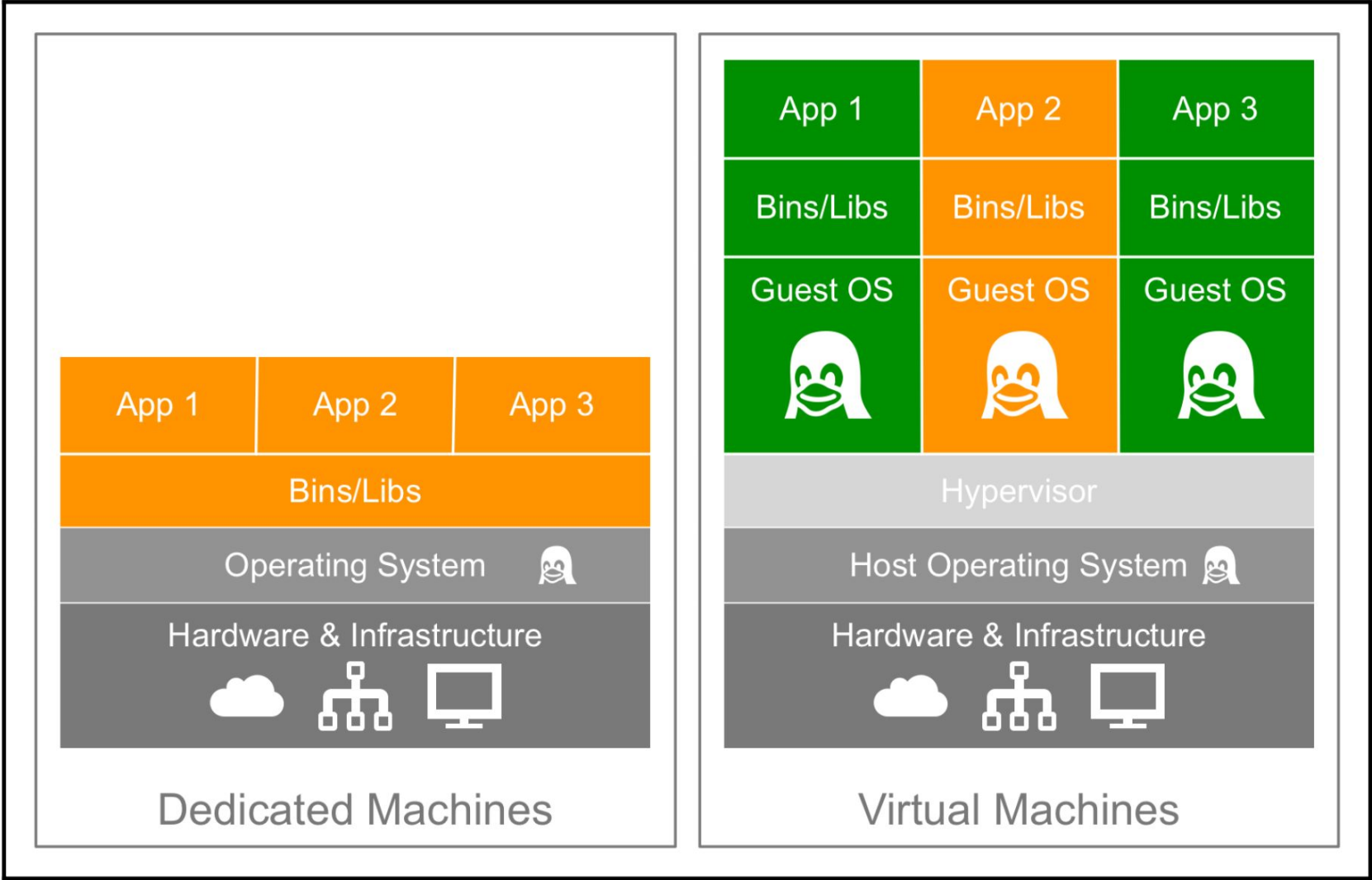
# Docker: Conceptos

Una plataforma para:

