Docker & Cloud

Una breve introducción a los contenedores enfocado especialmente en Docker.

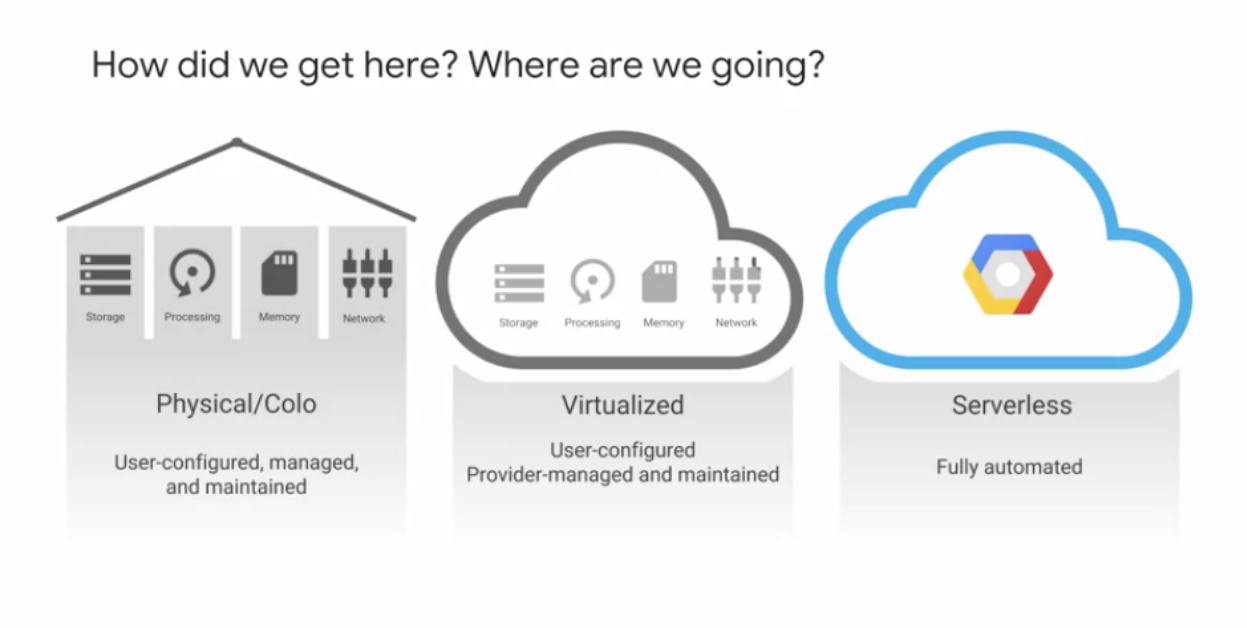
Principio de los contenedores

Un contenedor es una caja metálica sellada,
rígida y reutilizable que se utiliza para contener

mercancías que requieren transporte por barco, camión o ferrocarril.

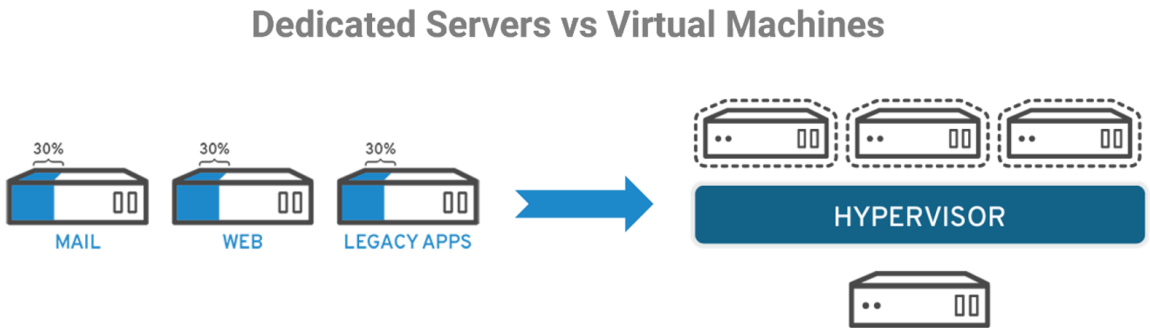
El contenedor debe estar construido para un uso repetido, fácil de llenar o vaciar y especialmente diseñado para facilitar el transporte y separación de mercancías.

Evolución



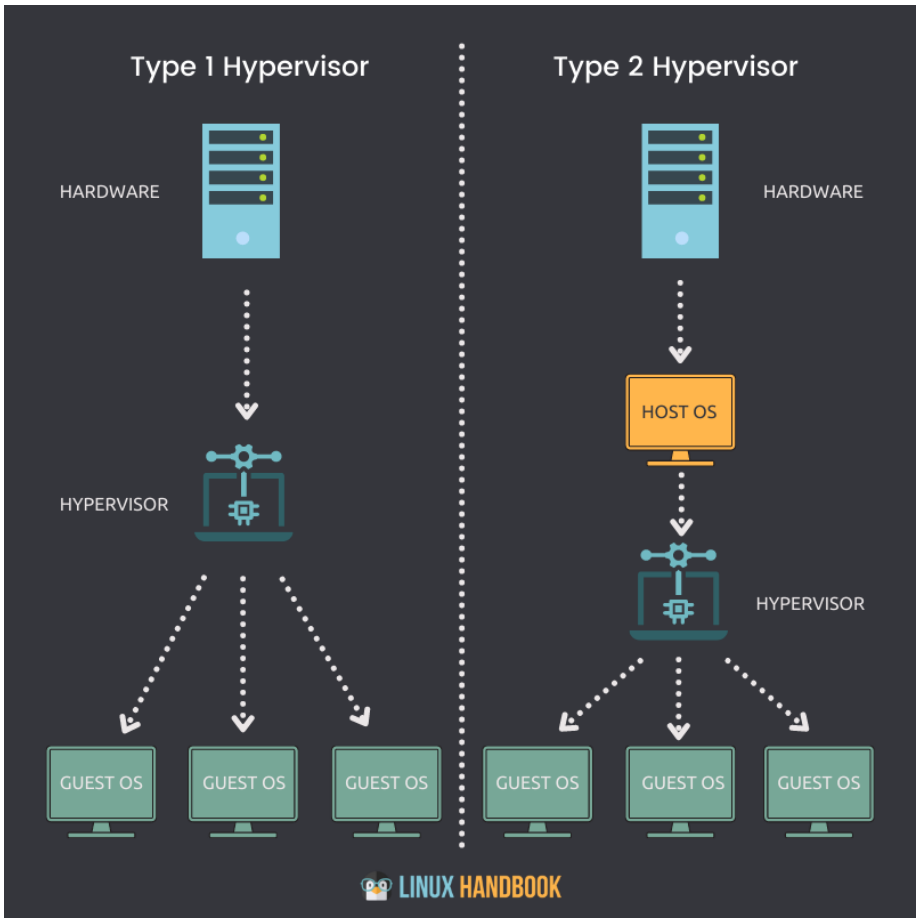
https://techprodezza.code.blog/2020/06/14/google-cloud-platform-gcp/

