

Instrukcja konfiguracji środowiska

1. WPROWADZENIE	1
1.1 MASZINY WIRTUALNE	1
2. URUCHOMIENIE MASZINY WIRTUALNEJ	3
2.1 OBRAZ BAZOWY	3
2.2 KONFIGURACJA MASZINY WIRTUALNEJ Z UŻYCIEM OBRAZU BAZOWEGO	3
2.3 LOGOWANIE DO SYSTEMU I WERYFIKACJA USTAWIEŃ	7
2.4 PODSTAWOWE POLECENIA W SYSTEMIE DEBIAN GNU/LINUX	10
3. KONFIGURACJA USŁUG SERWERA APLIKACJI INTERNETOWYCH	12
3.1 INSTALACJA I KONFIGURACJA SERWERA HTTP APACHE	12
3.1.1 INSTALACJA SERWERA Z REPOZYTORIUM PAKIETÓW	12
3.1.2 STRUKTURA PLIKÓW KONFIGURACYJNYCH SERWERA	13
3.1.3 KONFIGURACJA HOSTÓW WIRTUALNYCH	15
3.2 INSTALACJA I KONFIGURACJA BAZY DANYCH MONGODB	17
3.3 INSTALACJA I KONFIGURACJA INTERPRETERA PHP	18
4. PRZYGOTOWANIE MASZINY WIRTUALNEJ DO TRANSPORTU	21
4.1 ROZMIAR OBRAZU DYSKU MASZINY WIRTUALNEJ	21
4.2 EKSPORT MASZINY WIRTUALNEJ WRAZ Z USTAWIENIAMI	21
4.2.1 EKSPORT MASZINY DO PLIKU W FORMACIE OVFA	21
4.2.2 IMPORT PLIKU OVFA	24
DODATEK A. ZALETY MASZYN WIRTUALNYCH I ICH ROLA W BRANŻY IT	25

1. Wprowadzenie

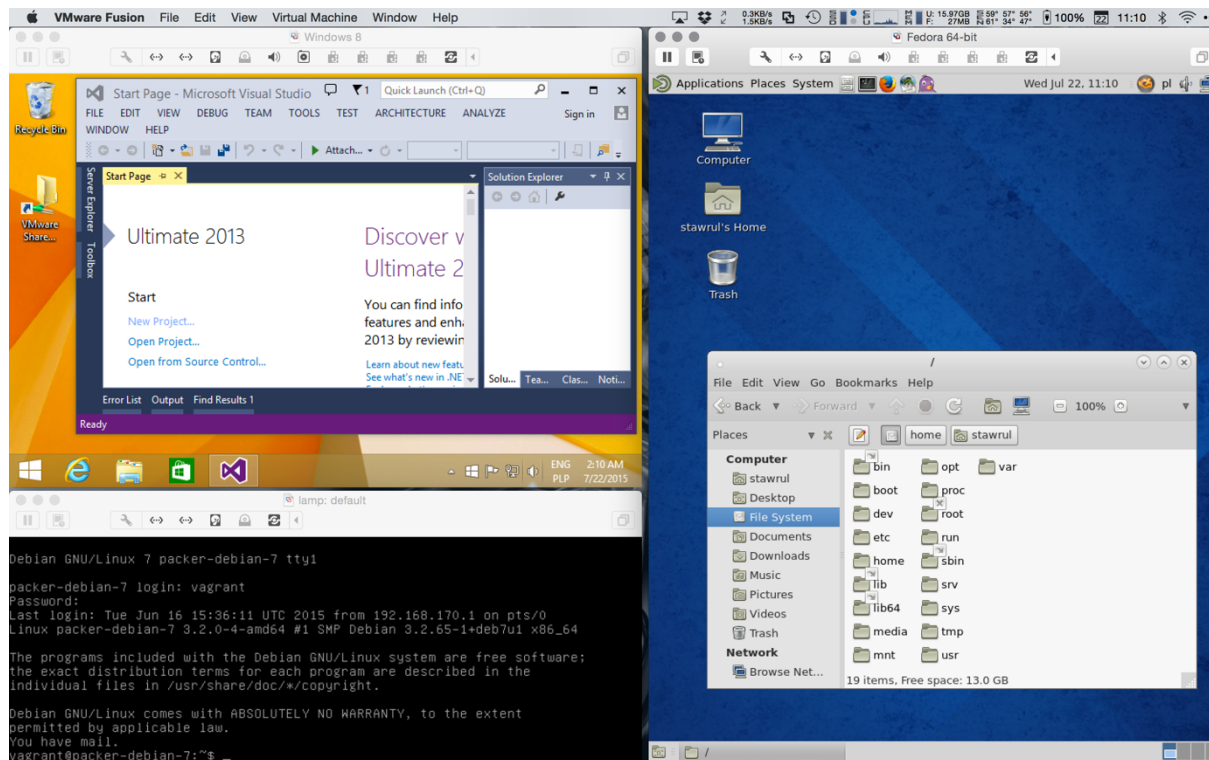
Instrukcja opisuje konfigurację środowiska na potrzeby projektu z przedmiotu Wytwarzanie Aplikacji Internetowych. Środowisko opiera się na maszynie wirtualnej, którą należy wykorzystać w czasie rozliczania poszczególnych etapów projektu. Zaleca się, aby maszyna wirtualna była stosowana również w czasie pracy nad aplikacją na komputerze osobistym co ułatwi późniejsze uruchomienie projektu w sali laboratoryjnej.

1.1 Maszyny wirtualne

Maszyny wirtualne pozwalają na uruchomienie systemu operacyjnego *gościa* w obrębie aplikacji, działającej w przestrzeni użytkownika systemu operacyjnego *hosta*. Dzięki temu możliwe jest uruchomienie na pojedynczej maszynie fizycznej (na jednym komputerze) wielu systemów operacyjnych równocześnie (rys. 1). System operacyjny gościa wymaga do pracy podstawowych komponentów, spotykanych w rzeczywistych komputerach: procesora, pamięci operacyjnej, pamięci trwałej, urządzeń wejścia/wyjścia. System gościa nie otrzymuje bezpośredniego dostępu do komponentów maszyny fizycznej – tymi zarządza system hosta. Aby system gościa mógł działać,

konieczne jest zatem zrealizowanie wymaganych przez niego komponentów w przestrzeni wirtualnej. Stąd właśnie określenie *maszyna wirtualna* – w kontraście do *maszyny fizycznej* – czyli składająca się z wirtualnych (a nie fizycznych) komponentów. Więcej informacji na temat maszyn wirtualnych znaleźć można w *Dodatku A. Zalety maszyn wirtualnych i ich rola w branży IT*.

Rys. 1 System hosta (macOS 10.10) z trzema maszynami wirtualnymi: Windows 8.1 (po lewej u góry), Fedora Linux (po prawej), Debian 7.8 w trybie tekstowym (po lewej, na dole).



W dalszej części opisane zostanie wykorzystanie darmowego oprogramowania Oracle VirtualBox, które oferuje pełną wirtualizację dla niezmodyfikowanych systemów gości. Aplikacja jest dostępna do pobrania ze strony projektu: <http://www.virtualbox.org>. Dostępne są wersje dla różnych systemów hostów: Windows, macOS, Linux oraz Solaris. Pomiedzy różnymi systemami operacyjnymi hosta mogą występować różnice w układzie interfejsu programu VirtualBox. Prezentowane w dalszej części zrzuty ekranu pokazują interfejs dla systemów z rodziny Windows. Zaleca się wykorzystanie programu VirtualBox w najnowszej wersji dostępnej na stronie produktu. Jego instalacja nie odbiega od instalacji innych aplikacji w systemie, więc jej opis został pominięty.

W punkcie 2. *Uruchomienie maszyny wirtualnej* omówiono przygotowanie konfiguracji maszyny wirtualnej, w oparciu o bazowy obraz ze strony przedmiotu. Opis kończy się uruchomieniem maszyny i zweryfikowaniem wprowadzonych ustawień. Punkt 3. *Konfiguracja usług serwera aplikacji internetowych* opisuje operacje do wykonania w systemie gościa, w celu uzyskania funkcjonalnego serwera aplikacji WWW. Tak przygotowaną maszynę wirtualną należy wykorzystać do pracy nad projektem i jego późniejszej prezentacji na komputerze laboratoryjnym. Informacje o przenoszeniu maszyny wirtualnej z komputera osobistego na stację roboczą w sali laboratoryjnej przedstawiono w punkcie 4. *Przygotowanie maszyny wirtualnej do transportu*.