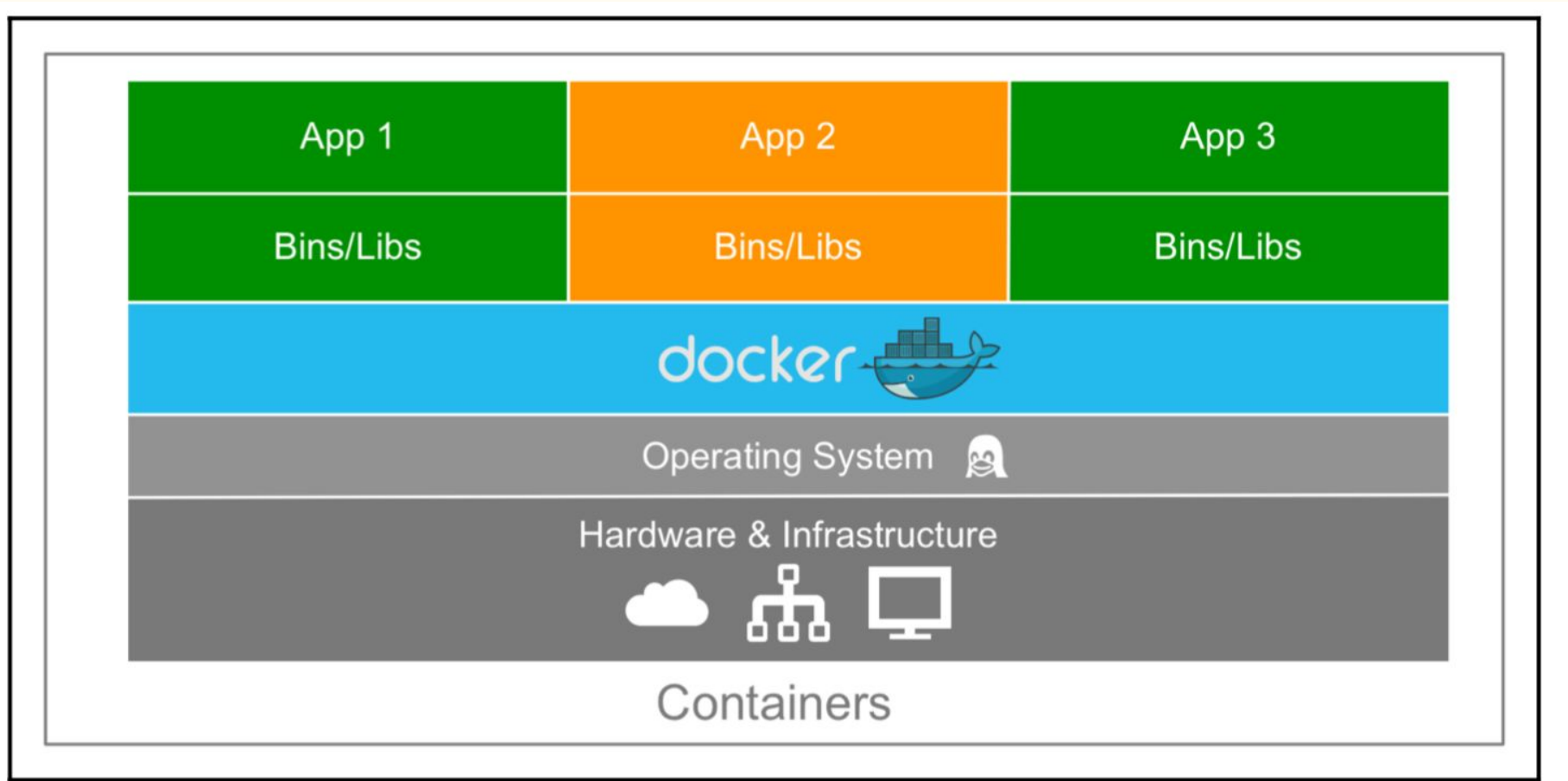
**“desarrollar, desplegar y ejecutar aplicaciones con containers”**

- Una “**imagen**” es un paquete ejecutable que contiene todo lo necesario para ejecutar una aplicación [código, librerías de ejecución, ambiente y configuración]
- Un “**container**” es una instancia de ejecución de una imagen.
- Docker fue creado en el año 2013 como una evolución de LXC
- Escrito en Go 

