

VAGRANT

Konfiguracja maszyny wirtualnej



ZANIM ZACZNIESZ!

Zweryfikuj czy masz **włączoną sprzętową wirtualizację** wchodząc w **menadżer zadań**, **wydajność**, klikając na **CPU** i weryfikując pole **wirtualizacja**.

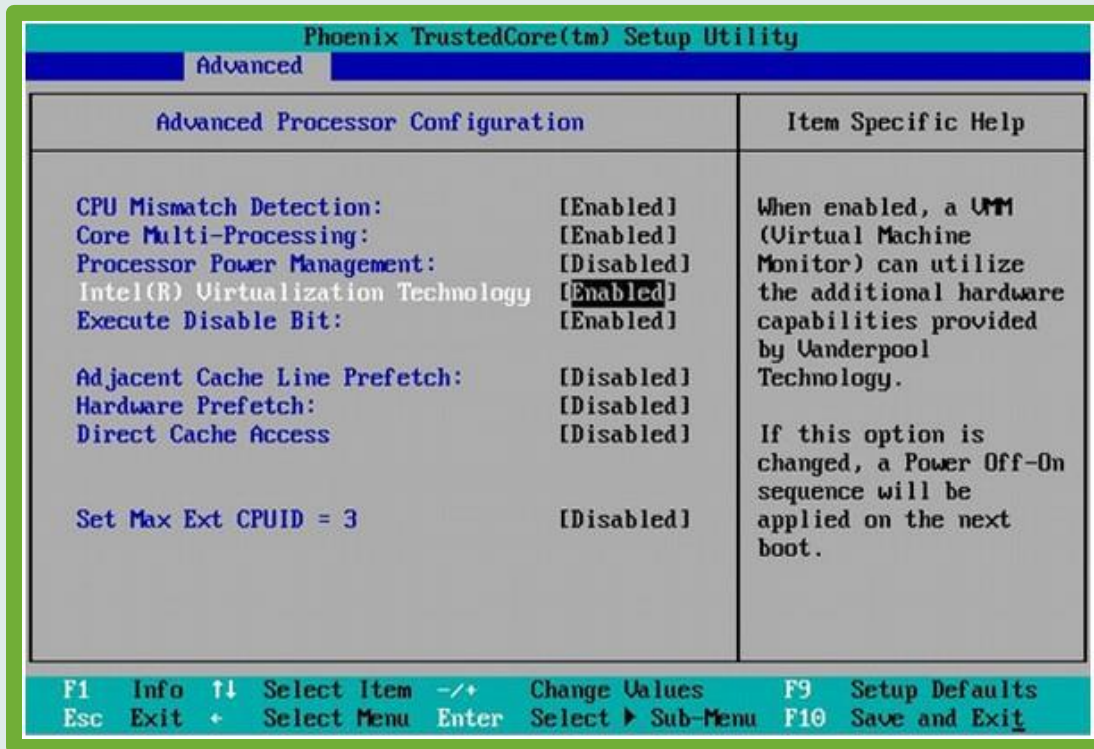
The screenshot shows the Windows Task Manager Performance tab. The 'CPU' section is selected, displaying a graph of CPU usage and a table of system specifications. The 'Wirtualizacja' (Virtualization) status is highlighted as 'Włączone' (Enabled).

Wykorzystanie	Szybkość	Szybkość podstawowa:	3,50 GHz	
58%	4,38 GHz	Gniazda:	1	
Procesy	Wątki	Dojścia	Rdzenie:	4
251	2937	118111	Procesory logiczne:	4
Czas pracy				0:04:58:19
Pamięć podręczna poziomu 1:				256 KB
Pamięć podręczna poziomu 2:				1,0 MB
Pamięć podręczna poziomu 3:				6,0 MB

Wirtualizacja: **Włączone**

ZANIM ZACZNIESZ!

Jeżeli wirtualizacja jest wyłączona musisz znaleźć stosowne pole w swoim BIOS/UEFI (Zazwyczaj pod nazwą **virtualization** – różne dla każdego dostawcy) i ustawić w trybie **enabled**.



