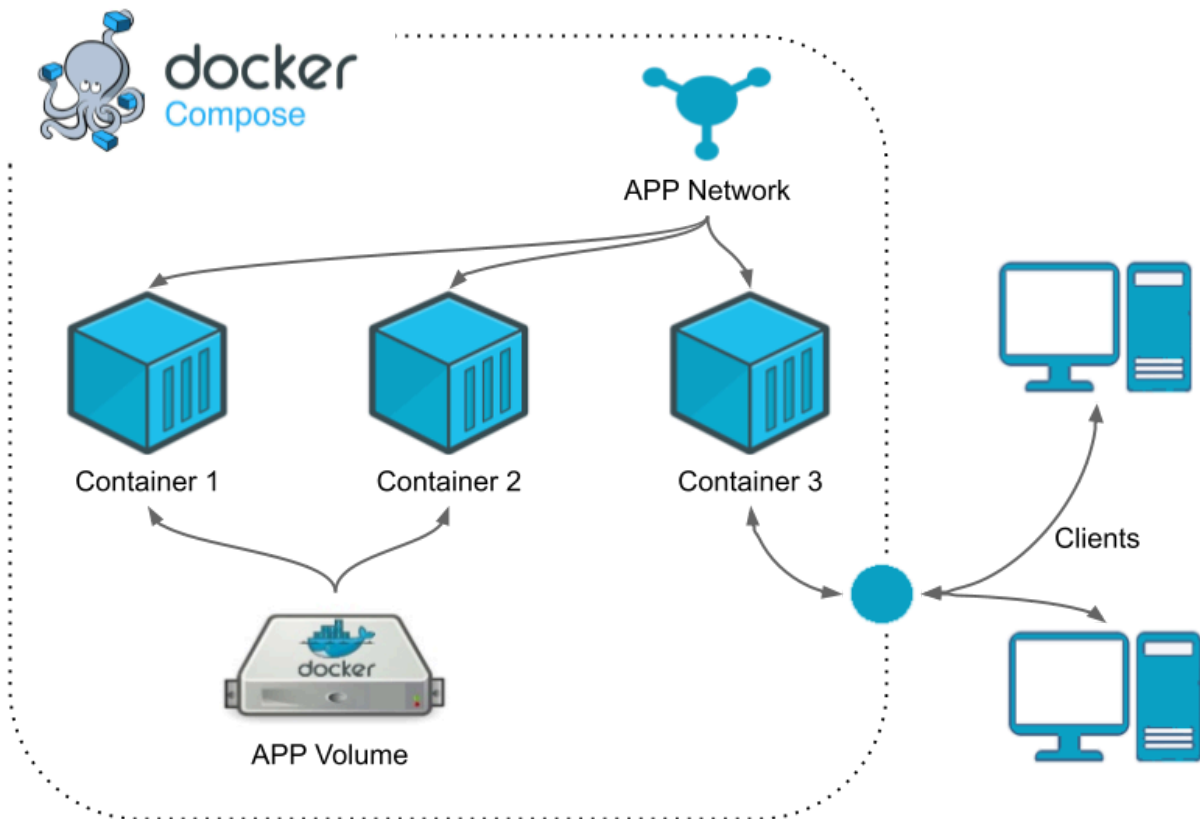
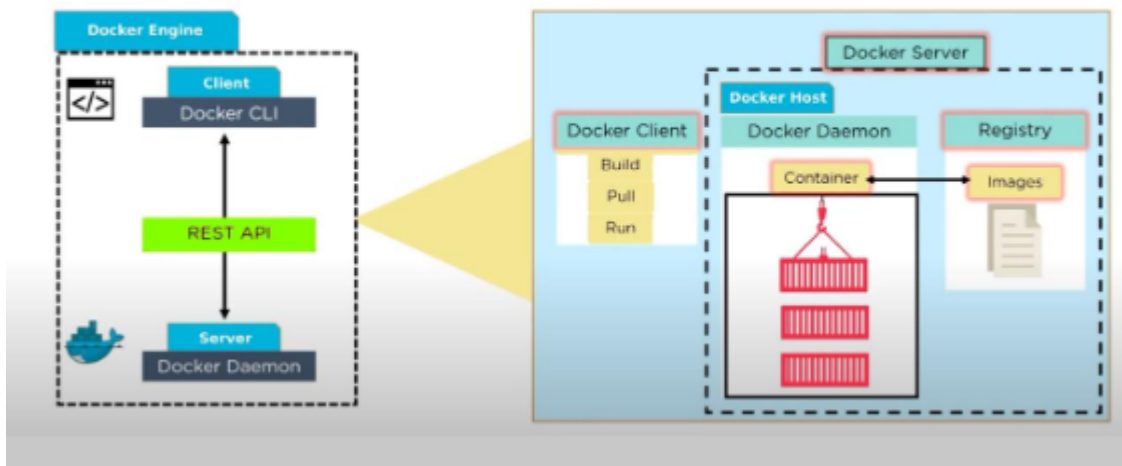