2. Uruchomienie maszyny wirtualnej

2.1 Obraz bazowy

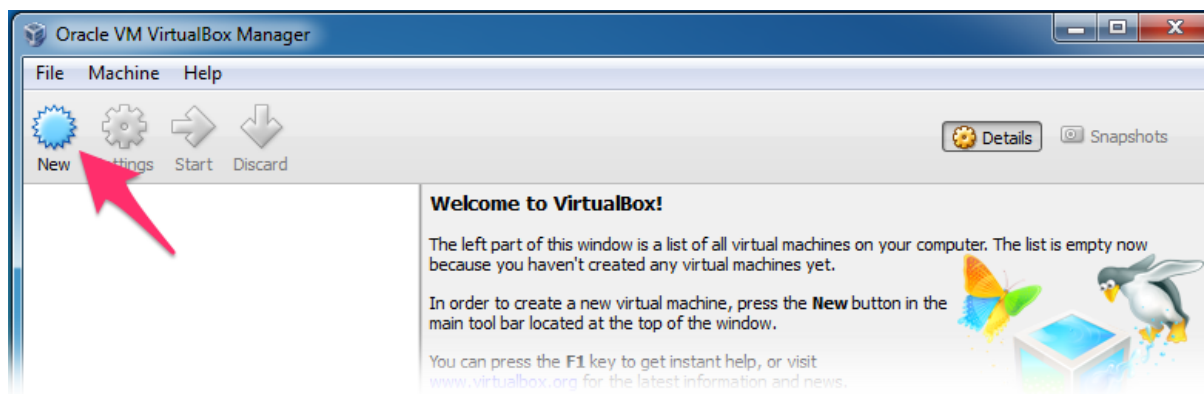
Bazowy obraz jest dostępny do pobrania na stronie przedmiotu. Ma on postać archiwum ZIP o rozmiarze ok. 180 MB, którego rozpakowanie da plik VDI o rozmiarze ok. 750 MB. Limit rozmiaru dysku maszyny wirtualnej został ustawiony na 8 GB, co jest wystarczające do zainstalowania wszystkich wymaganych komponentów i realizacji projektu. Po przeprowadzeniu operacji konfiguracyjnych opisanych w dalszej części plik VDI uzyska rozmiar ok. 1,4 GB.

Plik w formacie VDI (ang. *Virtual Disk Image*) stanowi obraz dysku twardego maszyny wirtualnej z zainstalowanym systemem operacyjnym Debian 9. System nie posiada powłoki graficznej – konfiguracja przebiega w trybie tekstowym, jak w typowych zastosowaniach serwerowych. Krótki przegląd podstawowych poleceń do pracy z tym systemem w trybie tekstowym znajduje się w punkcie 2.4 *Podstawowe polecenia w systemie Debian GNU/Linux*.

2.2 Konfiguracja maszyny wirtualnej z użyciem obrazu bazowego

Plik VDI stanowi jedynie obraz dysku twardego maszyny wirtualnej. Aby go wykorzystać, konieczne jest skonfigurowanie pozostałych komponentów maszyny, jak np.: pamięć operacyjna, procesor, karty sieciowe itd. Przygotowanie maszyny wirtualnej przebiega w kilku krokach opisanych poniżej.

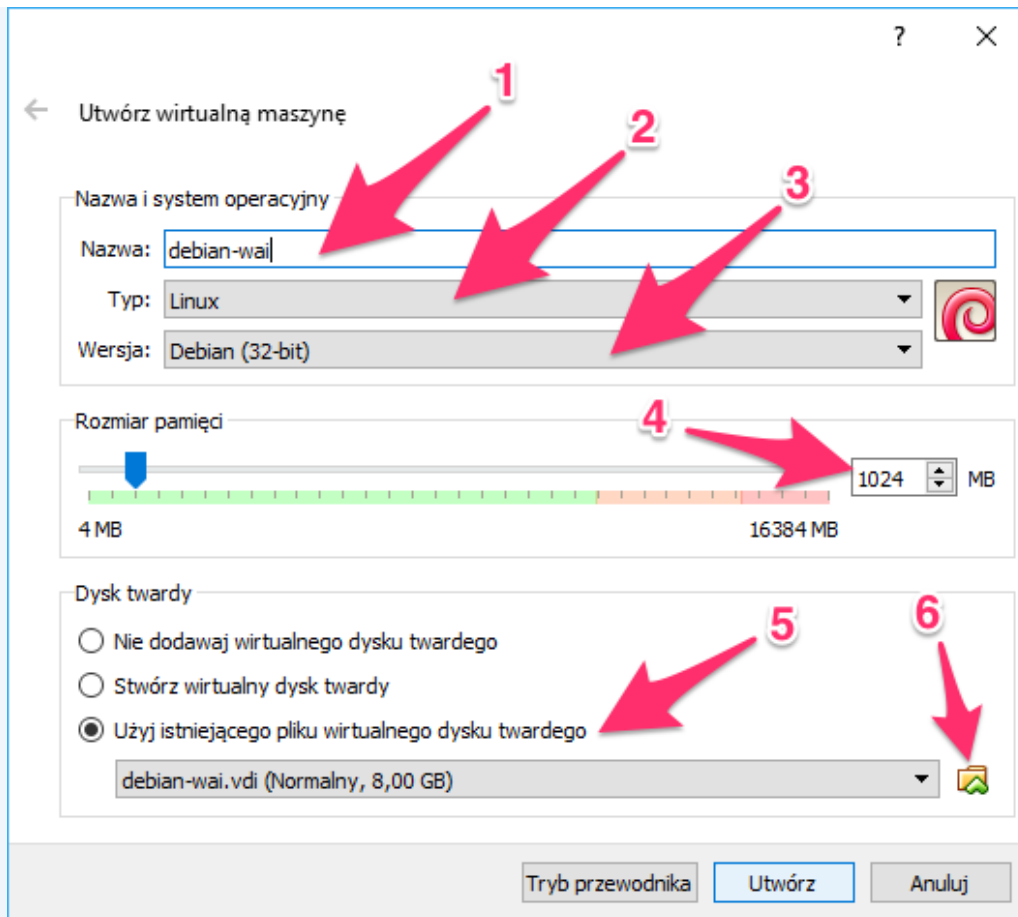
Krok 1 W głównym oknie programu VirtualBox należy kliknąć przycisk *New*, aby dodać nową maszynę wirtualną:



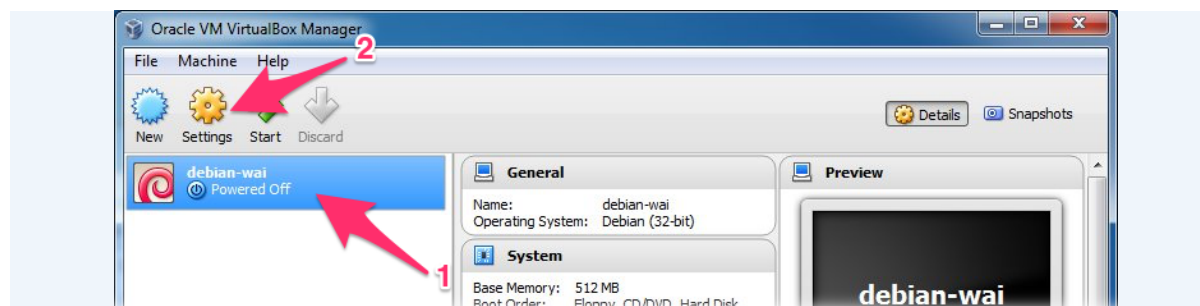
Krok 2 W oknie, które się pojawi, należy wprowadzić podstawowe ustawienia maszyny wirtualnej:

- nazwa maszyny jest dowolna (w dalszych przykładach wykorzystywana będzie nazwa *debian-wai* – pole (1) na zrzucie ekranu poniżej);
- typ systemu: Linux (2);
- wersja: Debian (32-bit) (3);
- rozmiar pamięci: 1024 MB (4);