Colocation

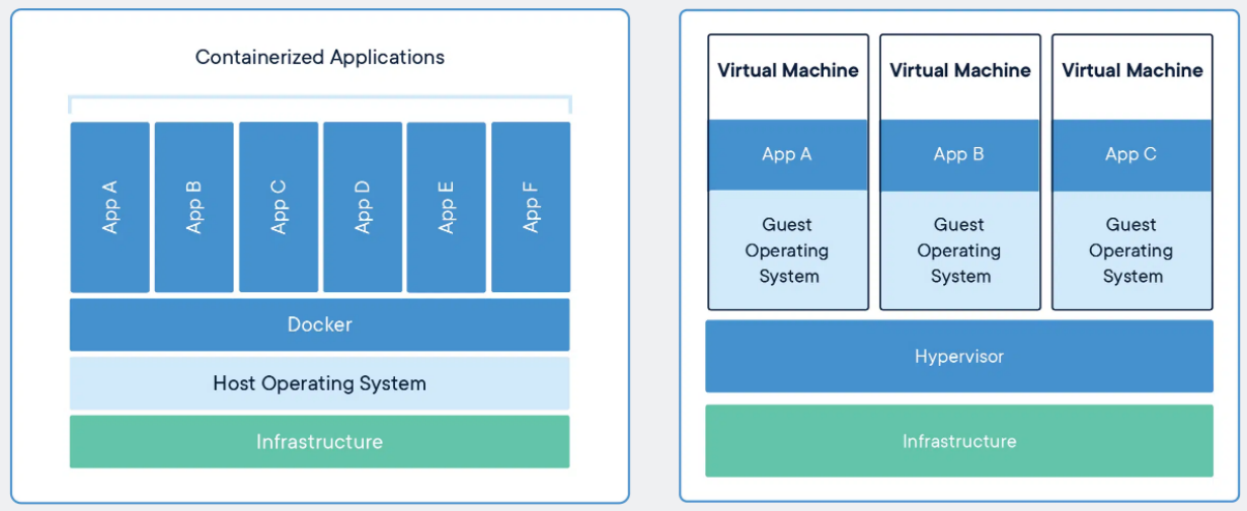


https://blog.equinix.com/blog/2020/02/25/how-to-speak-like-a-data-center-geek-bare-metal-vs-virtualization-vs-serverless/

