

Övning – users and groups

Denna övning syftar till att du ska bli mer bekväm men hur användare och grupper fungerar i containrar då user namespace inte används (vilket är default i Docker). Följ punkterna nedan och svara på frågorna.

Användbara kommandon kan vara: ***id, cat /etc/passwd***

Tips: Lägg på `--rm` när du startar containrar så att de automatiskt tas bort när du stänger ner dem.

1. Starta en container från image `alpine:latest`. (***docker container run --rm -it alpine***)

- Vilken användare blir du i containern?

- Om du tittar i hosten, vilken användare kör processen `/bin/sh` i containern?

2. Start samma container som i punkt 1 men gör en bind mount från en katalog på din host som innehåller filer som din användare äger till valfri path i containern.

- Lista filerna i katalogen inifrån containern. Vem är ägare?

- Skapa en ny fil inifrån containern, vem kommer att äga den om du stänger ner containern och tittar i hostens filsystem?

3. Starta nu containern som i punkt 2 fast som en annan användare än root. Tips - kan göras med `--user` flaggan i run-kommandot, se <https://docs.docker.com/engine/reference/run/#user>.

- Behöver användaren finnas i containern för att det ska gå?

- Kan den användaren skriva filer till mountpointen?

4. Starta containern som i punkt 1 och mounta en volume istället för en bind mount. Skapa en fil i volumen inifrån containern.

- Vem äger filen om du tittat i hostens filsystem (leta reda på var filerna sparas med `docker volume inspect`)

5. Skriv en Dockerfile som utgår från `alpine:latest` och som skapar och sätter en användare. Default kommando kan ärvas från base image. Bygg imagen och starta en container utifrån den.

- Vilken användare blir du nu när den startats?

6. Starta containern från punkt 5 men mounta också en volume till den.

- Kan användaren skriva till volymen?

- Hur löser officiella images som tex redis detta problem? Tips: kolla i redis Dockerfile.