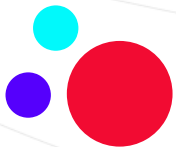


Wstęp do wirtualizacji

infoShare Academy



HELLO

Michał Jerzy Wierzbicki

Dev turned DevOps





Agenda

1. Co to jest wirtualizacja
2. Po co wirtualizacja
3. Hyperwizory
4. Stwórzmy sobie VM ręcznie
5. Vagrant – wprowadzenie
6. Vagrant – providers
7. Vagrant – box
8. Vagrant – provisioners
9. Vagrant – w akcji



01. Co to jest **wirtualizacja**?

czyli o czym ten kurs nie będzie

Co to jest **wirtualizacja** ?

Na czym się skupimy

Maszyny Wirtualne (VM)

Pełna wirtualizacja

Parawirtualizacja

Wsparcie sprzętowe (Intel VT, AMD-V)

Hypervizory

O czym tylko wspomnimy

Virtual LAN (VLAN)

Virtual eXtended LAN (VXLAN)

Virtual routing and forwarding (VRF)

Open vSwitch

Virtual Logical Unit Number (VLUN)

Virtual Memory

Input/Output Memory Management Unit (IOMMU)

Single root input/output virtualization (SR-IOV)



Co to jest **wirtualizacja** ?

Wirtualizacja – zazwyczaj na “chłopski rozum” to w softwarze udawanie że jest więcej hardware’u

Maszyna wirtualna (VM) – “komputer (guest) w komputerze (host)”

Pełna wirtualizacja – można odpalić dowolny system, jest wolniej niż przy parawirtualizacji

Parawirtualizacja – guest wie, że jest wirtualizowany, ale wymaga to ingerencji w kod guesta

Intel VT-x (CPU), VT-d (I/O), VT-c (I/O)

AMD-V (CPU), AMD-Vi (I/O)



01. Po co wirtualizacja?

technologie nie powstają bez powodu



Po co **wirtualizacja** ?

Prostsza administracja

Im więcej klientów o
odmiennych potrzebach
współdzielących
środowisko, tym większy
ból głowy.

(Dependency Hell)



Po co **wirtualizacja** ?

Prostsza administracja

Im więcej klientów o
odmiennych potrzebach
współdzielących
środowisko, tym większy
ból głowy.
(Dependency Hell)

Konsolidacja

Obecny trend w data center to
zastępowanie 4 szaf po sufit
pełnych starych serwerów
pojedynczą szafą pełną
nowych serwerów.



Po co **wirtualizacja** ?

Prostsza administracja

Im więcej klientów o
odmiennych potrzebach
współdzielących
środowisko, tym większy
ból głowy.
(Dependency Hell)

Konsolidacja

Obecny trend w data center to
zastępowanie 4 szaf po sufit
pełnych starych serwerów
pojedynczą szafą pełną
nowych serwerów.

U mnie działa

Gdy osoba instalująca
oprogramowanie na
produkcji dostaje jakiś
błąd, nic jej nie ucieszy jak
stare dobre "ale u mnie
działa".

Hyperwizory

Typu 1

Nie wymaga systemu operacyjnego (bo nim jest)
Używany głównie w datacenter i korporacjach
Mniejszy overhead

Np:

Xen

VMWare ESX(i)

Hyper-V (Azure)

Typu 2

Wymaga systemu operacyjnego
Używany głównie przez developerów
Prostota (relatywnie) użycia

Np:

Virtualbox

KVM

VMWare Workstation

Hyper-V (Windows)



01. Stwórzmy sobie VM ręcznie

żeby potem docenić automatyzację



Vagrant - wprowadzenie

“takie CLI dla różnych hyperwizorów”

Realizuje ideę Infrastructure as a Code (IaC): Vagrantfile (plik) + vagrant (program) = VM

Dokumentacja: <https://www.vagrantup.com/docs>



Vagrant - providers

Vagrant provider to middleware pomiędzy vagrantem (i konfiguracją z Vagrantfile) a programem który na koniec ma faktycznie postawić VM.

Vagrant wspiera out-of-the-box:

- Virtualbox
- Hyper-V
- VMWare Workstation / Fusion
- Docker

a dzięki pluginom (custom provider):

- libvirt
- Azure
- Google Compute Engine

i wiele innych



Vagrant - **box**

Przygotowany specjalnie pod konkretnego providera szablon systemu bazowego.

<https://app.vagrantup.com/boxes/search>

Boxy można tworzyć np. przy użyciu innego narzędzia firmy HashiCorp, Packer

<https://www.packer.io/plugins/builders/vagrant>

Albo ręcznie przy wykorzystaniu samego Vagranta

<https://www.vagrantup.com/docs/boxes/base>



Vagrant - provisioners

Vagrant provisioner to middleware pomiędzy vagrantem (i konfiguracją z Vagrantfile) a programem który dokona konfiguracji już postawionej maszyny wirtualnej.

Vagrant wspiera out-of-the-box:

- file upload
- shell upload & execution
- Ansible
- Puppet
- Chef
- Salt
- docker



01.

Vagrant – w akcji

a terraform i compose są jeszcze fajniejsze

**THANK YOU
FOR YOUR
ATTENTION**

infoShareAcademy.com

PYTANIA

infoShare Academy