**Java alkalmazások Dockerrel**

Ha az **alkalmazásunkat Docker konténerben** szeretnénk futtatni, akkor **először létre kell hozni a Docker image-et** -> ezt standard Dockeres eszközökkel is meg tudjuk tenni; ehhez **először létre kell hozni egy Dockerfile file-t,** amiben **le kell írni, hogy hogy állítjuk össze a Docker image-t;** sorai:

**FROM** – **már milyen létező imageből indulunk ki**? nem kell mindig elölről kezdeni, hanem **támaszkodhatunk már meglévő imagekre** -> az ’adoptopenjdk’ egy ingyenes jdk implementáció, disztribúció és a Docker Hubon vannak fent image-ek, amelyek tartalmaznak egy OP rendszert és a hozzá tartozó JDK-t is, jelen esetben egy 13as jdk-t fogunk használni (egy olyan imageből indulunk ki, amin már egy 13as jdk rajta van)

**RUN** - majd létrehozunk egy könyvtára**t mkdir opt/app**

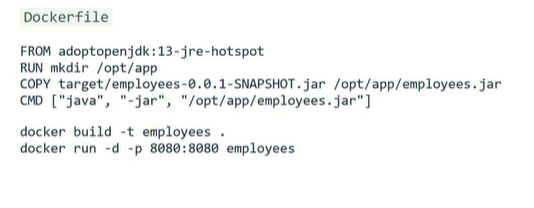
**COPY** **– jar állomány átmásolása -> a konténeren belülre**

**CMD** – kiadjuk a java virtuális gép indítása parancsot

ha kiadjuk még a **docker build -t employees . –** parancsot (pont = adott könyvtárban -> jelentése: olvassa fel a docker file-t ill. az összes többi filet is megkapja a docker, hogy tudja ezeket a fileokat bemásolni az imagebe; a copy parancs valójában ezért fog működni; -t kapcsoló: megmondjuk, hogy az imagenek a neve (=tag) employees legyen

ezzel létrejön egy image employees névvel, és benne van egy .jar állományunk; ha el akarjuk indítani az imaget, vagyis létre akarunk hozni egy konténert, a docker run parancsot kell kiadni; -d kapcsoló: visszakapujk a promtot; -p : a **hostunkra hozzuk ki a konténerenek a 808as portját;** utolsó paraméter (employees) magának az imagenek a neve

**ekkor az adott image alapján elindul az adott konténer és a cmd parancs miatt a konténeren belül a java -jar parancs is, vagyis a konténeren belüli java alkalmazásunk is, ami a 8080-as porton várja a kéréseket; mivel a portot is kivezettük a -p kapcsolóval, ha localhost :8080-at címezzük meg, akkor az automatikusan továbbításra kerül a konténer 8080-as portjára**



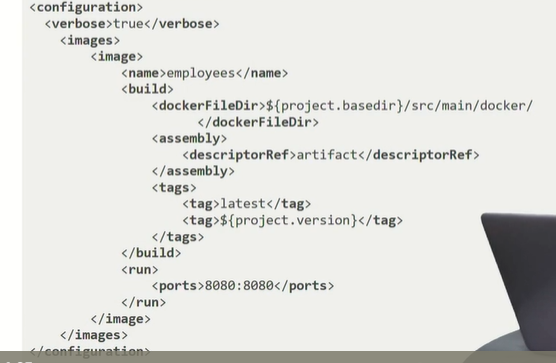
**Maven Plugin**

**lehetőség van a Docker Mavenből való vezérlésére is;** erre több Maven plugin is van, mi a **fabric**-ost nézzük

pom.xml függőség felvétele:

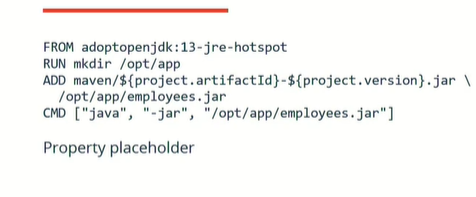


majd konfiguráció megadása:



**Dockerfile – Maven plugin esetén**

Hasonlít a már előbb megírthoz, annyi különbséggel, hogy itt működik a **property place holder**, a maven propertyjeit kicseréli ez a maven plugin a docker fileban és úgy futtatja a docker parancsokat; a generált artifakt nevére mavne propertykkel hivatkozunk, ami így ki lesz cserélve



**Parancsok**

**Ha Mavenen keresztül szeretnénk a Dockert irányítani, akkor:**

**mvn package docker: build** (=docker plugin build goalja; legyártja az adott imaget)

**mvn docker: start** (elindítja az imaget)

**mvn docker: stop** (ez törli is! minden egyes indításkor újat fog létrehozni; plusz kapcsoló ezt kiküszöböli)

**12Factor hivatkozás: Disposability**

*Ha Dockert szeretnénk használni, fontos, hogy tartsuk be:*

**gyorsan induljanak és álljanak is le az alkalmazásaink**

működjön a **’graceful shutdown’ –** ha **leállítjuk** az alkalmazást a **szabályos módon**, **tegyen rendet** maga után

**ne hagyjon maga után inkonzisztens adatokat** -> **ellentmondásos adat ne maradjon** az adatbázisban, filerendszerben

*batch folyamatoknál se legyen ellentmondásos* adat; ezek hosszú folyamatok, ezért különösen fontos; kétféle módszer erre:

*1.Használjunk tranzakciókat*: ha leállítjuk az alkalmazást graceful módon, akkor az **meg fogja várni a tranzakció befejezését**; ha kilőjük az adott konténert, akkor **a tranzakciónak a commit művelete** nem fog lefutni és így nem marad inkonzisztens adat

*2. Idempotencia*: akkor idempotens egy művelet, hogyha **2x lefuttatjuk mindkétszer ugyanazt az eredményt adja vissza**; ne legyen gond a kétszer egymás utáni futásból PUT

**Kérdések**

*Mik a lépései az alkalmazásunk Docker konténerben való futtatásának?*

*Mi az a Docker file és hogy áll össze?*

*Milyen paranccaal buildeljük le az imaget?*

*Konténer indításaor milyne kapcsolóvla kapjuk vissza apromptot?*

*Mitől indul el az elindított konténerben a java lakalmazásunk?*

*Miket kell beállítani a pomban a Docker Mavenből való kezeléséhez?*

*hogy néz ki ilyenkor a Dockerfile, miket használ?*

*Hogyan buildeljük le a zimage,t indítunk és állítunk le konténert Mavenből?*

*docker használata esetén milyen 12Factor hivatkozásokat tartsunk be?*