Virtualization

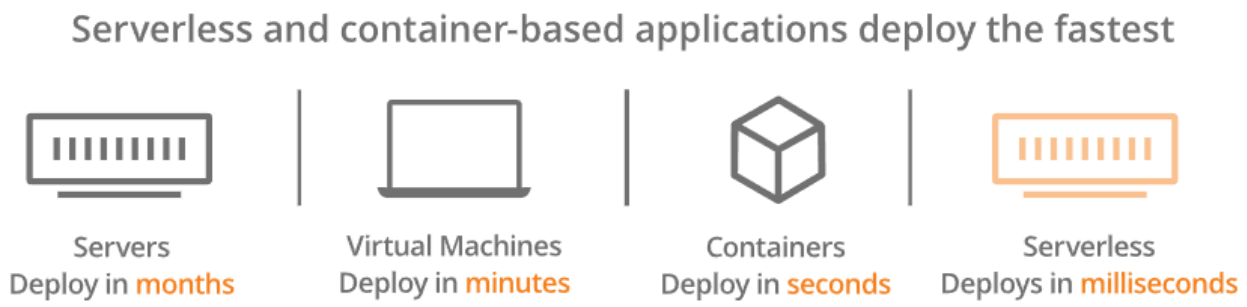


https://linuxhandbook.com/what-is-hypervisor/

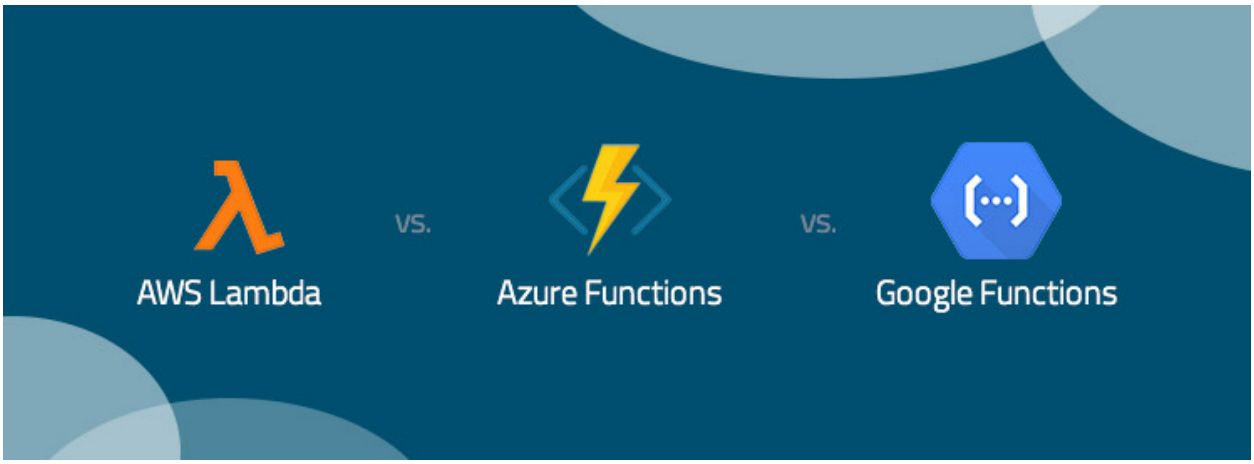


Comparación y explicación de los contenedores vs VMs

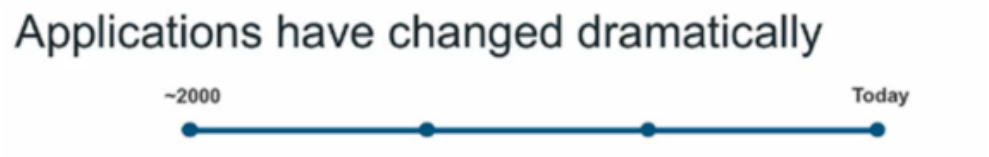
Serverless



https://www.cloudflare.com/es-es/learning/serverless/serverless-vs-containers/



Aplicaciones monolíticas vs microservicios



Monolithic

Slow Changing

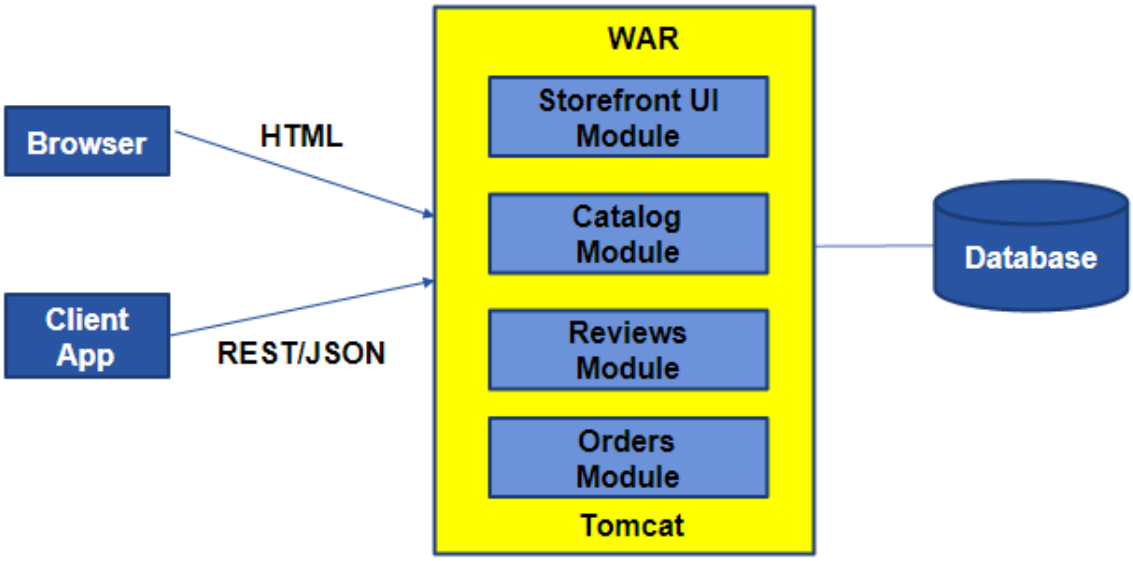
Big Server

A Decade Ago (and still valid)

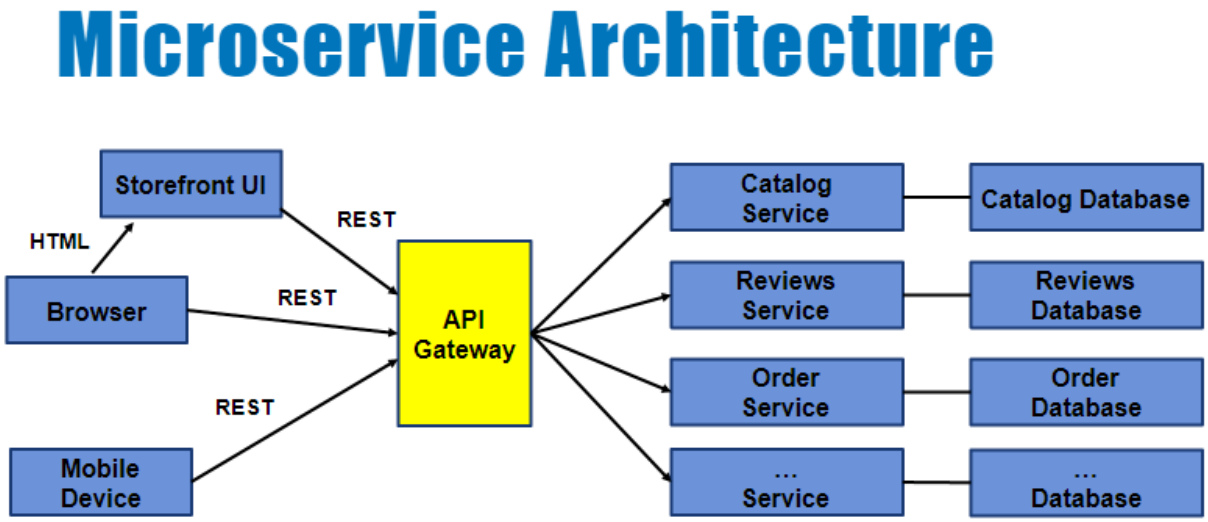
- Apps were monolithic
- Built on a single stack such as .NET or Java
- Long Lived
- Deployed to a single server