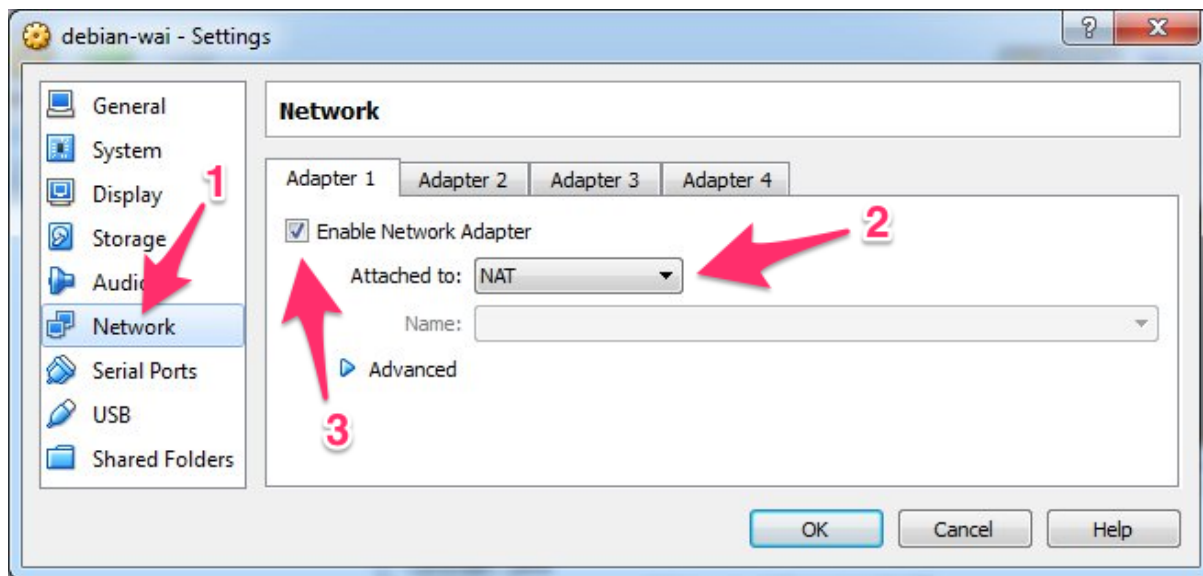
Następnie konieczne jest wybranie obrazu dysku twardego dla maszyny wirtualnej. Należy zaznaczyć opcję *Użyj istniejącego pliku wirtualnego dysku twardego (Use an existing virtual hard drive file)* (5) i wybrać z dysku plik *debian-wai.vdi* pobrany ze strony przedmiotu (6). Kliknięcie przycisku *Utwórz (Create)* kończy proces dodawania nowej maszyny wirtualnej.



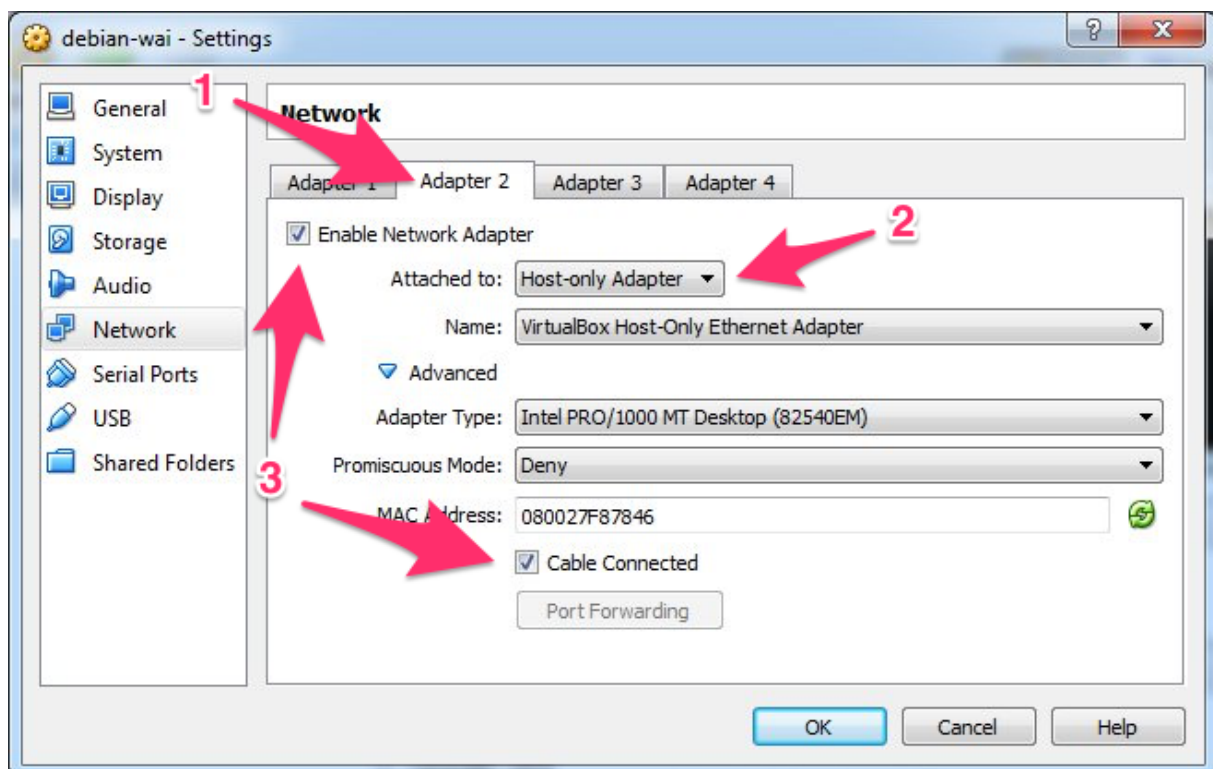
Krok 3 Przed pierwszym uruchomieniem konieczne jest wprowadzenie dodatkowych ustawień. Nowa maszyna widoczna jest w głównym oknie programu VirtualBox (1). Należy ją zaznaczyć, poprzez kliknięcie lewym przyciskiem myszy i wybrać przycisk *Settings* (2) na belce narzędziowej.



Krok 4 W oknie ustawień należy przejść do sekcji *Network* (1). Pozwala ona na konfigurację adapterów (kart) sieciowych. Pierwszy adapter umożliwi maszynie wirtualnej dostęp do Internetu. Karta z połączeniem typu NAT (2) udostępni systemowi gościa połączenie internetowe systemu hosta. Należy upewnić się, że pole *Enable Network Adapter* (3) jest zaznaczone.



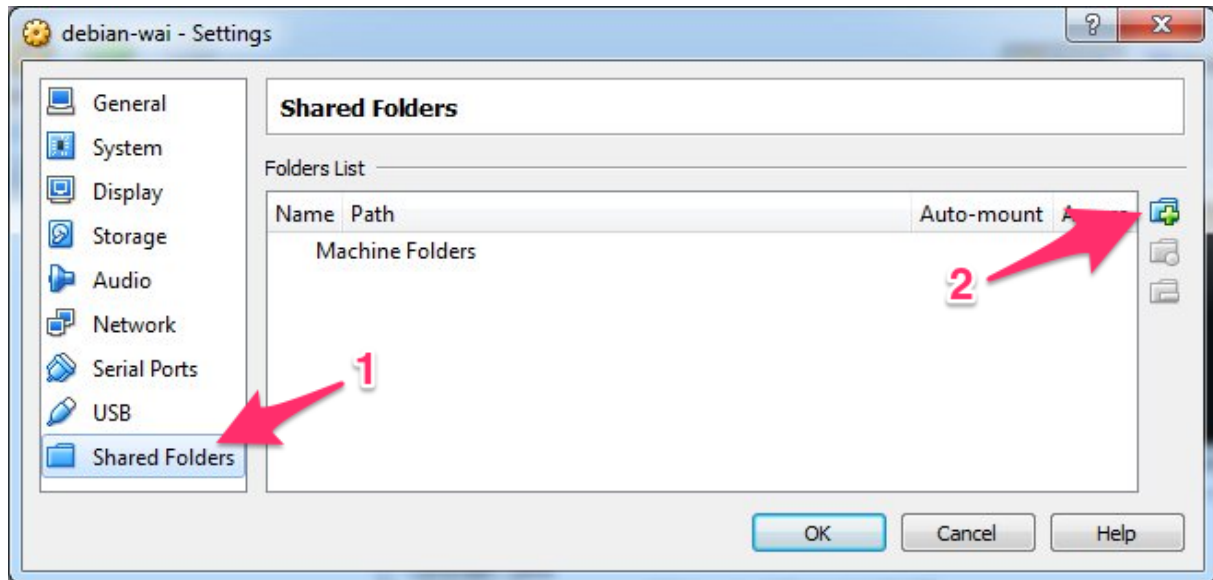
Krok 5 Następnie należy przejść na zakładkę *Adapter 2* (1). Drugi adapter sieciowy umożliwi dostęp do systemu gościa z poziomu systemu hosta. Dzięki temu możliwe będzie testowanie usług serwerowych, działających w systemie gościa, przy użyciu przeglądarki internetowej, uruchomionej w systemie hosta. Aby uzyskać taki efekt, konieczne jest wybranie połączenia *Host-only Adapter* (2). Na koniec należy upewnić się, że pola *Enable Network Adapter* i *Cable Connected* są zaznaczone. To kończy konfigurację adapterów sieciowych.