Implementación con Docker



Components of Docker



MARIO CAMACHO PRIETO
2º ASIR / CAJA MÁGICA
ARQUITECTURA EN LA NUBE

1. Introducción y Arquitectura (Página 1)

- Diagrama de Implementación (Docker Compose) : Ilustra un entorno con múltiples contenedores (Container 1, 2, 3) conectados a una red (APP Network) y compartiendo un volumen (APP Volume).
- Diagrama de Componentes de Docker : Muestra la arquitectura del Docker Engine , detallando la interacción entre el cliente (Docker CLI) , la REST API y el servidor (Docker Daemon). También describe el Docker Host , que gestiona el Daemon , el Registro (Registry) y las imágenes (Images).

2. Instalación y Verificación (Página 2)

- Comprobar la versión de Docker : Se utiliza el comando `docker --version` para verificar la instalación y la versión (ej. 24.0.5).
- Instalar Docker (en Ubuntu/WSL) : Se muestra el comando `sudo apt update && sudo apt install -y docker.io` para la instalación del paquete `docker.io`.
- Ejecutar imagen de prueba 'hello-world' : Se introduce el comando `sudo docker run hello-world` como prueba básica.

3. Comandos Básicos y Gestión (Página 3)

- Resultado de 'hello-world' : El sistema descarga la imagen y muestra el mensaje de confirmación "Hello from Docker!".
- Ver contenedores (activos y detenidos) : Se usa `sudo docker ps -a` para listar todos los contenedores, mostrando el contenedor 'hello-world' con estado "Exited".
- Ver imágenes locales : Se usa `sudo docker images` para listar las imágenes descargadas, como 'hello-world'.

4. Creación y Ejecución de Imagen Personalizada (Página 4)

- Construir imagen personalizada (Apache + PHP) : Se utiliza `sudo docker build -t mi_apache_php .` para crear una imagen.
- Proceso de construcción: Se muestran los pasos, como Step 1/5 FROM ubuntu:20.04 y Step 2/5 RUN apt-get install... , culminando en Successfully built....
- Ejecutar el contenedor en background : Se lanza el contenedor con `sudo docker run -d -p 8080:80 --name web_test mi_apache_php` , mapeando el puerto 8080 del host al 80 del contenedor.

5. Verificación del Contenedor Personalizado (Página 5)

- Entrar al contenedor : Se accede al contenedor en ejecución con `sudo docker exec -it web_test bash`.
- Listar archivos web: Dentro del contenedor, se ejecuta `ls /var/www/html` para verificar la presencia de `index.html` e `info.php`.
- Comprobar la página PHP desde el host : Se usa `curl http://localhost:8080/info.php` para probar la aplicación.
- Resultado del Curl: Se recibe una respuesta HTML que muestra "PHP Version 7.4.3" e "Información desde el contenedor Docker con Apache + PHP".

