

Dockerfile Avanzado

Arturo Silvelo

Try New Roads

Ejemplo 1: Mínimo

- Contenido del `Dockerfile`

```
FROM alpine:3.20  
CMD ["echo", "Hola Docker!"]
```

- Construcción de la imagen

```
docker build -t app-minimo .
```

- Ejecución de la imagen

```
docker run --rm app-minimo
```

El `build` debe ejecutarse donde se encuentra el fichero `Dockerfile`.

El `context` `.` también debe ejecutarse con la ruta correcta.

Ejemplo 2: Multi-stage

Este ejemplo muestra cómo crear una imagen Docker optimizada usando `multi-stage`. Se parte de una aplicación sencilla en `Node` que crea un servidor

- Dockerfile.before: Imagen sin multi-stage

```
FROM node:20
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
CMD ["node", "app.js"]
```

- Creación:

```
docker build -f Dockerfile.before -t app-simple .
```

- Ejecución y comprobación:

```
docker run --rm --init -p 3000:3000 app-simple
curl http://localhost:3000
```

- Dockerfile.after: Imagen con multi-stage.

```
FROM node:20 AS build
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .

FROM node:20 AS production
WORKDIR /app
COPY --from=build /app/app.js .
COPY --from=build /app/package.json .
RUN npm install --production
CMD ["node", "app.js"]
```

- Creación:

```
docker build -f Dockerfile.after -t app-multi-stage .
```

- Ejecución y comprobación:

```
docker run --rm --init -p 3000:3000 app-multi-stage  
curl http://localhost:3000
```