Krok 6 Ostatnim elementem jest konfiguracja katalogu współdzielonego pomiędzy systemem hosta i systemem gościa. Pozwoli to na łatwą wymianę plików. Wszystkie pliki umieszczone w katalogu współdzielonym będą natychmiast widoczne dla systemu w maszynie wirtualnej. Mechanizm ten

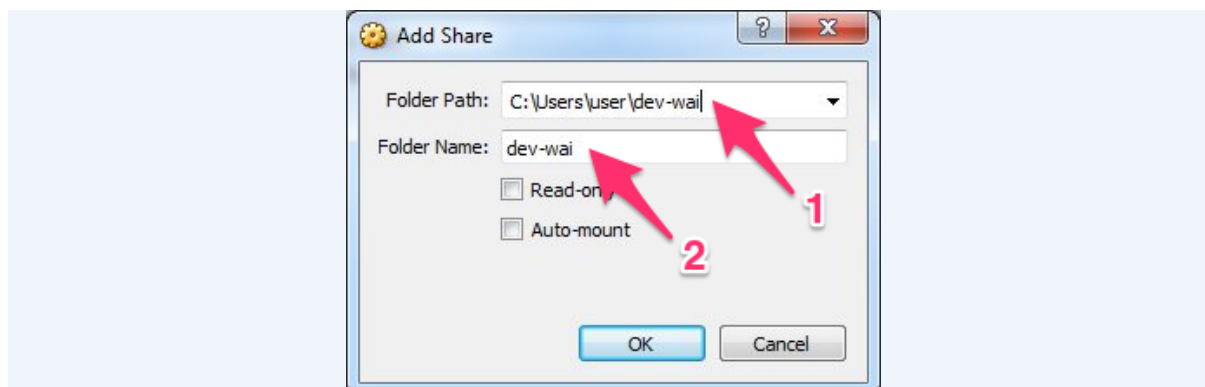
zostanie wykorzystany, aby umieścić w systemie gościa pliki aplikacji internetowej, która ma być przez niego hostowana.

W oknie ustawień należy przejść do sekcji *Shared Folders* (1) i wybrać przycisk po prawej stronie okna (2), aby dodać nowy katalog współdzielony.

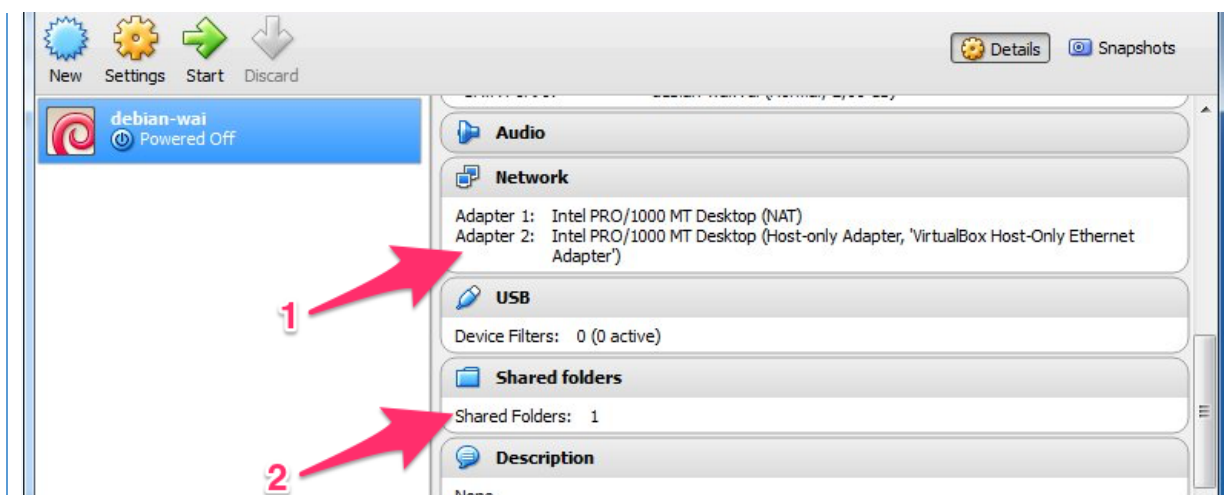


Krok 7 W nowym oknie należy podać ścieżkę do katalogu współdzielonego w systemie hosta (1). Nazwa katalogu na dysku może być dowolna. Dla systemu gościa zasób współdzielony będzie widoczny pod nazwą podaną w polu *Folder Name* (2), która musi mieć postać dev-wai. Obraz bazowy został tak skonfigurowany, aby automatycznie montować udział o takiej właśnie nazwie. Nazwa katalogu w systemie plików hosta może być inna niż nazwa udziału, przykładowo: ścieżka do katalogu współdzielonego w systemie hosta ma postać D:\projekty\wai-etap1, ale w polu *Folder Name* pozostaje nazwa dev-wai. Wprowadzone ustawienia należy zaakceptować przyciskiem *OK*, a następnie zamknąć główne okno ustawień maszyny, ponownie klikając *OK*.

Uwaga: w przypadku gdy rolę systemu hosta pełni system Windows nie należy umieszczać katalogu współdzielonego na *Pulpicie*.



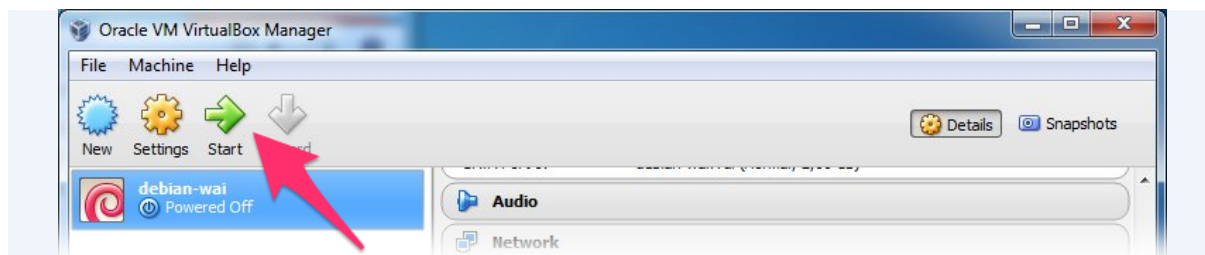
Krok 8 Na koniec należy zweryfikować, czy w głównym oknie programu VirtualBox widoczne są wprowadzone wcześniej ustawienia: dwa adaptory sieciowe (1) oraz jeden katalog współdzielony (2). Jeśli konfiguracja przebiegła pomyślnie, można przystąpić do uruchomienia maszyny wirtualnej.



2.3 Logowanie do systemu i weryfikacja ustawień

Na tym etapie nastąpi uruchomienie maszyny wirtualnej i logowanie do systemu gościa. Po pierwszym logowaniu należy zweryfikować, czy system gościa prawidłowo rozpoznał ustawienia maszyny wirtualnej, w szczególności konfigurację interfejsów sieciowych i katalogu współdzielonego. W tym celu konieczne jest wykonanie przedstawionych poniżej kroków.

