
Entorno de Desarrollo

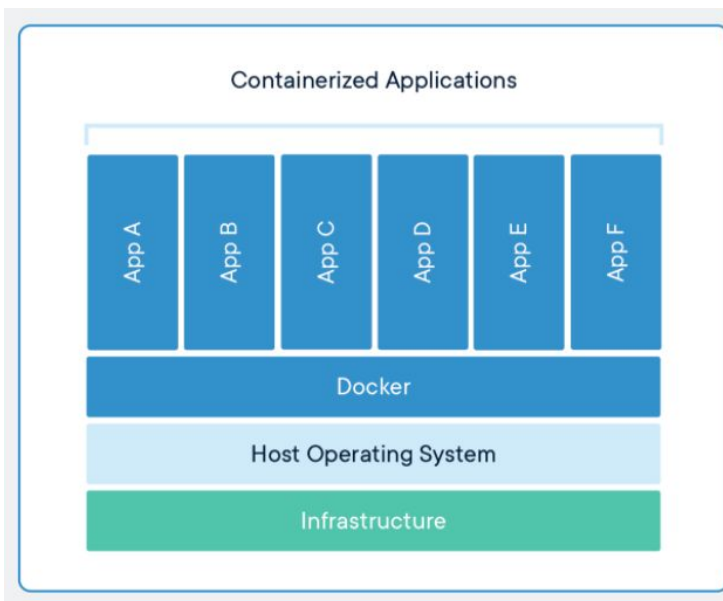
— **Cloud Computing** : Universidad Galileo —

Retos de los entornos de desarrollo

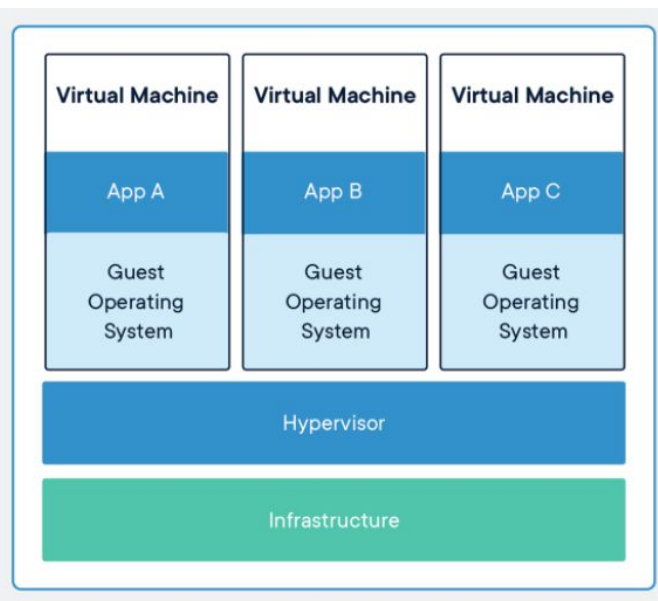
- Diferentes Sistemas Operativos de los Desarrolladores.
- Diferentes versiones en compiladores/intérpretes.
- Problemas en los pasos a Producción.
- Evitar el "en mi computadora funcionaba".

Posibles Soluciones

Docker

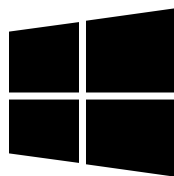


Máquinas Virtuales



Docker

- Las imágenes ocupan poco espacio.
- Portabilidad.
- Ampliamente adoptados por la comunidad y los Cloud providers.
- Facilidad de pasar escalar un servicio en producción de forma horizontal.
- Disponible para los principales Sistemas Operativos de Escritorio.



Terminos Basicos de Docker



Imagen: Es la Aplicación generada a partir del source code y el docker build.

Container: Es la Imagen en ejecución por medio del docker run.

Registry: Cloud Service que nos permite almacenar las imágenes de nuestra aplicación.

Cloud Docker Registries

- Docker Hub
- Amazon Elastic Container Registry
- Google Container Registry
- Azure Container Registry

¿Como empezamos con Docker?

Demo con un App de Angular

Demo

1. Definiendo y usando una Imagen para desarrollo.
2. Definiendo una imagen para producción
3. Subiendo la imagen de producción a Docker Hub

Dockerfile Desarrollo

 Dockerfile X

 Dockerfile

```
1  FROM node:12-alpine
2  WORKDIR /app
3
4  EXPOSE 4200
5  ENV PATH=/app/node_modules/.bin:$PATH
6
7  COPY package*.json /app/
8  RUN npm install
9
10 CMD ng serve --host 0.0.0.0
```

Dockerfile Desarrollo

```
1 FROM node:12-alpine
```

Indica que Imagen se utilizará de base para construir la nueva imagen.

```
2 WORKDIR /app
```

Establece el directorio en el que estaremos trabajando y afectará a la ejecución de otros comandos.

```
4 EXPOSE 4200
```

Indica con qué puertos trabaja nuestra aplicación y que deben ser expuestos.

```
5 ENV PATH=/app/node_modules/.bin:$PATH
```

Modifica la variable de entorno PATH

Dockerfile Desarrollo

```
7 COPY package*.json /app/
```

Copiar archivos a la imagen

```
8 RUN npm install
```

Ejecutar el comando para instalar los paquetes requeridos por la aplicación.

```
10 CMD ng serve --host 0.0.0.0
```