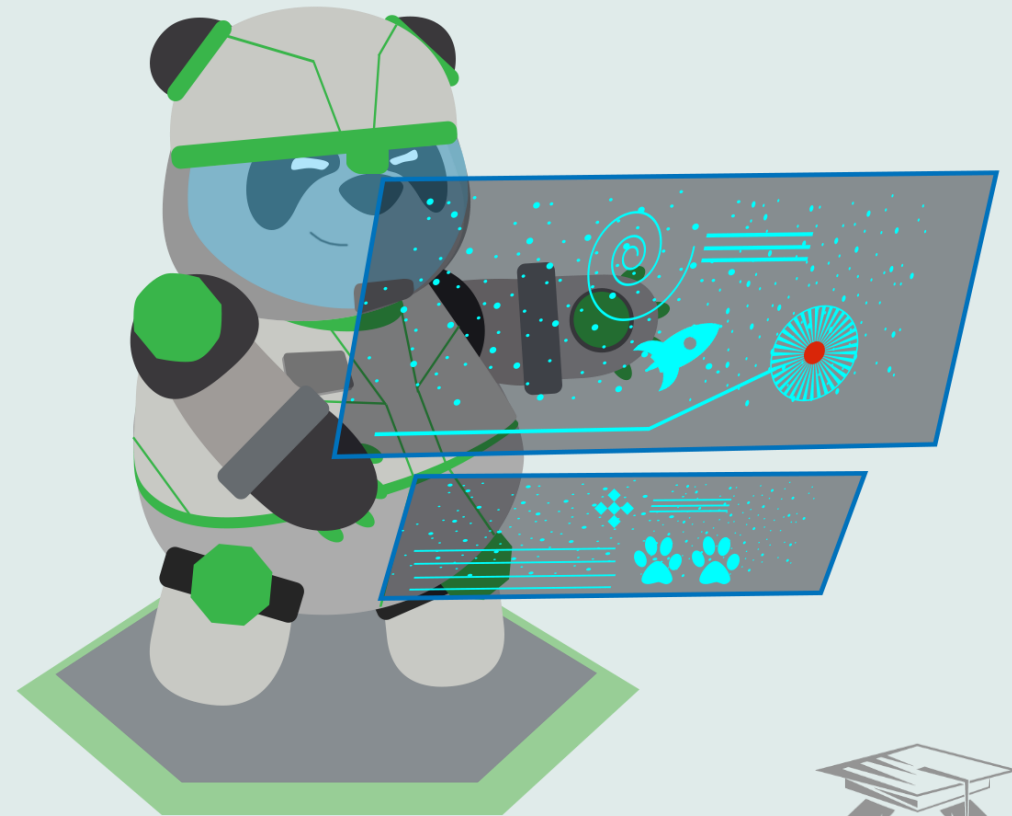
ZACZYNAJEMY

Na początek wejdź na stronę VirtualBox i pobierz klienta odpowiedniego dla Twojego systemu operacyjnego:

<https://www.virtualbox.org/wiki/Downloads>

VirtualBox 6.1.8 platform packages

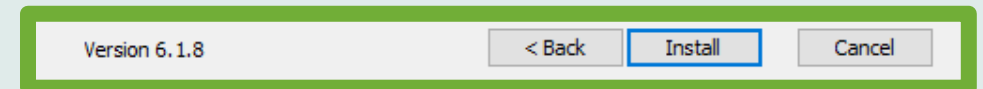
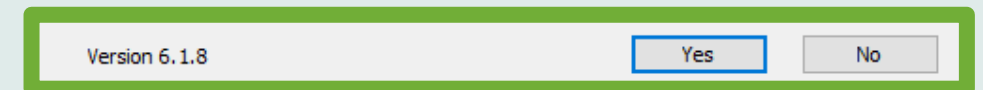
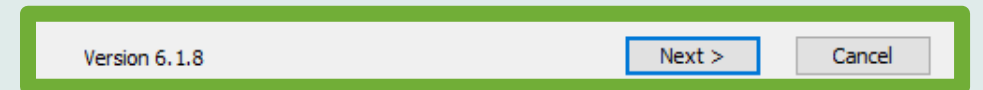
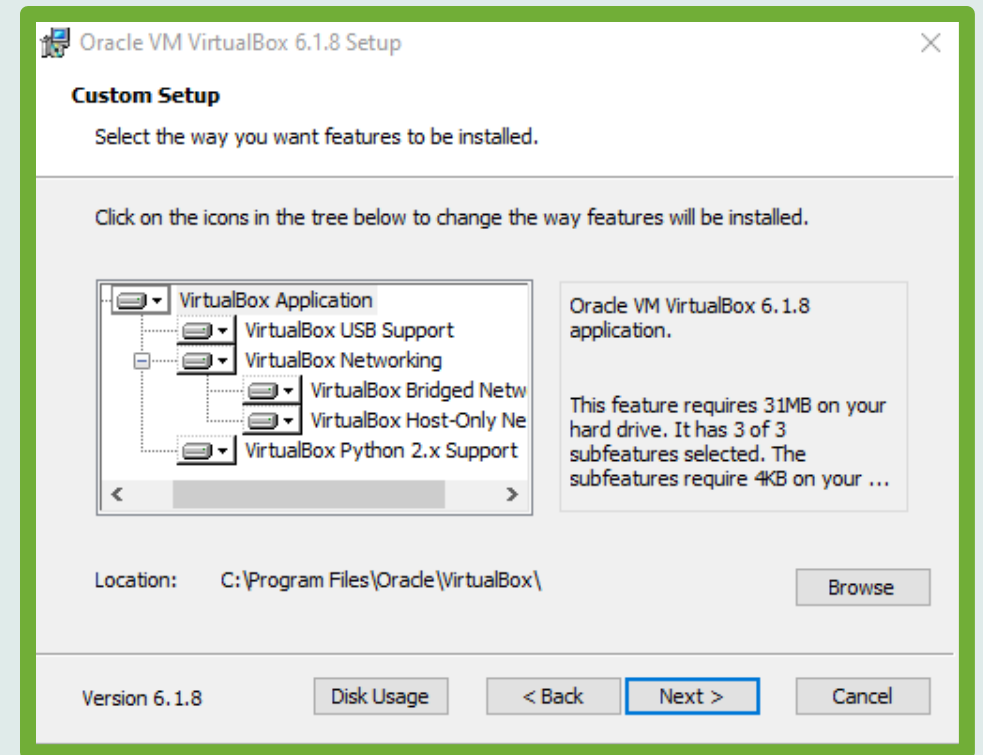
- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)