https://dockerlabs.collabnix.com/docker/Docker_VIT_Intro/Docker_VIT_Intro.html

A Close Look at Monolithic



https://dockerlabs.collabnix.com/docker/Docker_VIT_Intro/Docker_VIT_Intro.html



https://dockerlabs.collabnix.com/docker/Docker_VIT_Intro/Docker_VIT_Intro.html

Comparativa

	BARE METAL	VIRTUAL MACHINE	CONTAINER	SERVERLESS
Boot Time	~5-10 min	~2 min	~2 secs	~0.0003 secs
App Deployment Lifecycle	Deploy in weeks Live for years	Deploy in minutes Live for weeks	Deploy in seconds Live for minutes/hours	Deploy in milliseconds Live for seconds
Deployment Complexity	**Need to know:** 1. Hardware 2. OS 3. Runtime environment 4. Application code	**Need to know:** 1. OS 2. Runtime environment 3. Application code	**Need to know:** 1. Runtime environment 2. Application code	**Need to know:** 1. Application code
Investment	Buy/rent dedicated server	Rent a dedicated VM on a shared server	Rent containers and pay for actual runtime	Pay for compute resources used during runtime
Scaling	Can take months* Should be approved by a panel of experts	Takes hours Should be approved by adminstrators	Takes seconds Policy driven scaling	Takes milliseconds Event driven scaling

**For on-premises bare metal. Bare metal cloud services can be dynamically provisioned in minutes*

https://blog.equinix.com/blog/2020/02/25/how-to-speak-like-a-data-center-geek-bare-metal-vs-virtualization-vs-serverless/

Datos Relevantes

Fundado en el año 2010, lanzado en 2011 y liberado como proyecto open-source en Marzo de 2013. El lanzamiento de Docker inició una revolución en el desarrollo de aplicaciones, al democratizar los contenedores de software. Docker desarrolló una tecnología de contenedor de Linux, una que es portátil, flexible y fácil de implementar, además permite construir, distribuir y ejecutar cualquier aplicación en cualquier lado.

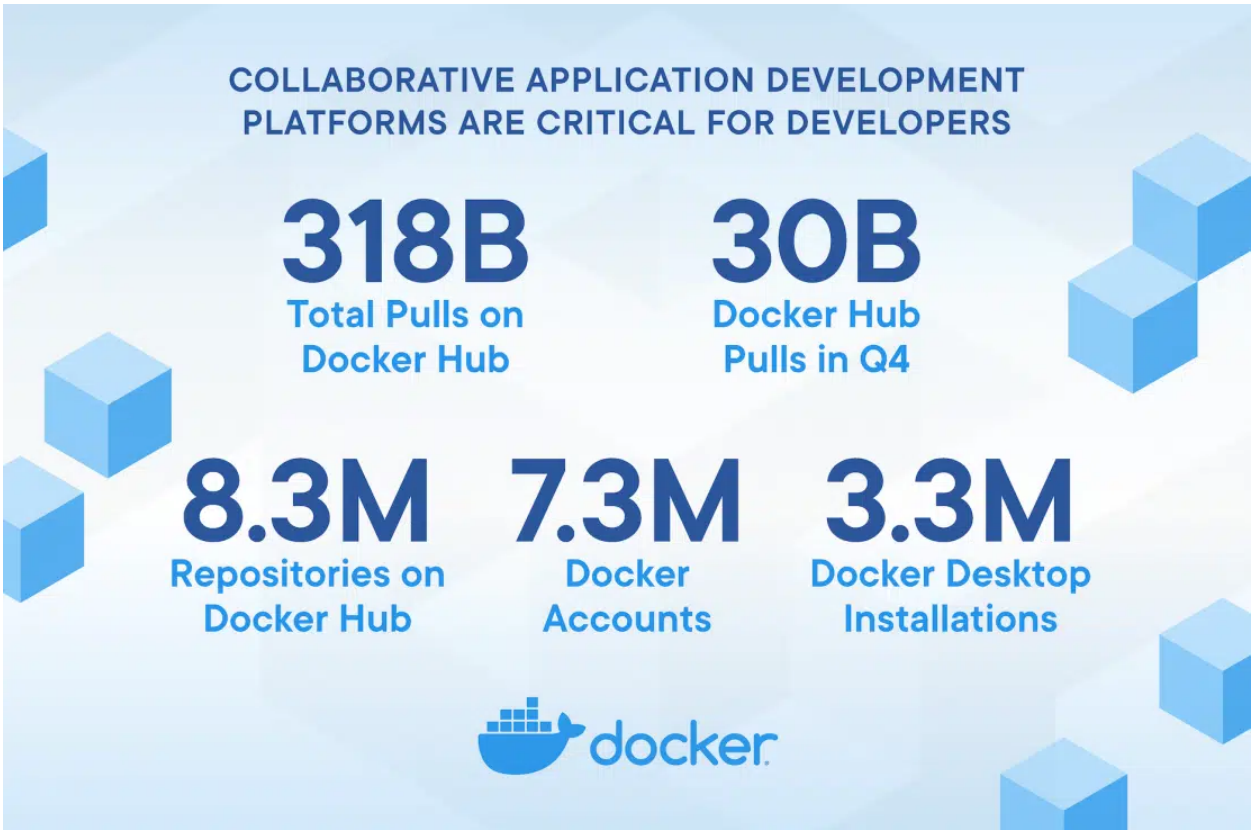
Accelerate how you build, share, and run modern applications.