Establece que comando se ejecutara al ejecutar la imagen

Creando y ejecutando la imagen

```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)
| => docker build . -t demo-docker:latest
```

```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)
| => docker image list
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
demo-docker	latest	9c6a321e1fb3	50 minutes ago	599MB

```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)
| => docker run -v ~/git/demo-docker:/app -v /app/node_modules --name demo -d demo-docker:latest
```

```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)
```

```
| => docker container list
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
fa1fc08b4bc0	demo-docker:latest	"docker-entrypoint.s..."	11 minutes ago	Up 11 minutes	4200/tcp	demo

Docker Compose

 docker-compose.yml ✕

 docker-compose.yml

```
1  version: "3.6"
2  services:
3    frontend:
4      build: .
5      ports:
6        - "4200:4200"
7      volumes:
8        - "/app/node_modules"
9        - ".: /app"
```

```
| ~/git/demo-docker @ Diego
| => docker-compose up
```

```
| ~/git/demo-docker @ Diego
| => docker-compose down
```

Dockerfile para producción

 Dockerfile.prod ×

 Dockerfile.prod

```
1 FROM node:12-alpine as build
2 WORKDIR /app
3 ENV PATH=/app/node_modules/.bin:$PATH
4 COPY . /app/
5 RUN npm install
6 RUN ng build --prod --outputPath=./dist/build
7
8 FROM nginx:1.19-alpine
9 COPY --from=build /app/dist/build /usr/share/nginx/html
```

```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)
```

```
| => docker build -f Dockerfile.prod -t "demo-docker-prod:latest" .
```

Subir la imagen a Docker Hub

```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)  
[| => docker tag demo-docker-prod:latest diegocdl/demo-docker-prod:latest ]
```

```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)  
[| => docker login  
Login with your Docker ID to push and pull images from Docker Hub. If you don't have  
a Docker ID, head over to https://hub.docker.com to create one.  
Username: diegocdl  
[Password:  
Login Succeeded ]
```

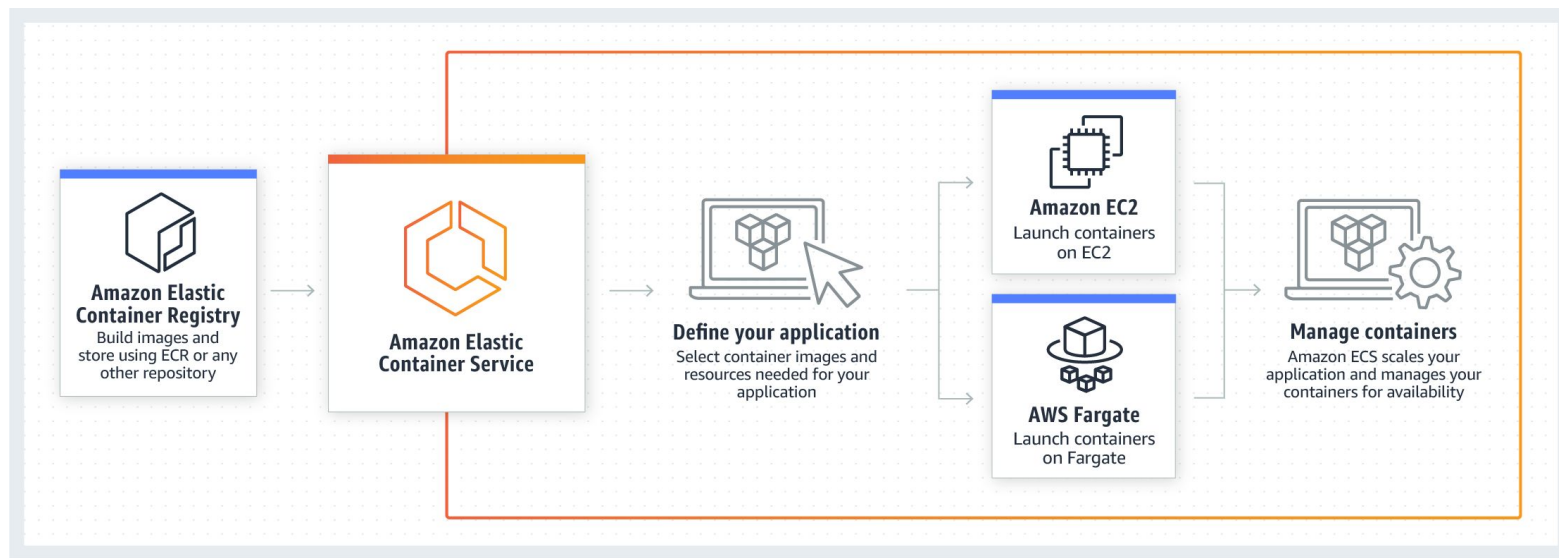
```
| ~/git/demo-docker @ Diegos-MacBook-Pro (diegocdl)  
[| => docker push diegocdl/demo-docker-prod ]
```

¿Y Como pasamos a producción la Imagen?

Cloud Services

- Amazon Elastic Container Service
- Google Cloud Run
- Azure Container Instances

AWS Elastic Container Service



AWS Cloud9

AWS Cloud 9

- Puede correr sobre un EC2 o cualquier servidor con SSH Server
- Capacidad de compartir el entorno de desarrollo para hacer Pair Programming.
- Buena integración con otros Servicios de AWS como Lambda