

docker-compose

variables

- maneras de definir variables para usar en el fichero **docker-compose.yml**
- están ordenadas de menos prioritaria a más prioritaria:
 1. se puede definir un fichero **.env** con variables para usar en el **docker-compose**: VARIABLE
 2. se pueden usar variables definidas en el shell dentro del **docker-compose** : **\${VARIABLE}**
 3. se pueden pasar variables desde la línea de comandos: `docker run -e <variable>=<valor>`
 4. se pueden especificar otros archivos de variables y pasarlas desde la línea de comandos: `docker run -env-file <fichero>`

`.env`

```
# variables de entorno preceden a estas
# variables pasadas por línea de comando preceden a estas
# docker run -e <variable>=<valor>
# se pueden especificar otras variables en otros archivos
# docker run -env-file <fichero>

MYSQL_ROOT_PASSWORD=toor
MYSQL_DATABASE=xxxx
MYSQL_USER=xxxx
MYSQL_PASSWORD=xxxx
MYSQL_STORAGE_PATH=/home/mate/Docker/.storage/mysql/xxxx
```

- para pasar variables a los contenedores, se puede:
 - usar la sección **environment**, donde definimos la variable (y podemos setear o hace que recoga el valor del shell)
 - usar la sección **env_file**, donde carga los valores desde un fichero externo
 - pasarlo por línea de comando con el parámetro **-e <VAR>[=<VALOR>]** y al igual que la sección **environment**, podemos setearlo directamente o dejar que coja el valor que haya en el shell
- para ver el resultado de aplicar varias configuraciones de variables y que se obtiene el resultado que se desea, podemos usar: `docker-compose config` y nos mostrará el fichero con el resultado final una vez hecha la conversión de las diferentes variables
- <https://docs.docker.com/compose/environment-variables/>

sftp

- mapea en la carpeta home del usuario especificado (no se puede sustituir el \$HOME)
- permite mapear varios usuarios
- permite encriptación de contraseñas
- permite certificados

`docker-compose.yml`

```
sftp:
  image: atmoz/sftp
  volumes:
    - ${PWD}/upload:/home/foo/upload
  ports:
```

```
- "2222:22"  
command: foo:user123:1001
```

portainer

[docker-compose.yml](#)

```
version: '2'  
  
services:  
  portainer:  
    image: portainer/portainer  
    restart: always  
    container_name: "portainer-app"  
#    command: --templates http://templates/templates.json  
    ports:  
      - "9000:9000"  
    networks:  
      - local  
    volumes:  
      - /var/run/docker.sock:/var/run/docker.sock  
      - ./data:/data  
  
networks:  
  local:  
    driver: bridge
```

mysql

[docker-compose.yml](#)

```
version: '3.6'  
  
services:  
  db:  
    image: mysql:5.7  
    volumes:  
      - db_data:/var/lib/mysql  
    environment:  
      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}  
      MYSQL_DATABASE: ${MYSQL_DATABASE}  
      MYSQL_USER: ${MYSQL_USER}  
      MYSQL_PASSWORD: ${MYSQL_PASSWORD}  
    networks:  
      mi-network:  
        ipv4_address: 172.22.0.10  
    ports:  
      - 3306:3306
```

```
volumes:
  db_data:
    driver: local
    driver_opts:
      type: 'none'
      o: 'bind'
      device: '${MYSQL_STORAGE_PATH}'

networks:
  # docker network create --attachable --subnet 172.22.0.0/24 mi-network
  mi-network:
    external: true
```

From:
<https://miguelangel.torresegea.es/wiki/> - **miguel angel torres egea**

Permanent link:
<https://miguelangel.torresegea.es/wiki/tech:docker:docker-compose?rev=1568277344>

Last update: **12/09/2019 01:35**

