**Feltöltés Git repositoryba**

1. Git repository létrehozása a webes felületen egy névvel; ezt majd a saját lokális gépen kiadjuk a parancsokat:

git init

git add .

git commit „First commit”

gir remote add origin https//github.com/username/employees.git

git push -u origin master

**12Factor hivatkozás – one codebase, one application**

verziókezelésre ajánlások

egy repository = egy alkalmazás

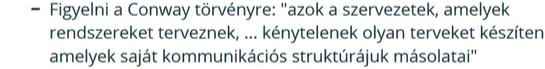
minden más, amit függőségként definiálunk, azt külön repoban kell tárolni

Milyen jelenségek utalhatnak ezen szabály megtörésére?

* **alkalmazás különböző moduljai különböző repokban**; a modulok szeparálását bár megkönnyíti, de a build folyamatot nagy mértékben megbonyolítja; inkább egy repoba!
* különböző külön futtatható alkalmazások, processzek, de ugyanazon az adatbázison dolgoznak és ezzel szembe menve mégis külön repokban vannak tárolva; ez se jó gyakorlat,; ha az egyik módosul valószínűleg a másiknak is kell; inkább menjenek egy repoba
* független különálló alkalmazások vannak egy repoban

Verziókezelésnél az az alkalmazásoknál alapvető igény, hogy a felhasználói felületen legegyszerűbben meg tudjuk nézni, hogy milyen verzió van feltelepítve -> ez megkönnyíti a hibakeresést; tudnunk kell azt is, hogy milyen környezetre milyen verzió van kirakva; ezért érdemes ezt valahol a felhasználói felületen elérhetővé tenni (Sugó vagy About, git esetén annak hashét kiírni)

**Alkalmazások és csapatok kapcsolata**

**Conway törvény: **

= ha az adott fejlesztő cég pl. 3 csapattal rendelkezik, hogyhogy nem 3 modulból áll az alkalmazás; cél, hogy a csapatok külön tudjanak dolgozni, így hatékonyabbak, kevesebbet kell köztük szinkronizálni

egy **codebase több team ellentmond a microservice elképzeléseinek**

túl nagy alkalmazás és több csapat dolgozik rajta; rövidtávon egyik csapat besegíthet a másiknak, de ez nagyon belassít; hosszútávon; minden csapatnak legyen meg a saját kis micoservice-e; library függőségeket vegyük ki

**Feltöltés Git repositoryba - gyakorlat**

1.regisztráció

2.új repository létrehozása - >név -> public

README. MD -t létre lehet hozni az Ideából is! (nem pipáltuk be a létrehozását a repo kreálásnál); -> ha mi hozzuk létre, a projekt főkönyvtárába kell tenni

1. loákális git repo létrehozása a gépen, majd hozzá kell adni a fileokat, commitolni őket, majd hozzáadjuk a távoli repot, majd felpusholjuk a lokalt a remoteba:

git init = lokális repo létrehozás

git add . – fileok hozzáadása a repohoz

a start.spring.io létrehozott git ignoreja figyel

git status

git commit -m ”Initial commit” – állományokat commitoltuk a lokál repoba

git remote add origin https\*\* - távoli repo hozzáadása

git push -v origin master - lokális repo módosítások eljuttatása/feöltöltése a távoli repoba

githubra vissza és refresh

**Kérdések**

*Mi az a one codebase one application elv?*

*Verziókezelésnél az az alkalmazásoknál mik az alapvető igények?*

*A Conway törvéy mit mond az alkalmazások és csapatok kapcsolatáról?*

*Mik a lépései a lokális repo létrehozásától az első pushig tartó folyamatnak?*