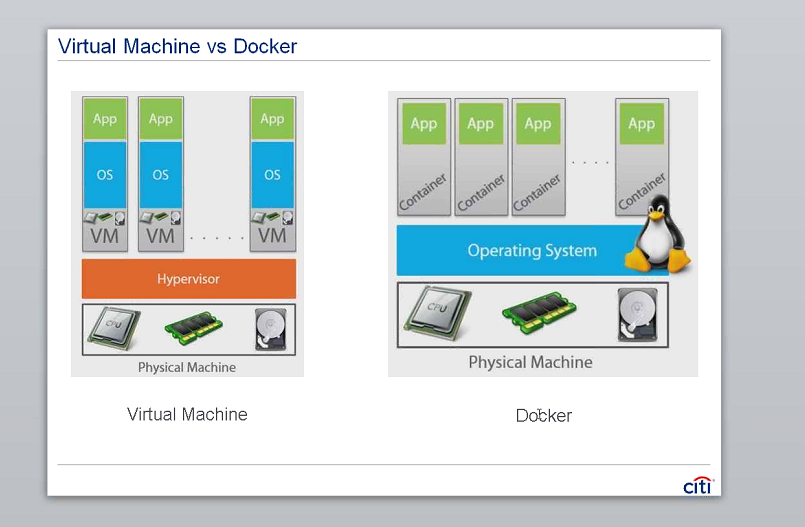
1. DOCKER IN ACTION

- zwiekszenie produktywności, praca w non-dev

- różnica z VM, virtualka pod kazda apke uzywa oddzielnego OS, a docker jednego

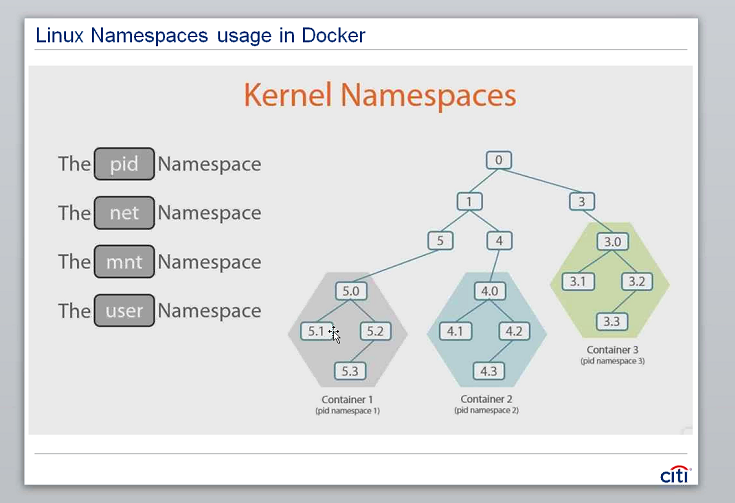


- nie obciża systemu jak wirtualka, nie ma pełnej separacji

- jest to runnable image

- uzywa wielu z linuxa dobryxh

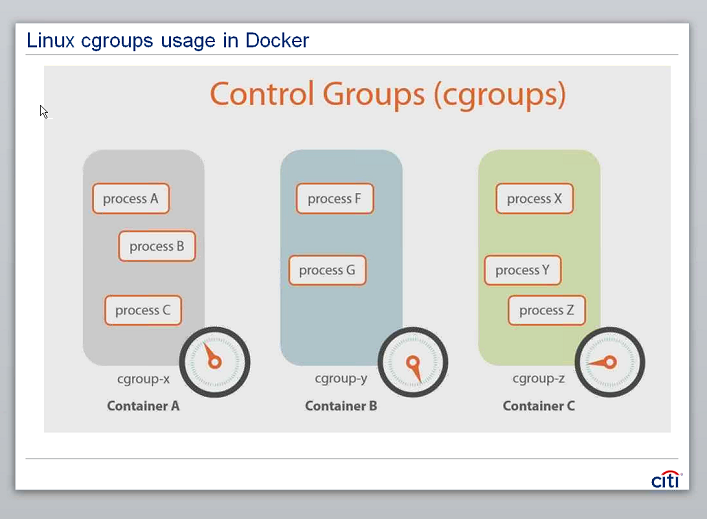
-- namespace ilka



- doker może kilowac hosta

- może pobrac wszystkie zasoby hosta

- mozna cgrupy z limitacj zasobów



Demo :

- dokiero uzywa warstw (layer), mozna kolejne warstwy na wartwach

- budujemy image z warstwami, i tworzymy image

- docker run –d –name „xx” example\_1 (obraz) „some param passed to entrypoint”

-d – deamon

- clear czyści console

-- more start.sh – wypisuje co w skrypcie

- mozna podać port –p $PORT:80 –v $VOLUME: - PRZY docker run

- w docker jest html\_content , tam mozna index.html – i on wyświetlany w przegladarce po starcie dockera

- linking – jak 1 kontener to 1 proces, to możemy chcec zlinkować te procesy na rożnych kontenerach dockera

- more Dokerfile

- ./clean.sh, build.sh

-- --docker run –d link docker-demo-test –name use\_image

-- docker exec link\_example env

- docker ps | grep docker-demo-test – widzimy ip I port

- jest UI do podgldania – wizualizacja jakie element s dostepne I podpite do dokr demo-test

- kazdy sobie odpala obraz lokalnie i moze pracowac

- mozna limitować, ale jak caly to działa jak na lokalu performance

- jak image 800MB to malo w pamici, nawet jak 20 uzyc to mniej niż 40 miega

-- jak warstwy w miejcu serwera, to nie trzeba dodatkowej pamieci

- wew mamy komponenty jak jmx, redis, docker-ui, docker-layers

-- mozna na raz odpalać całość