

Masz Linuksa? Masz chmurę!*

Wojciech Urbański



* oczywiście, że nie zrobię chmury z mojego komputera, ale....



Jaki masz problem?

- Chcę szybko tworzyć sobie maszyny wirtualne do pracy
 - Jak w chmurze - wybieram obraz i maszyna od razu działa
- Nie chcę się przejmować konfiguracją sieci ani adresami IP maszyn
- Chcę, żeby maszyny również mogły się komunikować po nazwach
- Nie chcę chmury publicznej:
 - Bo drogo
 - Bo nie chce mi się dbać o bezpieczeństwo tych instancji



Prosta decyzja: wirtualizacja

Wirtualizacja typu 1

- Brak oddzielnego SO
- Osobny sprzęt

Wirtualizacja typu 2

- Działa w systemie operacyjnym
- Współdzieli zasoby

Parawirtualizacja

- Zwirtualizowany system wie, że jest zwirtualizowany

Kandydat numer 1: VirtualBox

- ✓ Wieloplatformowy
- ✓ Może działać bez interfejsu graficznego (VBoxManage)
- ✓ Można też wszystko wyklikać
- ✓ Wspiera OVA
- ✓ Dobrze znany
- ✓ Całkiem dobra wydajność

✗ Jest od Oracle



Kandydat numer 2: KVM

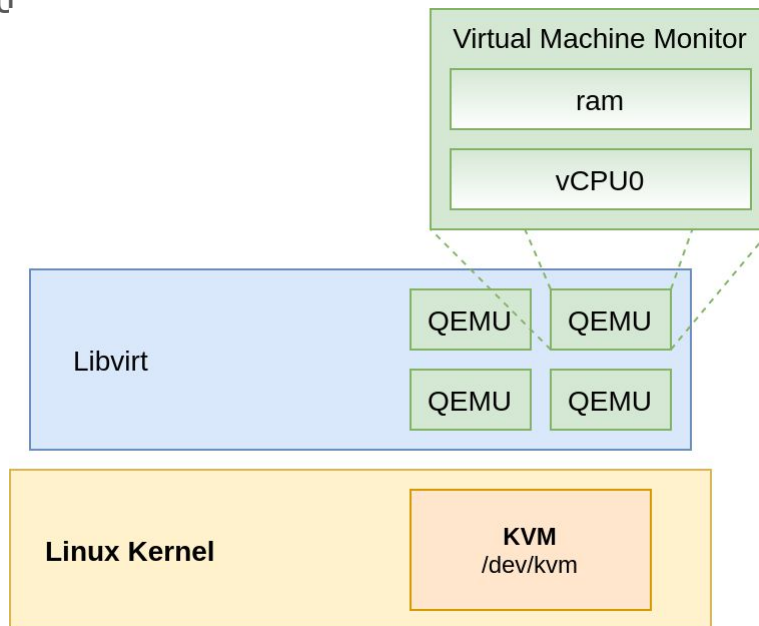
Kernel-based Virtual Machine

- ✓ Wbudowany w Linuksa
- ✓ Kolega polecał
- ✓ Nie używałem nigdy świadomie
- ✗ Czym się tym zarządza?
- ✗ Jakies XML-e, jakieś QEMU, WTF?



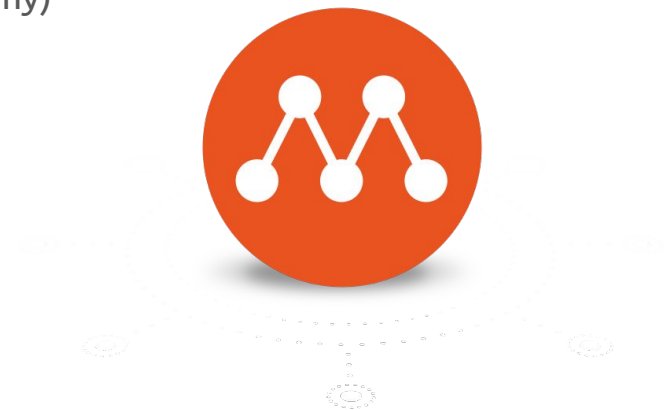
KVM, QEMU, Libvirt

- **/dev/kvm** - interfejs do abstrakcji sprzętu
- **QEMU** - Virtual Machine Monitor
- **Libvirt** - kontroluje QEMU (więc i KVM) z przestrzeni użytkownika



Jak tym zarządzać?