6. Ejemplo de Dockerfile (Páginas 5-6)

- Resumen del Dockerfile : Se presenta el código fuente mínimo del Dockerfile usado para construir la imagen `mi_apache_php`.
- Instrucciones del Dockerfile:
 - `FROM ubuntu:20.04`
 - `RUN apt-get update && apt-get install -y apache2 php libapache2-mod-php`
 - `COPY ./html /var/www/html`
 - `EXPOSE 80`
 - `CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]`

Comprobar versión de Docker

Comando: `docker --version`

Verifica que Docker está instalado y la versión disponible.

```
Windows PowerShell

PS C:\Users\Mario> docker --version
Docker version 24.0.5, build 12345abcde
```

Instalar Docker (si no está instalado)

Comando: `sudo apt install -y docker.io`

Instala el paquete `docker.io` en Ubuntu/WSL.

```
Windows PowerShell

PS C:\Users\Mario> sudo apt update && sudo apt install -y docker.io
Reading package lists... Done
The following NEW packages will be installed: docker.io
Setting up docker.io (24.0.5-0ubuntu1) ...
Processing triggers for systemd...
```

Ejecutar imagen de prueba 'hello-world'

Comando: `sudo docker run hello-world`

Prueba básica para comprobar que Docker puede descargar imágenes y ejecutar contenedores.

```
Windows PowerShell

PS C:\Users\Mario> sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
Digest: sha256:abcdef1234567890
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Ver contenedores (incluye los detenidos)

Comando: `sudo docker ps -a`

Lista todos los contenedores, activos y detenidos, para verificar el estado de ejecución.

```
Windows PowerShell

PS C:\Users\Mario> sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
alb2c3d4e5f6   hello-world    "/hello"                2 minutes ago Exited (0) 1 minute
ago            nostalgic_morse
```

Ver imágenes locales

Comando: `sudo docker images`

Muestra las imágenes descargadas en el host Docker.

```
Windows PowerShell

PS C:\Users\Mario> sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    e7d4f0c0e5e6  2 weeks ago   13.3kB
```

Construir imagen personalizada (Apache + PHP)

Comando: `sudo docker build -t mi_apache_php .`

Construye una imagen Docker a partir de un Dockerfile que instala Apache y PHP.

```
Windows PowerShell

PS C:\Users\Mario> sudo docker build -t mi_apache_php .
Sending build context to Docker daemon 3.072kB
Step 1/5 : FROM ubuntu:20.04
--> 1d622ef86b13
Step 2/5 : RUN apt-get update && apt-get install -y apache2 php libapache2-mod-php
--> Running in 5a6b7c8d9e0f
Processing triggers for systemd ...
Successfully built abcdef123456
Successfully tagged mi_apache_php:latest
```

Ejecutar el contenedor en background y mapear puertos

Comando: `sudo docker run -d -p 8080:80 --name web_test mi_apache_php`

Lanza el contenedor en segundo plano exponiendo el puerto 80 del contenedor en el puerto 8080 del host.

```
Windows PowerShell

PS C:\Users\Mario> sudo docker run -d -p 8080:80 --name web_test mi_apache_php
b7c9d8e6f5a4c3b2d1e0f9a8b7c6d5e4f3a2b1c0d
```

Entrar al contenedor y listar archivos web

Comando: `sudo docker exec -it web_test bash` `ls /var/www/html`

Se accede al contenedor para comprobar que los archivos web (index.php, info.php) están presentes.

```
Windows PowerShell

PS C:\Users\Mario> sudo docker exec -it web_test bash
root@b7c9d8e6f5a4:/# ls /var/www/html
index.html  info.php
```

Comprobar la página PHP desde el host

Comando: `curl http://localhost:8080/info.php`

Desde el host se comprueba que la aplicación dentro del contenedor sirve correctamente la página PHP.

```
Windows PowerShell

PS C:\Users\Mario> curl http://localhost:8080/info.php
<!DOCTYPE html>
<html>
<body>
<h1>PHP Version 7.4.3</h1>
<p>Información desde el contenedor Docker con Apache + PHP</p>
</body>
</html>
```

Ejemplo de Dockerfile (resumen)

Ejemplo mínimo de Dockerfile usado para crear la imagen `mi_apache_php`:

```
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y apache2 php libapache2-mod-php
COPY ./html /var/www/html
EXPOSE 80
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```