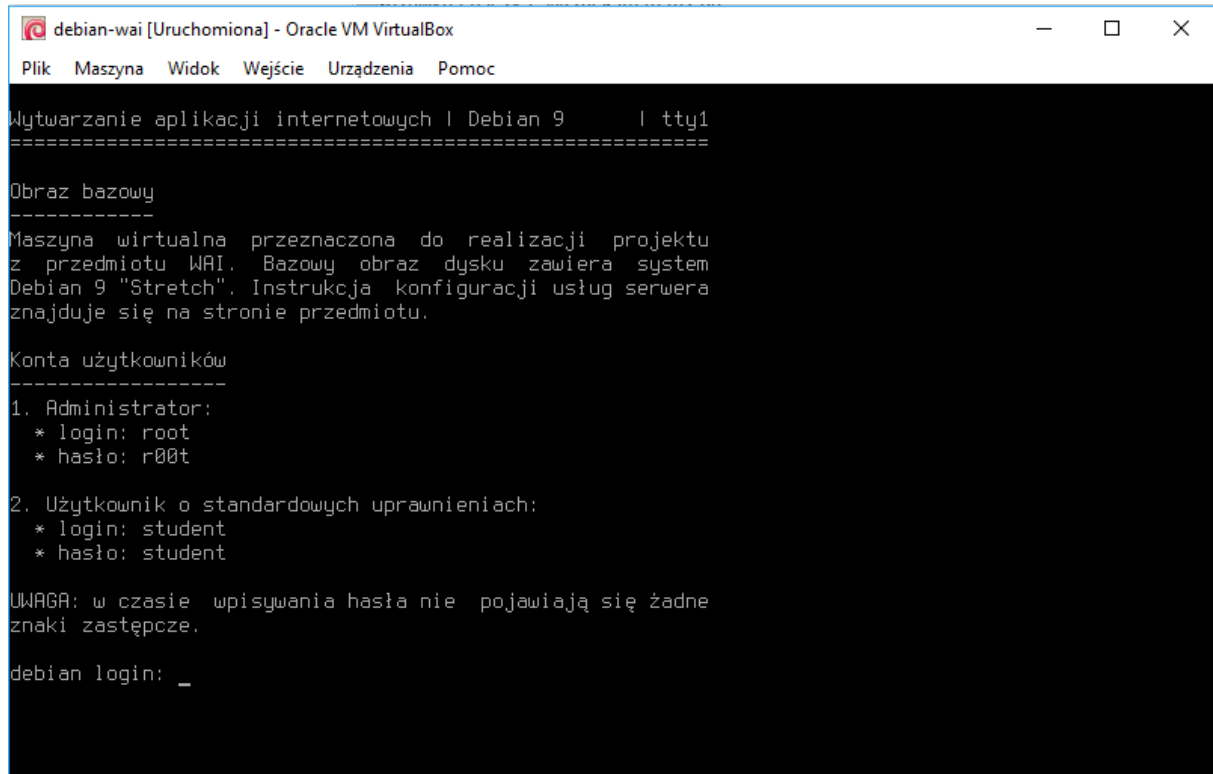
Krok 1 Uruchomienie maszyny wirtualnej odbywa się poprzez kliknięcie przycisku *Start* na belce narzędziowej.



Krok 2 Wyświetlone zostanie nowe okno, prezentujące wyjście wideo maszyny wirtualnej. Rozpoczęta zostanie sekwencja startowa maszyny, a następnie systemu operacyjnego gościa.



Krok 3 Jeśli uruchomienie przebiegnie pomyślnie, wyświetlony zostanie poniższy ekran powitalny systemu gościa. Bazowy obraz zawiera konta dwóch użytkowników, dane do logowania wyświetlone są na ekranie powitalnym.



```
debian-wai [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc

Wytwarzanie aplikacji internetowych I Debian 9      I tty1
=====

Obraz bazowy
-----
Maszyna wirtualna przeznaczona do realizacji projektu
z przedmiotu WAI. Bazowy obraz dysku zawiera system
Debian 9 "Stretch". Instrukcja konfiguracji usług serwera
znajduje się na stronie przedmiotu.

Konta użytkowników
-----
1. Administrator:
  * login: root
  * hasło: r00t

2. Użytkownik o standardowych uprawnieniach:
  * login: student
  * hasło: student

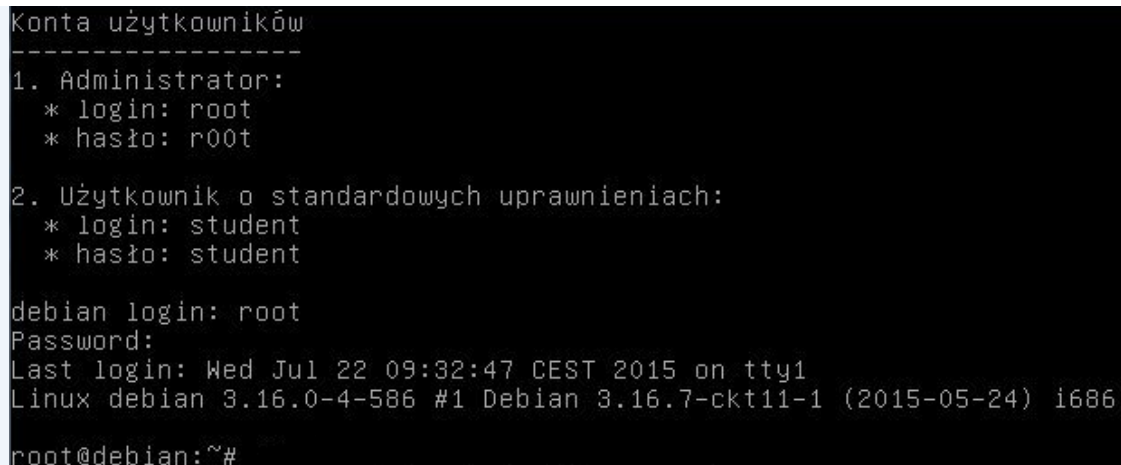
UWAGA: w czasie wpisywania hasła nie pojawiają się żadne
znaki zastępcze.

debian login: _
```

Krok 4 Należy zalogować się na konto administratora systemu:

- login: root
- hasło: r00t (*r-zero-zero-t*)

W czasie wpisywania hasła, na konsoli nie wyświetlają się żadne znaki zastępcze. Login i hasło należy zatwierdzić przyciskiem Enter. Jeśli logowanie przebiegnie pomyślnie, wyświetlony zostanie *znak zachęty* użytkownika root:



```
Konta użytkowników
-----
1. Administrator:
  * login: root
  * hasło: r00t

2. Użytkownik o standardowych uprawnieniach:
  * login: student
  * hasło: student

debian login: root
Password:
Last login: Wed Jul 22 09:32:47 CEST 2015 on tty1
Linux debian 3.16.0-4-586 #1 Debian 3.16.7-ckt11-1 (2015-05-24) i686

root@debian:~#
```

Znak zachęty wskazuje, że system oczekuje na wprowadzenie polecenia. Zwyczajowo znak zachęty w systemach z rodziny UNIX/Linux kończy się symbolem # dla użytkownika root (jak w przykładzie powyżej) lub symbolem \$ dla pozostałych użytkowników.



W dalszej części instrukcji bloki oznaczone symbolem # zawierają komendy do wpisania w wierszu poleceń systemu gościa po logowaniu na konto administratora. Bloki oznaczone symbolem \$ zawierają instrukcje, które nie wymagają uprawnień administracyjnych do działania (użytkownik student w systemie gościa). W blokach oznaczonych znakami zachęty każdy wiersz stanowi osobne polecenie, które należy zatwierdzić przyciskiem Enter. Kolejne polecenie może zostać wpisane dopiero, gdy w terminalu tekstowym ponownie wyświetlony zostanie znak zachęty.

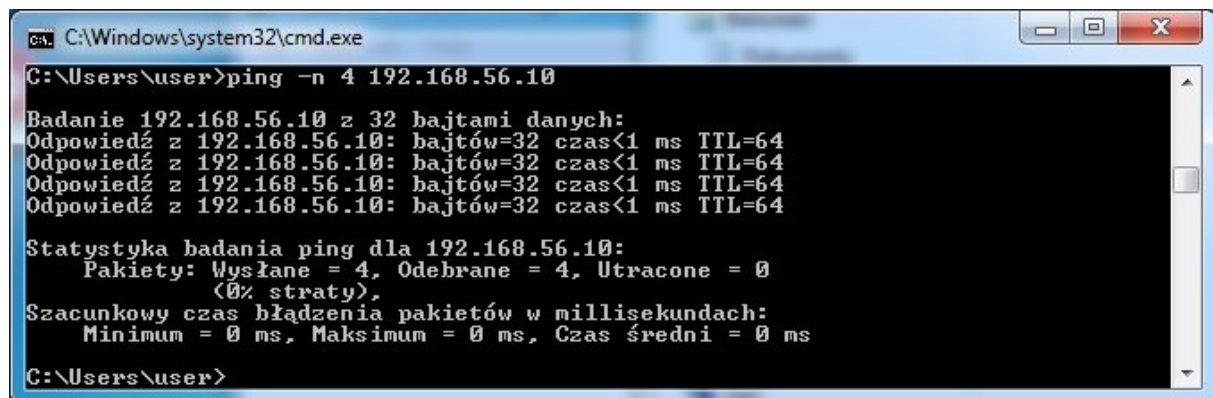
Krok 5 Należy zweryfikować, czy system gościa posiada dostęp do Internetu, wywołując polecenie ping z adresem wybranego serwera internetowego, np.:

ping -c 4 pg.edu.pl

```
root@debian:~# ping -c 4 pg.edu.pl
PING pg.edu.pl (153.19.40.170) 56(84) bytes of data.
64 bytes from pg.edu.pl (153.19.40.170): icmp_seq=1 ttl=56 time=10.9 ms
64 bytes from pg.edu.pl (153.19.40.170): icmp_seq=2 ttl=56 time=12.2 ms
64 bytes from pg.edu.pl (153.19.40.170): icmp_seq=3 ttl=56 time=11.9 ms
64 bytes from pg.edu.pl (153.19.40.170): icmp_seq=4 ttl=56 time=15.5 ms

--- pg.edu.pl ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 10.920/12.674/15.584/1.752 ms
root@debian:~#
```