- Vagrant?
 - Dużo obrazów (różnej jakości...)
 - Konfiguracja w Ruby
 - Problemy z instalacją libvirt plugin
- Multipass?
 - Nowość od Canonical
 - Działa również na Windowsie i Macu (natywne platformy)
 - Teoretycznie łatwe w użyciu
 - Linux: snap-only! :(
- Może coś napisać samemu?
 - Dystrybucje tworzą gotowe obrazy dla KVM
 - Można sobie stworzyć obrazy samemu, np. Packerem
 - To nie może być trudne...



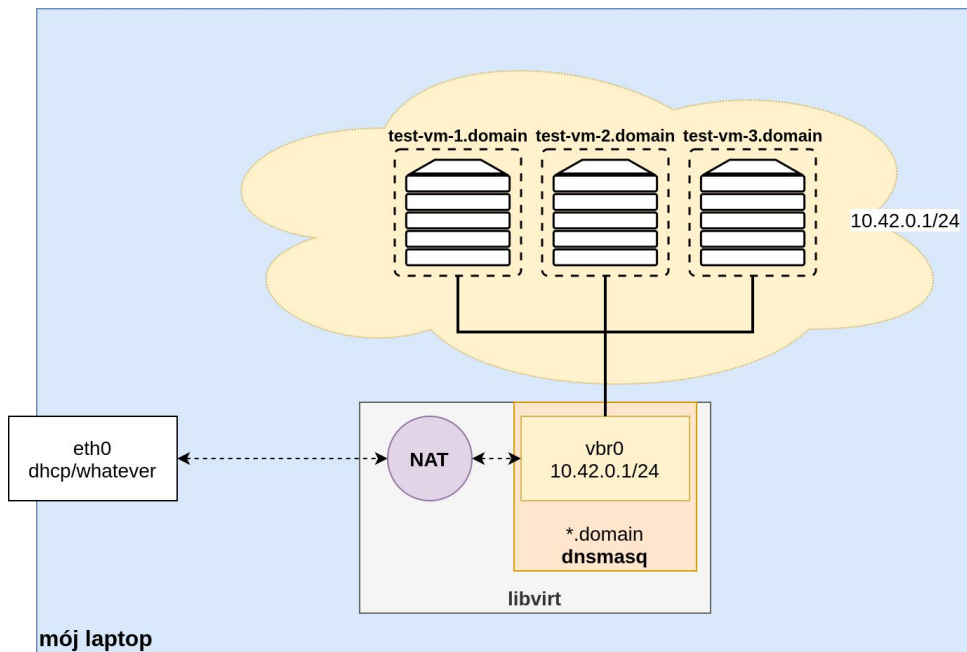
Krok 1: tworzenie maszyny wirtualnej w KVM

- GUI: virt-manager
 - New vm -> install from ISO
 - virt-manager -> import image -> konfiguracja
- Wykorzystanie gotowych "cloud image"
 - skopiuj/konwertuj obraz używając QEMU -> virt-install
 - Wymagana wstępna konfiguracja (klucze, sieć...) -> cloud-init
- **cloud-init**
 - Wymaga źródła danych (chmury mają swoje, ale jest też **nocloud**)
 - Pliki user-data i meta-data podane jako... CD-ROM!



Krok 2: sieci w kvm

- Trzy dostępne w `libvirt` tryby sieci:
 - Bridge
 - NAT
 - Passthrough
- NAT -> `dnsmasq` (DHCP + DNS)
 - Definicja puli adresowej sieci i puli DHCP
 - Rejestracja w DNS nazw hostów z DHCP
 - Możliwość zdefiniowania "domeny"



Krok 3: Komunikacja host <> VM

- Ulotna, tymczasowa domena - nie chcę jej konfigurować w publicznym DNS
- Trwałe przekierowanie zapytań dla danej domeny:
 - Network-manager
 - dnsmasq
- **Systemd-resolved...**

[● ◀] **systemd**

/bin/bash



Krok 4: automatyzacja

*Jeśli potrzebujesz w bashu tablic asocjacyjnych...
...to potrzebujesz czegoś lepszego od basha.
~Paulo Devopselho*

Funkcjonalności: tworzenie sieci, pobieranie obrazów z sieci, tworzenie maszyn

260 linii kodu w bashu z tablicami asocjacyjnymi. (27 w pythonie)

Działa? No zaraz zobaczemy...





Podsumowując

github.com/wurbanski/makeshift-cloud

Czy to działa? No działa xD

Czy polecam innym? Pewnie!*

Jedna osoba poza mną z tego skorzystała, więc to 200% lepsze przyjęcie niż myślałem.



Dziękuję!