VIRTUALBOX

Zainstaluj **VirtualBox**. Na panelu po lewej możesz wybrać folder instalacji. Pamiętaj również, że VirtualBox może chwilowo zrestartować połączenie sieciowe, więc zapisz wszystkie aktywne zadania.

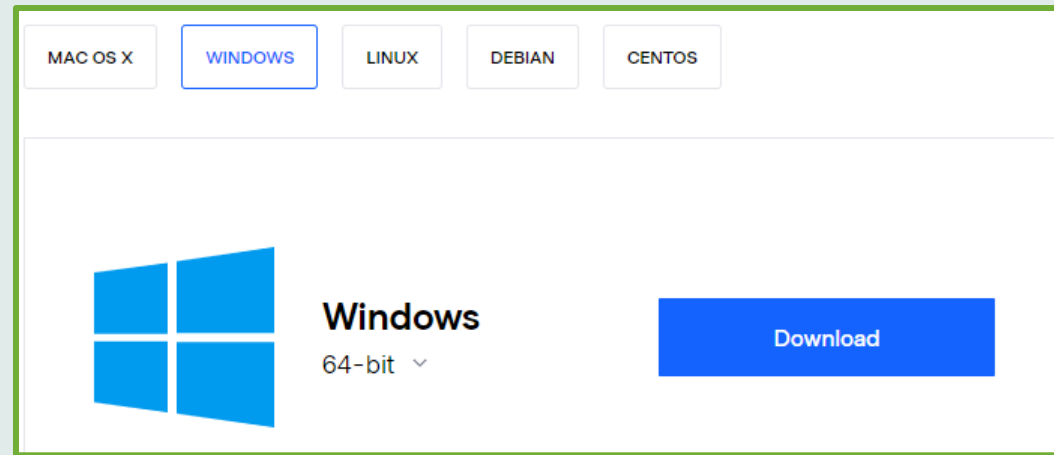
Korzystaj z przycisków **next**, **yes** i **install**, aby kontynuować.



VAGRANT

Następny krok to instalacja Vagranta.
W tym celu wejdź na stronę poniżej.
Pamiętaj, aby wybrać odpowiednią platformę
oraz wersję (32/64 bit).

<https://www.vagrantup.com/downloads>



VAGRANT



Podobnie jak z VirtualBox wykorzystaj przyciski **next** i **install**, aby zainstalować Vagranta. Po tym kroku musisz zrestartować komputer.

GIT

Trzecie i ostatnie narzędzie do zainstalowania to Git. Możesz skorzystać też z wersji przenośnej (**Portable**), która nie wymaga instalacji, a tylko pobrania.

<https://git-scm.com/download/win>

Other Git for Windows downloads

Git for Windows Setup

32-bit Git for Windows Setup.

64-bit Git for Windows Setup.

Git for Windows Portable ("thumbdrive edition")

32-bit Git for Windows Portable.

64-bit Git for Windows Portable.



LOKALIZACJA DEMO

Przejdź do folderu z Gitem (niezależnie czy zainstalowanym, czy przenośnym) i uruchom konsolę `git-cmd`. Konsola, domyślnie, otworzy się w Twoim katalogu użytkownika.