Krok 6 Należy zweryfikować, czy system hosta ma dostęp do systemu gościa. W tym celu należy wywołać polecenie ping z poziomu systemu hosta podając adres IP systemu gościa. Obraz bazowy został skonfigurowany tak, aby wykorzystywać adres 192.168.56.10 dla adaptera sieciowego w trybie *host-only* (do komunikacji z systemem hosta). Jeśli rolę hosta pełni system Windows, należy wykorzystać aplikację wiersza poleceń cmd.exe:



```
C:\Windows\system32\cmd.exe
C:\Users\user>ping -n 4 192.168.56.10

Badanie 192.168.56.10 z 32 bajtami danych:
Odpowiedź z 192.168.56.10: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 192.168.56.10: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 192.168.56.10: bajtów=32 czas<1 ms TTL=64
Odpowiedź z 192.168.56.10: bajtów=32 czas<1 ms TTL=64

Statystyka badania ping dla 192.168.56.10:
    Pakiety: Wysłane = 4, Odebrane = 4, Utracone = 0
            (0% straty),
Szacunkowy czas błędzenia pakietów w milisekundach:
    Minimum = 0 ms, Maksimum = 0 ms, Czas średni = 0 ms

C:\Users\user>
```

Krok 7 Kolejnym krokiem jest weryfikacja, czy w systemie gościa został prawidłowo zamontowany katalog współdzielony. Służy do tego polecenie df (1), opcja -h (*human-readable*) spowoduje wyświetlenie rozmiarów systemów plików w postaci czytelnej dla człowieka:

df -h

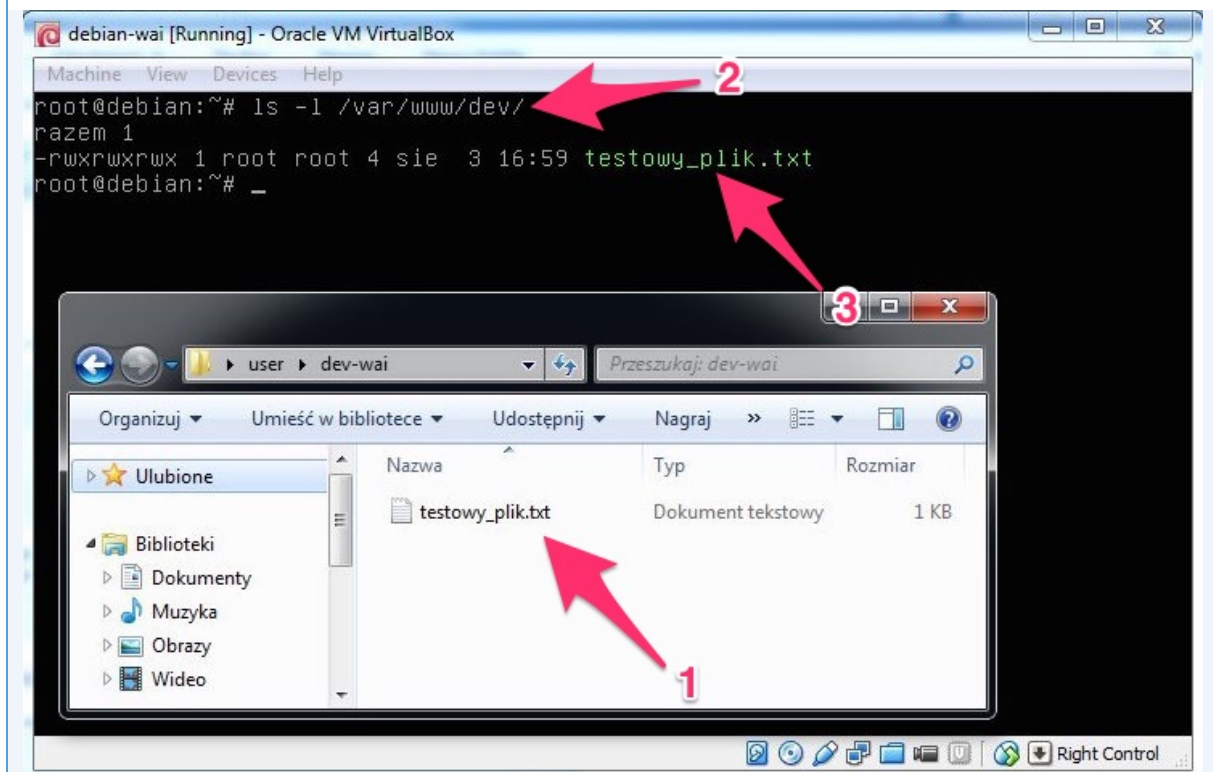
Wśród wyników polecenia powinna być widoczna pozycja /var/www/dev (2):

```
root@debian:~# df -h
System plików  rozm.  użyte  dośw.  %uż.  zamont.  na
/dev/sda1      2,0G   370M   1,6G   19%    /
udev           10M     0    10M    0%    /dev
tmpfs          101M    4,4M   96M    5%    /run
tmpfs          251M     0   251M    0%    /dev/shm
tmpfs          5,0M     0    5,0M    0%    /run/lock
tmpfs          251M     0   251M    0%    /sys/fs/cgroup
none           917G    86G   831G   10%    /var/www/dev
root@debian:~#
```

Krok 8 Następnie należy umieścić testowy plik w katalogu współdzielonym w systemie hosta (1) i przy użyciu polecenia ls (2):

`ls -l /var/www/dev/`

sprawdzić zawartość widoczną dla systemu gościa (3). Poniżej na pierwszym planie Eksplorator Windows (system host) pokazujący katalog współdzielony, a w tle maszyna wirtualna:



Jeśli wyniki uzyskiwane w systemie gościa były podobne do przedstawionych na przykładowych zrzutach ekranu, oznacza to, że konfiguracja maszyny wirtualnej jest poprawna. Możliwe jest zatem rozpoczęcie instalacji usług serwera aplikacji internetowych w systemie gościa. Zanim jednak to nastąpi, warto zapoznać się z podstawowymi poleceniami systemowymi, które będą wykorzystywane w dalszych etapach.

2.4 Podstawowe polecenia w systemie Debian GNU/Linux

Tabela poniżej zawiera zestawienie podstawowych poleceń, pozwalających na pracę w systemach z rodziny UNIX/Linux. Argumenty podane w nawiasach kwadratowych są opcjonalne. Szczegółowe

informacje na temat każdego z poleceń dostępne są w ich dokumentacji za pośrednictwem narzędzia `man` (ostatnia pozycja tabeli).

Polecenie	Opis
<code>ls [ścieżka]</code> <code>ls -l</code>	Wyświetla zawartość katalogu określonego ścieżką, podaną jako argument. Bez argumentu wyświetla zawartość bieżącego katalogu roboczego. Opcja <code>-l</code> zmienia format wyjścia na listę szczegółową.
<code>pwd</code>	Wyświetla pełną ścieżkę aktualnego katalogu roboczego.
<code>cd [ścieżka]</code> <code>cd /etc/apache2/</code>	Zmienia aktualny katalog roboczy na ten, który wskazuje ścieżka podana jako argument. W przypadku wywołania bez podanej ścieżki, przechodzi do katalogu domowego użytkownika. Ścieżka może być bezwzględna (zaczynająca się od znaku ukośnika: <code>/</code>) lub względna (w stosunku do aktualnego katalogu roboczego). W ścieżce mogą występować symbole zastępcze: ~ (tylda – katalog domowy), • (kropka – bieżący katalog), .. (dwie kropki – katalog nadrzędny).
<code>cp plik cel</code> <code>cp -r katalog cel</code>	Kopiuje plik do lokalizacji docelowej. Opcja <code>-r</code> umożliwia rekurencyjne skopiowanie katalogu i całej jego zawartości do lokalizacji docelowej.
<code>rm plik</code> <code>rm -r katalog</code>	Usuwa plik. Opcja <code>-r</code> umożliwia usunięcie katalogu wraz z całą jego zawartością.
<code>cat [plik]</code>	Wyświetla zawartość pliku podanego jako argument.
<code>nano [plik]</code> <code>nano httpd.conf</code>	Uruchamia prosty edytor tekstowy <code>nano</code> , otwierając plik podany jako argument. Podany plik nie musi istnieć w systemie plików – w takim przypadku zostanie on utworzony na dysku w chwili pierwszego zapisu. Bez nazwy pliku uruchamia edytor z pustym buforem. Skróty klawiszowe dla podstawowych operacji widoczne są w dolnej części edytora. Notacja <code>^O</code> oznacza skrót <code>Ctrl+O</code> (zapis zmian do pliku), <code>^X</code> – <code>Ctrl+X</code> (zamknięcie edytora), itd.
<code>vi [plik]</code>	Uruchamia edytor tekstowy <code>vi</code> otwierając plik podany jako argument (dla wytrawnych).
<code>apt-get ...</code> <code>apt-get install w3m</code>	<i>Specyficzne dla systemu Debian GNU/Linux i jego pochodnych.</i> Narzędzie do zarządzania pakietami oprogramowania. Umożliwia instalację pakietów z repozytoriów dystrybucji, automatycznie zarządza zależnościami.
<code>man polecenie</code> <code>man ls</code>	Wyświetla dokumentację polecenia, wskazanego w argumencie wywołania. Aby opuścić dokumentację należy wcisnąć klawisz <code>q</code> .
<code>shutdown -h now</code>	Powoduje wyłączenie systemu.

