

Dr. Hajnal Éva: Haladó
programozás

HALADÓ PROGRAMOZÁS

Docker

Konténer, konténerizált alkalmazások

A szoftvertervezésben a konténerezés

operációs rendszer szintű virtualizációt
vagy

alkalmazásszintű **virtualizációt** jelent
több hálózati erőforráson keresztül,

szoftveralkalmazások a konténereknek
nevezett elszigetelt felhasználói terekben
futhatnak bármilyen felhőben vagy nem
felhős környezetben, típustól és gyártótól
függetlenül.

Konténerek: Docker, Kubernetes, MS
Azúre,

A Docker egy olyan eszköz, amely lehetővé teszi a fejlesztők, rendszergazdák stb. számára, hogy könnyen telepítsék alkalmazásaikat egy homokozóban (ún. konténerek), hogy a gazdagép operációs rendszeren, azaz Linuxon futhassanak.

A Docker fő előnye, hogy lehetővé teszi a felhasználók számára, hogy egy alkalmazást az összes függőséggel együtt szabványosított egységbe csomagoljanak szoftverfejlesztéshez.

A virtuális gépektől eltérően a konténerek nem rendelkeznek nagy többletterheléssel, így lehetővé teszik az alapul szolgáló rendszer és erőforrások hatékonyabb felhasználását.

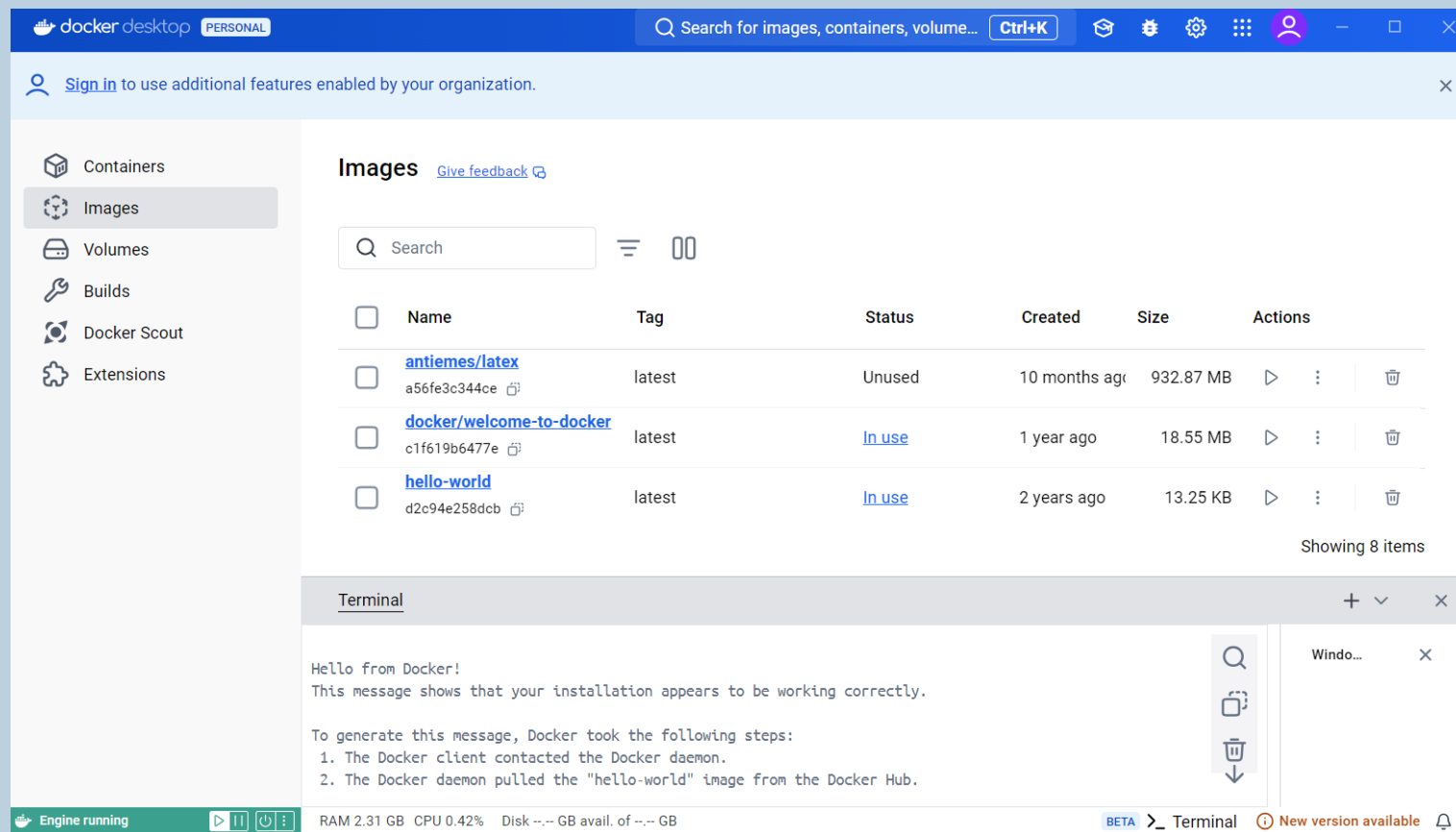
Mi a docker?