Aby uruchomić nasze demo, potrzebujesz przynajmniej **20 GB** wolnego miejsca na dysku. Jeżeli posiadasz tyle na dysku C, możesz pozostać w tym katalogu, jeżeli nie to przejdź do wybranego folderu na innym dysku: `<litera dysku>:`, a następnie `cd <ścieżka\do\folderu>`.

bin	20.04.2020 18:10	Folder plików	
cmd	01.06.2020 13:41	Folder plików	
dev	20.04.2020 18:10	Folder plików	
etc	20.04.2020 18:10	Folder plików	
mingw64	20.04.2020 18:10	Folder plików	
tmp	20.04.2020 18:10	Folder plików	
usr	20.04.2020 18:10	Folder plików	
git-bash	20.04.2020 17:34	Aplikacja	150 KB
git-cmd	20.04.2020 17:34	Aplikacja	149 KB
LICENSE	20.04.2020 18:10	Dokument tekstowy	19 KB
magic.py	20.04.2020 05:11	Plik PY	9 KB
post-install	20.04.2020 18:10	Plik wsadowy Win...	2 KB
README.portable	20.04.2020 18:10	Plik PORTABLE	4 KB

```
C:\Users\XXX\Vagrant>cd d:\Panda  
  
C:\Users\XXX\Vagrant>d:  
  
d:\Panda>
```

POBRANIE REPOZYTORIUM

Teraz musisz pobrać repozytorium z github na swój komputer.
Zrobisz to za pomocą komendy:

```
git clone https://github.com/Panda-Academy-Workshops/Minikube
```

Wejdź do katalogu z repozytorium i przejdź na gałąź `clean`:

```
cd Minikube
```

```
git checkout clean
```

W konsoli, będąc nadal w katalogu Minikube, wpisz komendę `vagrant up`.

URUCHOMIENIE LABORATORIUM

Aby wejść do maszyny wirtualnej, w konsoli, będąc nadal w katalogu Minikube, wpisz komendę `vagrant ssh`. Minikube działa z użytkownikiem root dlatego pierwszą komendą którą musisz wydać jest:

```
sudo su
```

Przydatne jest też dodanie autouzupełniania dla `kubectl`:

```
source /usr/share/bash-completion/bash_completion
```

Na następnych stronach znajdziesz komendy, których możesz użyć do odtworzenia treści laboratorium

UŻYWANE KOMENDY - PODSTAWY

`docker ps --format {{.Names}}` – wyświetla nazwy uruchomionych kontenerów

`cd /etc/kubernetes/manifests` – katalog z plikami yml podów systemowych

`kubectl run nginx --image=nginx` – uruchomienie kontenera nginx

`cd /vagrant/ && kubectl apply -f rs.yml` – uruchomienie replica-set

`kubectl get pod -A` – wyświetla pody

`kubectl delete pod <nazwa poda>` – usuń pod

UŻYWANE KOMENDY - ETCD

```
ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --  
cacert=/var/lib/minikube/certs/etcd/ca.crt --  
cert=/var/lib/minikube/certs/etcd/server.crt --  
key=/var/lib/minikube/certs/etcd/server.key get / --prefix --  
keys-only – wyświetl dostępne klucze
```

```
ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --  
cacert=/var/lib/minikube/certs/etcd/ca.crt --  
cert=/var/lib/minikube/certs/etcd/server.crt --  
key=/var/lib/minikube/certs/etcd/server.key del  
/registry/pods/default/nginx – usuń klucz z podem nginx
```

UŻYWANE KOMENDY - SCHEDULER

```
kubectl run nginx3 --image=nginx -o yaml --dry-run=client > pod.yml
```

– stwórz “na sucho” definicję poda w pliku pod.yml

```
kubectl get nodes
```

– sprawdź dostępne węzły

```
nodeName: vagrant
```

– brakujący fragment w pliku pod.yml

```
kubectl apply -f pod.yml
```

– uruchomienie poda z pliku pod.yml

UŻYWANE KOMENDY - CONTROLLER

```
ETCDCTL_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --  
cacert=/var/lib/minikube/certs/etcd/ca.crt --  
cert=/var/lib/minikube/certs/etcd/server.crt --  
key=/var/lib/minikube/certs/etcd/server.key get / --prefix --  
keys-only|grep controller
```

– wyświetlenie listy kluczy z controller w nazwie

```
docker rm -f `docker ps --format {{.Names}}|grep nginx`
```

– usuń wszystkie kontenery z nginx w nazwie

`nodeName: vagrant` – brakujący fragment w pliku pod.yml

```
watch kubectl get pod -A
```

– obserwuj listę podów (wyjście – ctrl+c)

```
kubectl apply -f pod.yml
```

– uruchomienie poda z pliku pod.yml

VAGRANT - USUNIĘCIE

Aby całkowicie usunąć maszynę wirtualną należy w terminalu wewnątrz katalogu **Minikube** wpisać komendę **vagrant destroy -f**.



```
C:\Users\  
          \Downloads\Vagrant\devops-workshop>vagrant destroy -f  
==> default: Forcing shutdown of VM...  
==> default: Destroying VM and associated drives...
```

DZIĘKUJEMY !

Gotowe! Na koniec odinstaluj (jeżeli nie chcesz ich już używać) programy z pierwszych slajdów!

Dziękujemy za uczestnictwo w warsztatach!

