

Docker Alapok

Mi az a Docker?

A Docker egy platform alkalmazások fejlesztésére, szállítására és futtatására könnyűsúlyú konténerekben. A konténerek egy csomagban tartalmazzák a kódot és a függőségeket, így biztosítva a környezetek közötti konzisztenciát.

Főbb fogalmak

- **Image (Kép):** Egy csak olvasható sablon, amely utasításokat tartalmaz egy konténer létrehozásához.
- **Container (Konténer):** Egy image futtatható példánya.
- **Dockerfile:** Egy szöveges fájl, amely utasításokat tartalmaz egy Docker image elkészítéséhez.

Gyakori alap parancsok

```
# Docker verzió ellenőrzése
docker --version
```

```
# Futó konténerek listázása
docker ps
```

```
# Minden konténer listázása (a leállítottakat is)
docker ps -a
```

```
# Konténer létrehozása és egyben indítása
docker run <image_name>
```

```
# Konténer indítása interaktív módban (megnyitja a konténer terminálját).
# Itt az -i és a -t kapcsoló kerül együttes felhasználásra,
# ahol az -i adja az interaktív módot, hogy írni tudj a terminálba,
# a -t pedig a megszokott terminálélményt; kurzormozgatás,
# tördelés, prompt, stb.
docker run -it <image_name>
```

```
# Konténer indítása megadott image-el úgy, hogy leállítás
# vagy a parancs lefutása után törlődjön is
docker run --rm <image_name>
```

```
# Futtatott konténer leállítása
docker stop <container_id>
```

```
# Konténer törlése
docker rm <container_id>
```

```
# Konténer leállítása és törlése
docker rm -f <container_id>
```

Dockerfile alapok

A Dockerfile egy szöveges fájl, amely tartalmazza azokat az utasításokat, amelyek alapján a Docker egy image-et épít. Olyan, mint egy recept: megmondja, hogy milyen alaptól induljunk, mit kell bemásolni, telepíteni, beállítani.

Dockerfile felépítése

Egy, már órán elkészített példával élve:

```
FROM ubuntu:latest

RUN apt update && apt install -y cowsay

COPY cat.cow /usr/share/cowsay/cows

ENTRYPOINT ["/usr/games/cowsay", "-f", "cat"]

CMD ["Miauu"]
```

```
FROM ubuntu:latest
```

Az első sor az alap image-t adja meg. Innen indulunk, konténerünk alapja a legfrissebb ubuntu image lesz (erre utal a latest tag)

```
RUN apt update && apt install -y cowsay
```

A RUN paranccsal megadhatunk még build időben futtatandó parancsokat; ez jelen esetben kettő parancsot jelent, az apt update-t és az apt install-t. A && operátorral mondhatjuk meg azt, hogy két parancsot is futtatni szeretnénk, természetesen egymás után. A két parancs közül az utóbbi csak akkor fog lefutni, ha az előző sikeresen lefutott. A -y kapcsoló megmondja, hogy minden felajánlott lehetőségre, telepítési kérdésre igennel szeretnénk válaszolni.

```
COPY cat.cow /usr/share/cowsay/cows
```

A host gépen lévő cat.cow fájlt bemásolja a konténerbe, a cowsay program állatfájlainak könyvtárába, így a cowsay tud majd egy macskás ASCII figurát is használni. A copy után az első paraméter a HONNAN, a második a HOVÁ.

```
ENTRYPOINT ["/usr/games/cowsay", "-f", "cat"]
```

A belépési pont megmondja, hogy a konténer mindig a cowsay programmal induljon. A -f paraméterrel mondjuk meg neki, hogy mindig a macskás ASCII artot szeretnénk használni. Az ENTRYPOINT fix, ez futás közben nem fog felülíródni.

```
CMD ["Miauu"]
```

Alapértelmezett paraméter. Amennyiben a felhasználó megad egy saját paramétert, úgy ez felül fog íródni a program futtatásakor. Pl.:

```
docker run saját_catsay "Hello világ!"
```

Ilyenkor a macskánk nem azt fogja mondani, hogy Miauu, hanem, hogy Hello világ!.

Dockerfile parancsok

A `docker build` parancsával tudunk a Dockerfileunkból image-t készíteni a megfelelő paraméterezéssel. Nézzünk egy példát.

```
docker build -f mappa/xy.Dockerfile -t saját_image:1.0 .
```

Itt a `-f` kapcsoló adja meg a fájlt, ami jelen példánkban egy mappában van. Alapértelmezetten a Dockerfile nevű fájlt keresi.

A `-t` kapcsolóval megadhatjuk a kimeneti fájlt, itt a `saját_image` nevű imagenek adtunk egy verziókódot is, tagként.

A `.` a végén az aktuális könyvtárból fog építeni.

Ezt lefuttatva, amennyiben hiba nélkül végig fut, lesz egy használható image-ünk, amit indíthatunk.

Docker container futtatása

Egy, Dockerfileból készült imaget futtathatunk a megszokott módon, `docker run`-al. Jelen esetben a `-d` kapcsolót használjuk, ami a detached módot jelenti. Ilyenkor a konténer a háttérben fog elindulni, a terminálunkat visszakapjuk, a konténer azonosítójával egyetemben.

```
docker run -d saját_image:1.0
```