13 million +
developers

7 million +
applications

13 billion +
monthly image downloads

See who uses Docker



https://www.docker.com/blog/docker-index-shows-continued-massive-developer-adoption-and-activity-to-build-and-share-apps-with-docker/

¿Por qué Docker?

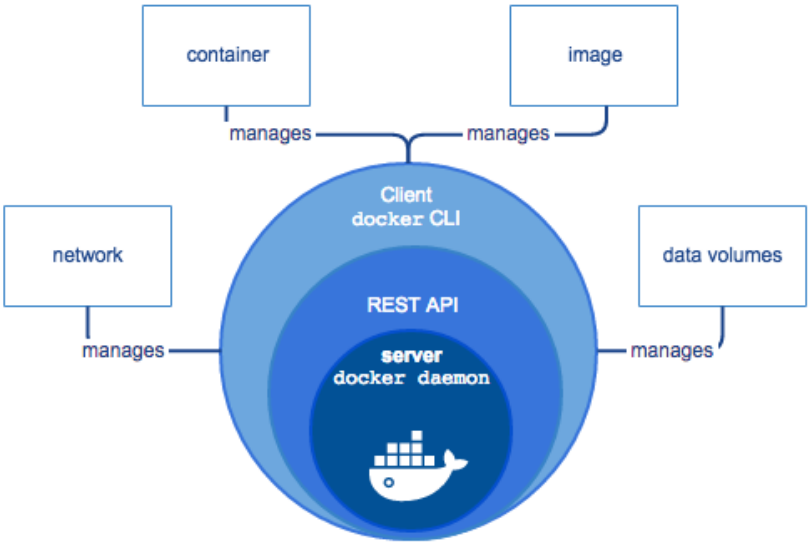
Problemas comunes

- Construir
- Distribuir
- Ejecutar
- Dependencias
- Entornos de ejecución (Hardware)
- Versionamiento
- Costos

Ventajas

- Flexible
- Distribuir
- Liviano
- Portable
- Escalable
- Efímero
- Seguros

Componentes



https://www.recursion.mx/docker/501/

Docker daemon: la que permite crear, destruir, escalar, es el que va a interactuar con el OS.

REST API: para comunicarse consigo mismo, remotamente, por medio de HTTP, para darle instrucciones al Docker daemon.

Docker CLI: para interactuar directamente con Docker por medio del cliente de Docker.

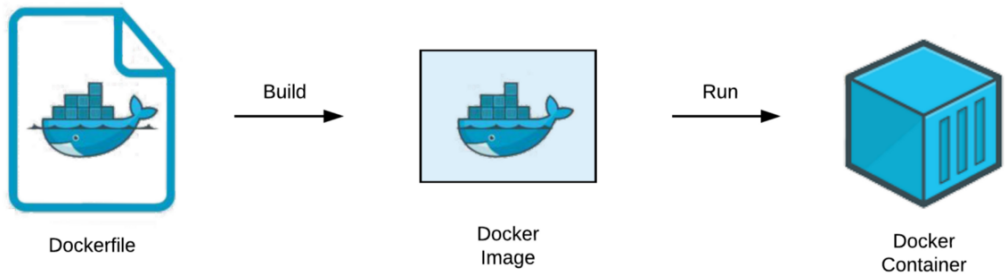
Container: donde se correrán los proyectos.

Images: los artefactos para empaquetar los contenedores y poder compartirlos.

Volumes: permite acceder al sistema de archivos de los contenedores.

Network: permite a los distintos contenedores comunicarse entre sí y con el mundo exterior.

Dockerfile: para construir imagenes personalizadas a partir de plantillas




```
# syntax=docker/dockerfile:1
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

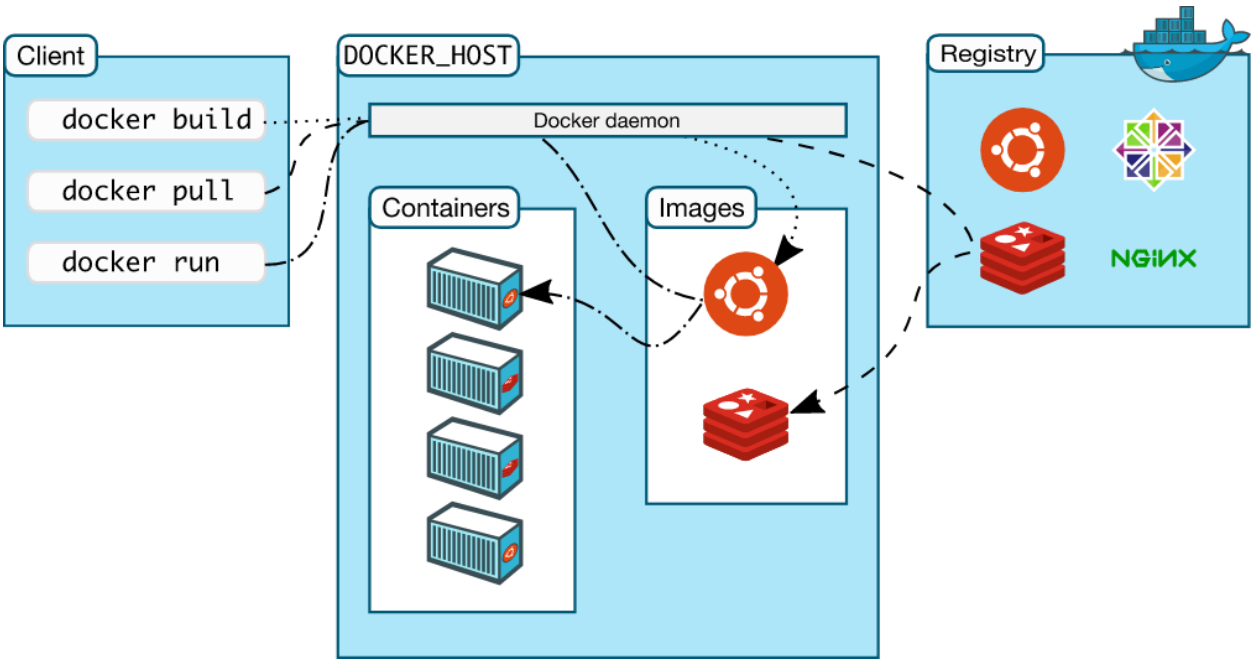
Docker Compose: compose es una herramienta para definir y ejecutar aplicaciones de varios contenedores, utiliza YAML para configurar los servicios y luego con un solo comando crea e inicia todos los servicios para la aplicación, ejm, Apache + PHP + MySQL