3. Konfiguracja usług serwera aplikacji internetowych

Dalsza część konfiguracji odbywa się w obrębie systemu gościa. Zainstalowany zostanie serwer HTTP Apache, baza danych MongoDB oraz interpreter języka PHP. Poszczególne komponenty zostaną skonfigurowane tak, aby utworzyć funkcjonalne środowisko do uruchamiania aplikacji internetowych. W czasie konfiguracji systemu przygotowane zostaną dwa hosty wirtualne. Pierwszy przeznaczony będzie do wytwarzania aplikacji. Drugi będzie imitować środowisko produkcyjne i przeznaczony jest do prezentacji projektu w czasie rozliczania kolejnych jego etapów.

Przed przystąpieniem do instalacji, konieczne jest przygotowanie menadżera pakietów poprzez zaktualizowanie informacji o zawartości repozytorium dystrybucji i zbudowanie lokalnej bazy dostępnych pakietów. Obraz bazowy nie zawiera tych informacji, co pozwoliło ograniczyć jego rozmiar. Przygotowanie menadżera pakietów wymaga wywołania polecenia:

```
# apt-get update
```

Należy zwrócić uwagę na ewentualne komunikaty błędów w wyjściu powyższego polecenia. Jeśli operacja przebiegła pomyślnie, możliwe jest rozpoczęcie instalacji pakietów.

3.1 Instalacja i konfiguracja serwera HTTP Apache

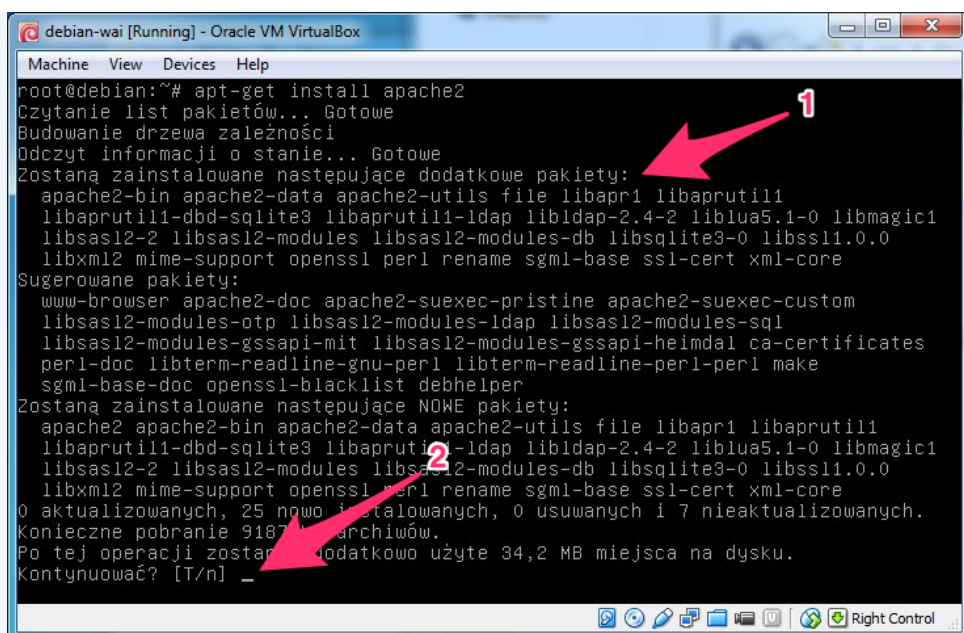
3.1.1 Instalacja serwera z repozytorium pakietów

Pierwszym komponentem do zainstalowania jest serwer Apache, który będzie przyjmował zapytania HTTP, kierowane do aplikacji internetowej. Instalacja przebiega z wykorzystaniem polecenia `apt-get`:

```
# apt-get install apache2
```

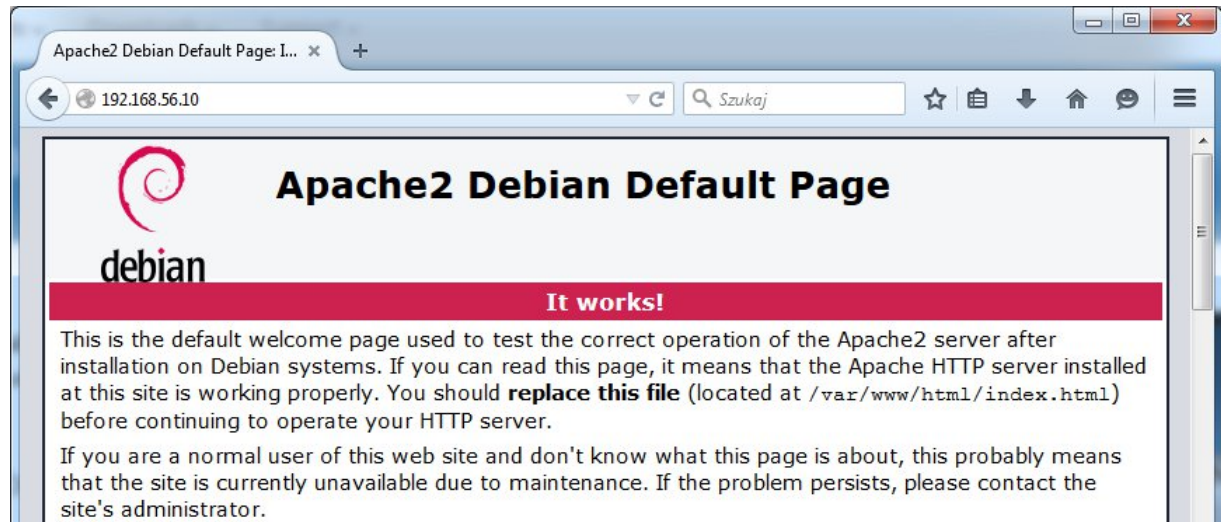
W przypadku gdy instalacja pakietu wymaga pobrania dodatkowych zależności (rys. 2, marker (1)), aplikacja `apt-get` poprosi o potwierdzenie, przed rozpoczęciem operacji. Na pytanie o potwierdzenie należy odpowiedzieć twierdząco – rys. 2, marker (2) – dopiero wtedy rozpocznie się instalacja pakietów.

Rys. 2 Instalacja oprogramowania z użyciem narzędzia `apt-get`.



Po zakończeniu instalacji pakietów, serwer Apache zostanie automatycznie uruchomiony i będzie serwował pliki znajdujące się w standardowym katalogu `/var/www/html`. Aby zweryfikować, czy serwer działa poprawnie, należy uruchomić przeglądarkę internetową w systemie hosta i wprowadzić adres systemu gościa: `192.168.56.10`. Widoczna powinna być strona powitalna serwera Apache, charakterystyczna dla systemu Debian GNU/Linux (rys. 2).

Rys. 2 | Strona powitalna serwera Apache w systemie Debian GNU/Linux.