Konténerizáció vs virtuális gép

- A konténerek a gazdagép operációs rendszer alacsony szintű mechanikájának kihasználásával a konténerek a
- számítási teljesítmény töredékével biztosítják
- a virtuális gépek izolálásának legnagyobb részét.
- A konténerek csomagolási mechanizmust kínálnak, amelyben az alkalmazások elvonatkoztatathatók attól a környezettől, amelyben ténylegesen futnak.
- Lehetővé teszi a konténer alapú alkalmazások egyszerű telepítését, függetlenül attól, hogy a célkörnyezet privát adatközpont, nyilvános felhő vagy akár egy fejlesztő személyes laptopja.
- Működési szempontból a hordozhatóságon kívül a konténerek részletesebb ellenőrzést is biztosítanak az erőforrások felett, így az infrastruktúra jobb hatékonyságot biztosít, ami a számítási erőforrások jobb kihasználását eredményezheti.
- Komoly szoftver cégek a fejlesztéseket konténerizált környezetben valósítják meg.

Gyakorlati alkalmazás

- Telepítés a tutorialok szerint
- Előfeltételek:
 - Git scm
 - Linux (pl.UBUNTU) vagy WSL (Linux Windows Alrendszer)
 - Docker alkalmazás



Docker teszt

- Docker run hello-world

Docker parancsok

- **Docker run** (kisbetű!)- konténer futtatása
- **Docker ps** –futó konténerek listázása
- **Docker ps -a** –összes konténer listája
- **Docker run –help** paracs tutorial
- **Docker rm xxxx** konténer törlése az id alapján
- **Docker rmi** image törlése
- **Docker image ls** image-ek listázása
- **docker build -t DOCKER_USERNAME/APPNAME** konténerizált alkalmazás készítése
- **docker push DOCKER_USERNAME/getting-started-todo-app** konténer feltöltése
- **docker compose watch** –alkalmazás konténerizációja

Fontos fogalmak

- Image (kép): Alkalmazásunk terve, azok a file-ok, amik szükségesek a futtatáshoz, amely a konténerek alapját képezik. **Az image-ek nem változtathatók!**
- Konténer(tároló): Docker-képekből készült, és futtatja a tényleges alkalmazást. Docker run paranccsal hozunk létre egy tárolót. Ez tartalmazza a forrásfile-okat.
- Docker démon: A gazdagépen futó háttérszolgáltatás, amely felügyeli a Docker-tárolók építését, futtatását és elosztását. A démon az operációs rendszerben futó folyamat, amellyel az ügyfelek beszélnek.
- Docker kliens: A parancssori eszköz, amely lehetővé teszi a felhasználó számára a démonnal való interakciót.
- Docker Hub: A Docker képek nyilvántartása. A rendszerleíró adatbázist az összes elérhető Docker-képfájl könyvtárának tekintheti. Igény esetén saját Docker-nyilvántartásokat is tárolhat, és képeket lehívhat.
 - Publikus image-ek
 - Privát image-ek

Videok

- <https://youtu.be/D0SDBrS3t9I>
- <https://docs.docker.com/get-started/introduction/build-and-push-first-image/>
- <https://hub.docker.com/search?badges=official>

Köszönöm a figyelmet!