```
version: "3"

services:
  mariadb:
    image: mariadb:10.7
    container_name: mariadb
    expose:
      - 3306
    restart: always
    volumes_from:
      - mariadb-data
    environment:
      MYSQL_ROOT_PASSWORD: secret
      MYSQL_DATABASE: mydb
      MYSQL_USER: myuser
      MYSQL_PASSWORD: password
  mariadb-data:
    image: mariadb:10.7
    container_name: mariadb-data
```

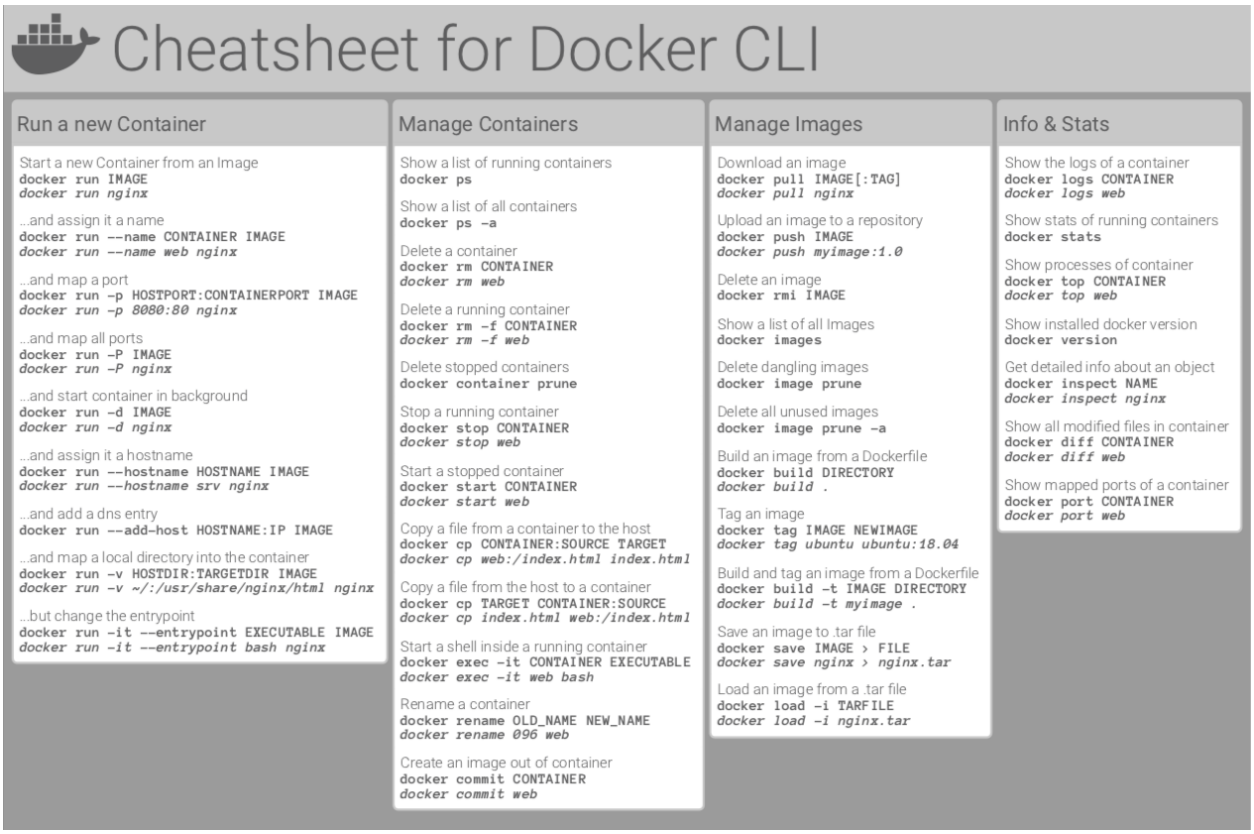
```
    volumes:
      - /var/lib/mysql
    command: "true"
  nginx:
    image: nginx:latest
    container_name: "nginx"
    build: ./nginx
    restart: always
    depends_on:
      - php
      - app-data
    ports:
      - "80:80"
      - "443:443"
    links:
      - php
    volumes_from:
      - app-data
  setup:
    build: ./php/
    volumes_from:
      - app-data
    command: "composer install -d /var/www/ --prefer-dist --no-progress"
  php:
    build: ./php/
    container_name: "php"
    restart: always
    depends_on:
      - setup
    expose:
      - 9000
    links:
      - mariadb
    volumes_from:
      - app-data
  app-data:
    image: php:7.4-fpm-alpine
    container_name: app-data
    volumes:
      - ./app:/var/www/
```

Docker Hub: es el repositorio proporcionado por Docker para buscar y compartir imágenes de contenedores con el equipo.