# Evolución



# Docker



# Docker: CLI

**## Ejecutamos una imagen de ejemplo**

```
docker run hello-world
```

**#--name asigna nombre, -d Daemon mode]**

```
docker run --name webserv -d nginx
```

**## Ejecutamos un comando dentro de un container en ejecución**

```
docker exec -it webserv bash
```

**## Listamos las imágenes disponibles localmente**

```
docker image ls
```

**## Listamos los containers**

```
docker container ls
```

**## Listamos los containers en ejecución**

```
docker ps
```

**## Detenemos un container en ejecución**

```
docker stop <container_id|name>
```

**## Matamos un container en ejecución**

```
docker kill <container_id|name>
```

**## Eliminamos un container [apagado]**

```
docker rm <container_id|name>
```

**## Ejecutamos una imagen de la distro Alpine en modo interactivo**

```
docker run -it alpine
```

# Docker: Dockerfile

Un Dockerfile define qué va dentro del ambiente de un container.

```
FROM alpine
LABEL maintainer="amansilla@gmail.com"
ENV TERM xterm
RUN apk update
RUN apk add python py2-pip uwsgi uwsgi-python
RUN mkdir -p /opt/ip-mac
WORKDIR /opt/ip-mac
COPY src/* /opt/ip-mac/
RUN pip install -r requirements.txt
USER 1000:1000
CMD uwsgi --ini /opt/ip-mac/app.ini

EXPOSE 8000
```

**FROM:** Origen de la imagen Base

**LABEL:** Adiciona Metadata

**ENV:** Setea variables de entorno

**RUN:** Ejecuta en un shell en la imagen

**WORKDIR:** Setea directorio actual

**COPY:** Copia a la imagen

**USER:** Setea el user para RUN,CMD,ENTRYPOINT

**CMD:** Ejecución por defecto, solo 1

**EXPOSE:** Publica un puerto del container



# Docker: Construyendo nuestra app mac-to-vendor

```
## Clonamos el repositorio de la clase
```

```
git clone https://github.com/c0r0nel/Clase\_Docker.git
```

```
## Ingresamos al repo recién clonado
```

```
cd Clase_Docker/mac-to-vendor
```

```
## Construimos la imagen, cuidado con el punto final, debe ir!
```

```
docker build -t ipmac:01 .
```

```
## Ejecutamos la app
```

```
docker run -p 8001:8000 -t ipmac:01
```

```
## Probamos desde otra consola
```

```
curl -s http://56ca8f8738ecc03cc4b2859db342e243@localhost:8001/f4:8c:50:9a:a3:03
```

# Docker: Publicando en una registry

## Nos Registramos en <https://canister.io/>

## Dentro de canister creamos un repositorio llamado ipmac

```
docker login -u="c0r0nel" -p="#CHINITOS#CHINITOS#CHINITOS" cloud.canister.io:5000
```

## Taggeamos las imágenes y la publicamos

```
docker tag ipmac:01 cloud.canister.io:5000/c0r0nel/ipmac:01
```

```
docker push cloud.canister.io:5000/c0r0nel/ipmac:01
```

```
docker pull cloud.canister.io:5000/c0r0nel/ipmac:01
```

# Docker: docker-compose

Compose es una herramienta para definir y ejecutar aplicaciones multi-container. A través de un archivo YAML se definen los servicios de la aplicación a modo de manifest. Luego, con un simple comando se crean todos los containers que componen la aplicación.

```
## Ingresamos al directorio samba-plex-rutorrent del repo
```

```
cd Clase_Docker/samba-plex-rutorrent
```

```
less docker-compose.yml
```

```
## arrancamos el conjunto
```

```
docker-compose up -d
```

```
## Probamos en un navegador Plex y rutorrent
```

```
http://localhost:32400/web/index.html
```

```
http://localhost:8080
```

```
## Probamos el servidor samba en un administrador de archivos o desde la consola con:
```

```
sudo apt install -y smbclient; smbclient -L localhost
```

# Referencias bibliográficas

- Docker Documentation. “Get Started, Part 1: Orientation and Setup,” October 31, 2019. <https://docs.docker.com/get-started/>.
- Docker Documentation. “Overview of Docker Compose,” October 31, 2019. <https://docs.docker.com/compose/>.
- McKendrick, Russ, and Scott Gallagher. **Mastering Docker: Master This Widely Used Containerization Tool**. Second edition. Birmingham, Mumbai: Packt, 2017.
- Raj, Pethuru, Jeeva S. Chelladhurai, and Vinod Singh. **Learning Docker: Optimize the Power of Docker to Run Your Applications Quickly and Easily**. 2. Auflage. Community Experience Distilled. Birmingham Mumbai: Packt Publishing, 2017.
- Khare, Neependra. **Docker Cookbook: 80 Hands-on Recipes to Efficiently Work with the Docker 1.6 Environment on Linux**. Quick Answers to Common Problem. Birmingham Mumbai: Packt Publishing, 2015.
- Langemak, Jon. **Docker Networking Cookbook: 60 Practical Recipes That Help You Gain Expertise with Docker Networking and Understand the Underlying Constructs That Make It All Work**. Birmingham Mumbai: Packt, 2016.