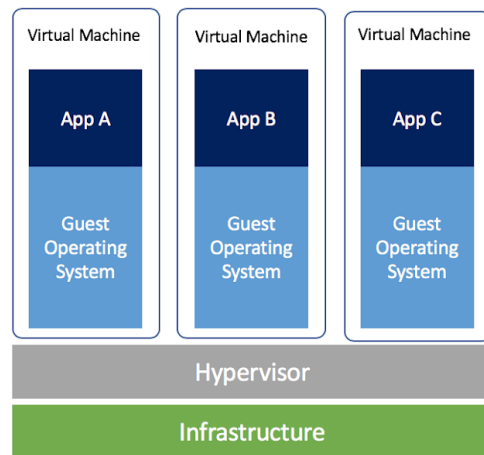
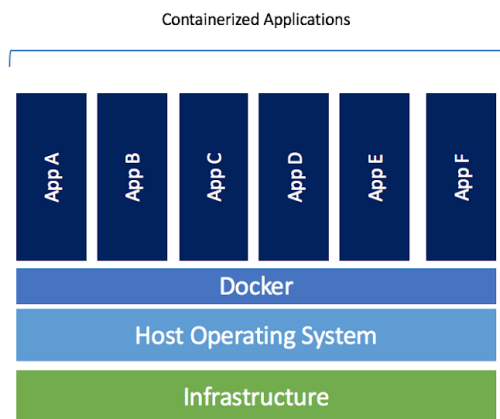


- Co to są wirtualki (właściwości, dlaczego warto używać, oprogramowanie wirtualizujące itd.),
- Plusy:
  - Izolacja: hyperwizory bardzo dobrze izolują gościa od hosta, dzięki czemu możesz używać maszyn wirtualnych do uruchamiania błędnego lub niezaufanego oprogramowania w dość bezpieczny sposób.
  - Snapshot: możesz zrobić „migawki” swojej maszyny wirtualnej, zapisując cały stan maszyny (dysk, pamięć itp.), wprowadzać zmiany w maszynie, a następnie przywrócić wcześniejszy stan. Jest to przydatne do testowania potencjalnie destrukcyjnych działań, między innymi."
- Minusy:
  - Maszyny wirtualne są zazwyczaj wolniejsze niż uruchamianie na sprzęcie fizycznym, więc mogą być nieodpowiednie do niektórych aplikacji.
- Hypervisors
  - [VirtualBox](#) (open-source)
  - [Virt-manager](#) (open-source, manages KVM virtual machines and LXC containers)
  - [VMWare](#) (commercial, available from IS&T [for MIT students](#))
  - [Xen](#) ( open-source, “Type 1” or “bare metal” which is more secure and faster, used in embedded)
  - Qubes OS (open-source, operating system with qubes isolationg parts of system, uses xen)
  - [libvirt](#) (manager)
- Hands on:
  - konfiguracja wirtualki na virtualbox (LUBUNTU 18!)
    - sciagnac lubuntu
    - skonfigurowac wirtualke
  - logowanie po ssh
    - sudo apt install openssh-server
    - sudo systemctl start sshd
    - skonfigurowac port forwarding w network settings: name ssh, protocol tcp, host port 3021, guest port 22
    - ssh -p 3021 [kn1@127.0.0.1](#)
  - zrobienie snapshotu
  - Przykład izolacji; Fork bomb [\(<https://en.wikipedia.org/wiki/Fork\\_bomb>](https://en.wikipedia.org/wiki/Fork_bomb) (|:(){ :|& };:|))
    - napisać skrypt fork.py:
 

```
import os
while True:
    os.fork()
```
    - uruchomić python3 fork.py
  - Przykład izolacji; sudo rm -rf --no-preserve-root /|
  - Utworzenie katalogu dzielonego między hostem i VM,
- Kontenery (właściwości, dlaczego warto używać, oprogramowania wirtualizującego itd.),



Kontenery to w zasadzie zestaw różnych funkcji bezpieczeństwa systemu Linux, takich jak wirtualny system plików, wirtualne interfejsy sieciowe, chroots, sztuczki z pamięcią wirtualną i tym podobne, które razem dają wrażenie wirtualizacji.

Nie są aż tak bezpieczne ani izolowane jak maszyny wirtualne, ale są dość bliskie i ciągle się poprawiają. Zwykle oferują wyższą wydajność i są znacznie szybsze do uruchomienia, ale nie zawsze.

Wzrost wydajności wynika z faktu, że w przeciwieństwie do maszyn wirtualnych, które uruchamiają całą kopię systemu operacyjnego, kontenery dzielą jądro Linuxa z hostem. Należy jednak pamiętać, że jeśli uruchamiasz kontenery Linux na Windows/macOS, to będzie musiał być aktywny Linux VM jako warstwa pośrednia między nimi.

- Plusy:
  - Systemy budowania
  - Środowiska programistyczne
  - Wstępnie zapakowane serwery
  - Uruchamianie niezauważalnych programów
  - Ocena prac studentów
  - (Niektóre) obliczenia w chmurze
  - Continuous Integration
  - Travis CI
  - GitHub Actions
  - Rozwiązywanie problemów z zależnościami – jeśli maszyna musi uruchamiać wiele usług z konfliktującymi zależnościami, mogą być one izolowane za pomocą kontenerów."
  
- Oprogramowanie:

- Amazon Firecracker
- Docker
- rkt
- lxc
- [singularity](#)
- Hands on na LUBUNTU 18:
  - konfiguracja kontenera na docker (UBUNTU): `curl -fsSL https://get.docker.com | sudo sh`
  - logowanie po ssh `docker run -it ubuntu bash`, `ctrl+p q`, `docker attach`,
  - wyłączenie i uruchamianie kontenerów, `docker stop`, `docker start`,
  - instalacja wcześniej przygotowanych kontenerów z repozytorium
  - uruchamianie singularity:
    - `singularity run library://sylabsed/examples/lolcow`
  - uruchamianie dockerowych kontenerów w singularity:
    - `singularity run docker://ubuntu`
  - pobieranie kontenerów:
    - `singularity pull ubuntu.sif docker://ubuntu`
  - wykonywanie poleceń:
    - `singularity exec ubuntu.sif ls`,
    - `singularity exec ubuntu.sif ls /opt`,
    - `ls /opt`.
  - instalacja mongoddb w wersji z persistent storage czyli serwis bazy i dane w oddzielnych kontenerach,
  - napisanie własnego `Dockerfile` i zabawa z repozytorium dockerhub i git (opcjonalnie w domu).