Docker Swarm: es una herramienta de orquestación de contenedores, lo que significa que permite al usuario administrar múltiples contenedores implementados en múltiples máquinas host.

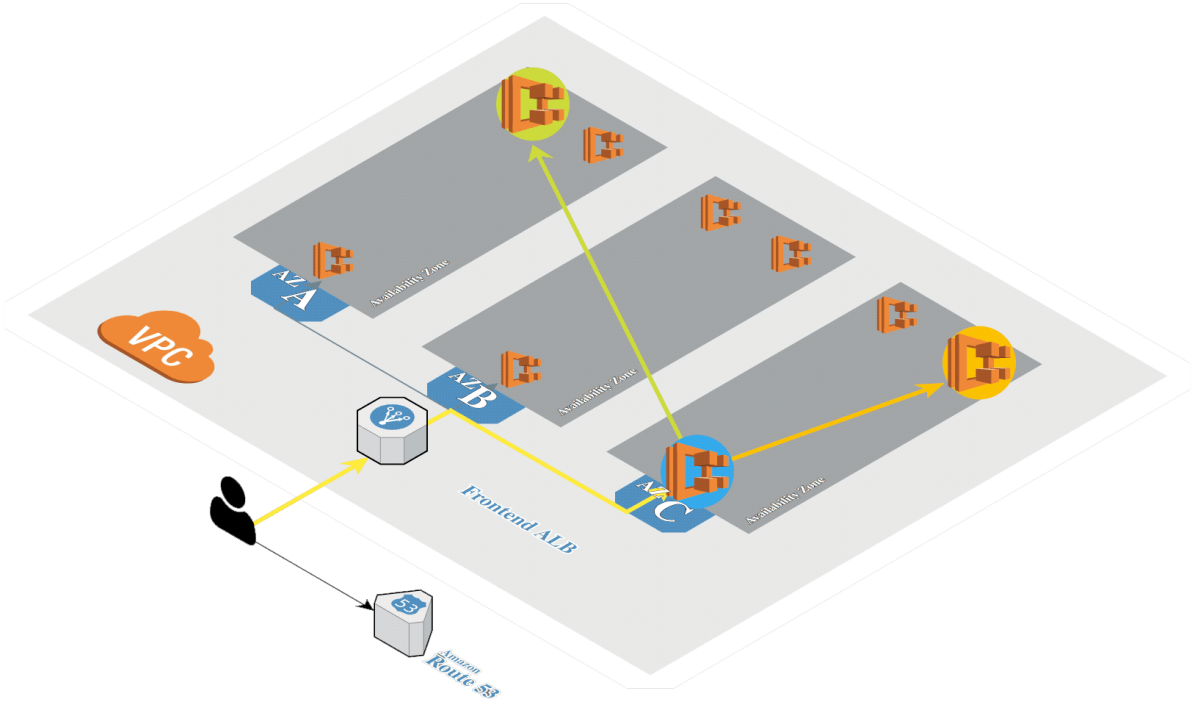
Hoja de trucos para Docker CLI



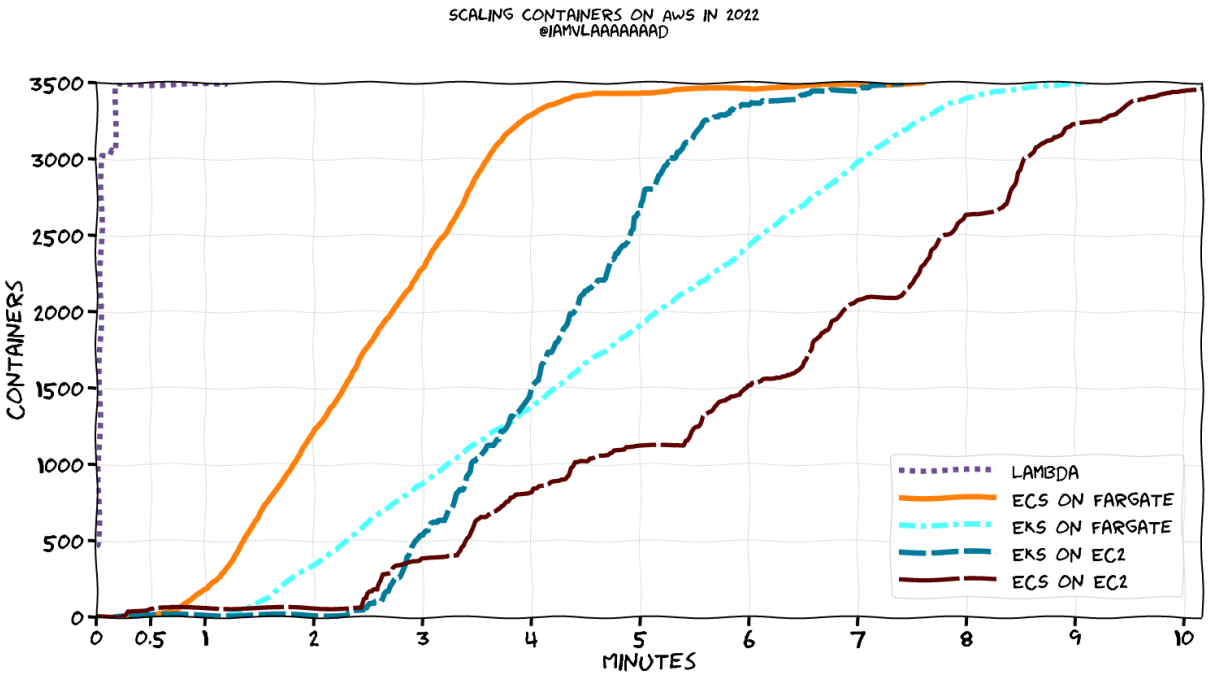
https://dockerlabs.collabnix.com/docker/cheatsheet/

ECS, Fargate, Compute Engine

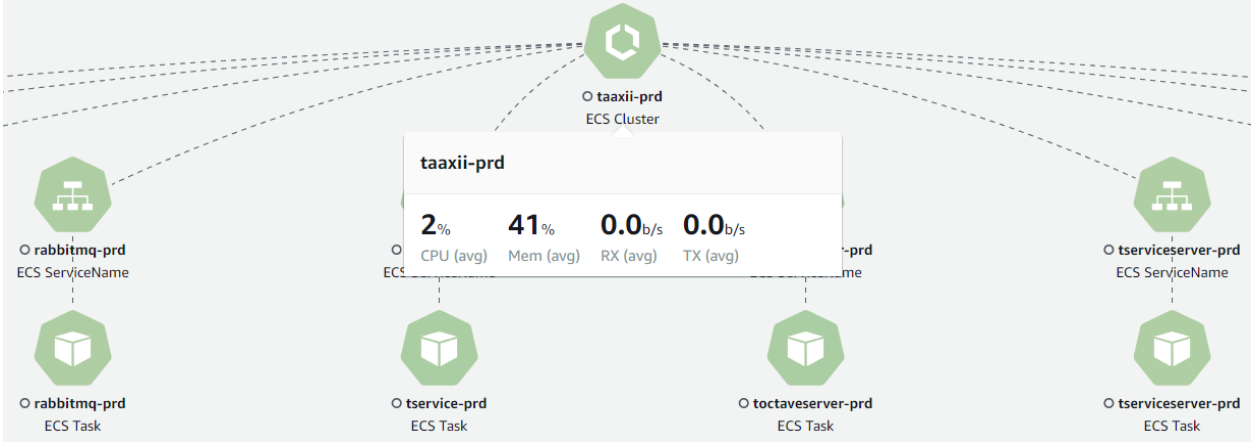
- el 80% de todas las aplicaciones en contenedores que se ejecutan en la nube se ejecutan en AWS
- 150% de crecimiento interanual de los servicios de contenedores de AWS
- más de 8 billiones de imágenes descargadas semanalmente con ECR

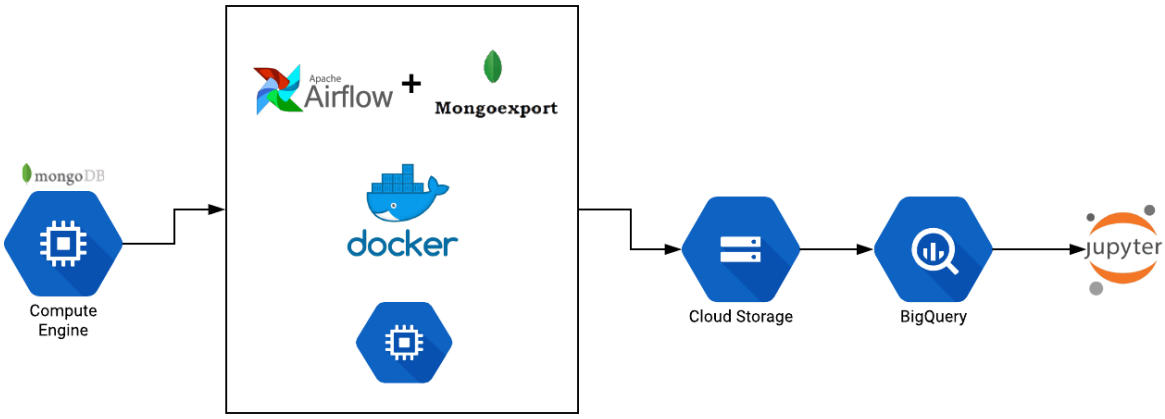


https://ecsworkshop.com/



https://www.vladionescu.me/posts/scaling-containers-on-aws-in-2022/





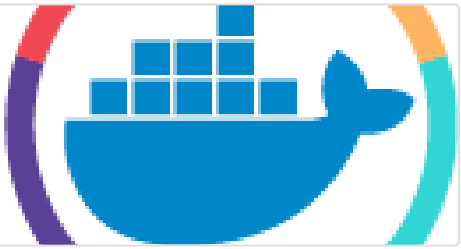
https://junjiejiang94.medium.com/get-started-with-airflow-google-cloud-platform-docker-a21c46e0f797

Referencias y links útiles

Docker Documentation

Home page for Docker's documentation

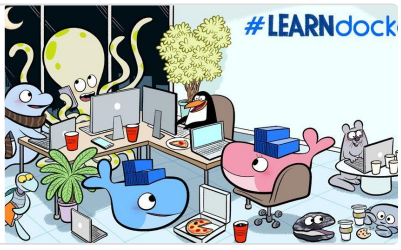
https://docs.docker.com/



🏗️ Docker Labs - The #1 Docker Tutorials and Free Resources for all Levels

A $0 Learning Platform for all Levels - from the ground up Over 500+ Highly Interactive Docker Tutorials and Guides Well tested on Docker Desktop and can be run on Browser (no Infrastructure required) DevOps Application Developers Solution

https://dockerlabs.collabnix.com/



Creating a container image for use on Amazon ECS

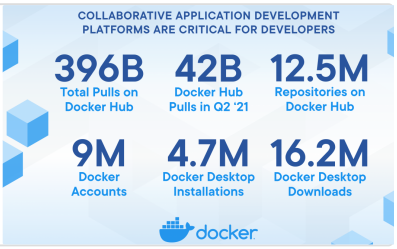
Amazon ECS uses Docker images in task definitions to launch containers. Docker is a technology that provides the tools for you to build, run, test, and deploy distributed applications in containers. Docker provides a walkthrough on deploying containers on Amazon ECS. For more information, see

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/create-container-image.html

Docker Index Shows Momentum in Developer Community Activity - Docker

The latest edition of the Docker Index is in, and it shows a continued growth in activity across the Docker community. The momentum we are seeing since the last Docker Index in February 2021 edition continues to grow. You'll recall that we started

https://www.docker.com/blog/docker-index-shows-surging-momentum-in-developer-community-activity-again/



Curso de Docker

Éste es un curso sobre los fundamentos de Docker, tanto para quienes no conocen esta tecnología y quieren aprender a usarla como para quienes la usan frecuentemente pero quieren profundizar en los detalles de cómo funciona en

https://platzi.com/cursos/docker/