3.1.2 Struktura plików konfiguracyjnych serwera

Konfiguracja serwera Apache znajduje się w katalogu `/etc/apache2/` (lokalizacja może się różnić w zależności od dystrybucji). Główny plik konfiguracyjny zwyczajowo nosi nazwę `apache2.conf` (jak w systemie Debian GNU/Linux) lub `httpd.conf`. Ze względu na dużą liczbę możliwych ustawień serwera, konfiguracja jest typowo rozdzielona pomiędzy kilka osobnych plików. Główny plik `apache2.conf/httpd.conf` zawiera podstawowe ustawienia i dołącza (przy użyciu dyrektywy `Include`) dalsze pliki, w których znajduje się konfiguracja poszczególnych modułów, hostów wirtualnych itd. Taki podział nie jest obowiązkowy – stanowi on konwencję, która ułatwia zarządzanie rozległą konfiguracją serwera. W taki właśnie sposób zorganizowana jest konfiguracja serwera Apache w systemie Debian GNU/Linux. Katalog `/etc/apache2` zawiera następujące elementy:

Element	Opis
<code>apache2.conf</code>	Główny plik konfiguracyjny, zawiera podstawowe ustawienia i dołącza pozostałe pliki konfiguracyjne.
<code>ports.conf</code>	Zawiera konfigurację portów, na których nasłuchuje serwer. Domyślnie wykorzystywany jest standardowy dla protokołu HTTP port 80.
<code>conf-available/</code>	Katalog zawierający dodatkowe ustawienia globalne. Pliki konfiguracyjne z tego katalogu <u>nie</u> są dołączane przez główny plik <code>apache2.conf</code> . Pliki te reprezentują konfiguracje dostępne (ang. <i>available</i>), ale nie necessarily aktywne (ang. <i>enabled</i> – patrz kolejna pozycja w tabeli). W katalogu tym można przechowywać opcjonalne konfiguracje, a następnie włączać je lub wyłączać w zależności od potrzeb. Włączenie pliku konfiguracyjnego powinno odbywać się poprzez utworzenie dowiązania symbolicznego do wybranego pliku w katalogu <code>conf-enabled</code> .
<code>conf-enabled/</code>	Katalog przeznaczony na dodatkowe ustawienia globalne. Pliki z tego katalogu są dołączane przez główny plik <code>apache2.conf</code> . Najczęściej będą się tu znajdować dowiązania symboliczne do plików zgromadzonych w katalogu <code>conf-available</code> .

mods-available/	Katalog zawierający dostępne konfiguracje modułów serwera. Pliki konfiguracyjne z tego katalogu <u>nie</u> są dołączane przez główny plik <code>apache2.conf</code> .
mods-enabled/	Katalog zawierający aktywne konfiguracje modułów serwera. Najczęściej będą się tu znajdować dowiązania symboliczne do plików zgromadzonych w katalogu <code>mods-available</code> .
sites-available/	Katalog zawierający konfiguracje dostępnych hostów wirtualnych. Pliki konfiguracyjne z tego katalogu <u>nie</u> są dołączane przez główny plik <code>apache2.conf</code> .
sites-enabled/	Katalog zawierający aktywne konfiguracje hostów wirtualnych. Najczęściej będą się tu znajdować dowiązania symboliczne do plików zgromadzonych w katalogu <code>sites-available</code> .

Zmiany konfiguracji wymagają jej ponownego załadowania. Odbywa się to poprzez restart serwera Apache:

```
# service apache2 restart
```

lub poprzez wykorzystanie operacji `reload`:

```
# service apache2 reload
```

Jeśli konfiguracja serwera jest błędna (np. zawiera nieprawidłowe wartości opcji konfiguracyjnych), powyższe polecenia wyświetlą stosowne komunikaty błędów.

Opisane powyżej pliki składają się na konfigurację globalną serwera, która jest wczytywana przy jego starcie. Możliwe jest również wykorzystanie konfiguracji lokalnej dla poszczególnych katalogów, z których serwowane są zasoby. Konfiguracje takie umieszczane są w plikach `.htaccess`. Ustawienia zawarte w tych plikach odnoszą się do katalogu, w którym dany plik się znajduje. Konfiguracja w pliku `.htaccess` może być modyfikowana bez restartowania serwera, gdyż pliki te są każdorazowo wczytywane, gdy serwowany jest zasób z danego katalogu. Stanowi to zaletę – zmiana konfiguracji nie wymaga przerwy w obsłudze żądań przez serwer – ale również wadę – konieczność parsowania pliku `.htaccess` odbija się na wydajności przetwarzania żądań.

Inną zaletą plików `.htaccess` jest to, że ich modyfikacja nie wymaga uprawnień do edycji plików zgromadzonych w katalogu `/etc/apache2/` (uprawnień administratora systemu), a jedynie praw dostępu do katalogu, z którego serwowane są pliki. Dzięki temu administrator serwera, hostującego wiele witryn, może umożliwić właścicielom poszczególnych aplikacji modyfikowanie pewnych aspektów ich działania, bez otwierania dostępu do konfiguracji globalnej (i bez konieczności restartowania serwera Apache). O tym, które elementy konfiguracji mogą być modyfikowane na poziomie plików `.htaccess`, decydują dyrektywy `AllowOverride`, stosowane w konfiguracji globalnej. Przykładowe zastosowania dyrektyw `AllowOverride` dla hostów wirtualnych znajdują się w dalszej części instrukcji.

3.1.3 Konfiguracja modułów serwera

Serwer HTTP Apache ma budowę modułową. Wiele jego funkcjonalności zaimplementowanych zostało w postaci modułów, ładowanych przy starcie serwera w zależności od aktualnej konfiguracji. Taka budowa ułatwia rozszerzanie możliwości serwera. Ponadto upraszcza ona konfigurację z punktu widzenia administratora: konieczne jest skonfigurowanie jedynie tych elementów, które są potrzebne do działania serwowanych aplikacji internetowych – pozostałe moduły nie zostaną uruchomione.

Jednym w często wykorzystywanych modułów serwera Apache jest `mod_rewrite`. Pozwala on na przepisywanie adresów z żądań HTTP na adresy, odpowiadające strukturze aplikacji wdrożonej na serwerze. Moduł `mod_rewrite` należy włączyć w konfiguracji serwera przy użyciu poniższego polecenia:

```
# a2enmod rewrite
```

Następnie konieczne jest ponowne uruchomienie serwera Apache (moduły ładowane są przy starcie):

```
# service apache2 restart
```

Dla aktywnych modułów, katalog `mods-enabled` zawiera dowiązania symboliczne, wskazujące na pliki konfiguracje, zgromadzone w katalogu `mods-available`.

3.1.4 Konfiguracja hostów wirtualnych

Na potrzeby realizacji projektu wykorzystany zostanie mechanizm hostów wirtualnych (ang. *virtual hosts*) serwera Apache. Przygotowane zostaną dwa hosty wirtualne, których zasoby zlokalizowane będą w katalogach:

- `/var/www/dev/` – host wytwórczy (deweloperski), przeznaczony do pracy nad aplikacją, katalog ten jest współdzielony z systemem hosta,
- `/var/www/prod/` – host imitujący środowisko produkcyjne, do wykorzystania w czasie rozliczania projektu.

Hosty będą posiadały różne ustawienia, co pozwoli na zobrazowanie różnic między środowiskiem wytwórczym i produkcyjnym oraz problemów, jakie mogą powstać w czasie wdrażania aplikacji na serwer docelowy. W środowisku produkcyjnym kluczowe jest bezpieczeństwo oraz wydajność przetwarzania zapytań, generowanych przez wielu użytkowników, którzy równocześnie korzystają z aplikacji internetowej. Środowisko deweloperskie powinno cechować się łatwością diagnozowania problemów, występujących w czasie pracy nad aplikacją, a w szczególności szybkim dostępem do komunikatów diagnostycznych. Wydajność w środowisku wytwórczym jest drugorzędna, ponieważ obsługuje ono tylko jednego użytkownika – samego dewelopera aplikacji. Z kolei komunikaty diagnostyczne w instancji produkcyjnej powinny być ukryte, ponieważ mogą one ujawnić informacje na temat aplikacji, które ułatwią atak potencjalnym napastnikom. Opisane wymagania będą miały odzwierciedlenie w konfiguracji hostów wirtualnych, która zostanie przedstawiona w dalszej części.

Konfiguracja dla instancji produkcyjnej zostanie przygotowana na bazie istniejącej już konfiguracji domyślnej witryny serwera Apache. Należy przejść do katalogu `/etc/apache2`:

```
# cd /etc/apache2
```

Następnie należy otworzyć do edycji plik `000-default.conf` (konfiguracja domyślnej witryny, dodana automatycznie w czasie instalacji serwera Apache) z katalogu `sites-enabled`, np.:

```
# nano sites-enabled/000-default.conf
```

Konieczne jest odnalezienie dyrektywy `DocumentRoot` (linia 12 pliku) i zmodyfikowanie jej jak poniżej oraz dodanie nowej sekcji `<Directory>` dla katalogu hosta produkcyjnego (pogrubioną czcionką oznaczono elementy nowe lub zmodyfikowane):

```
” DocumentRoot /var/www/prod/src/web
<Directory /var/www/prod/src/web>
    AllowOverride All
</Directory>
```

W dalszej części pliku należy zmodyfikować dyrektywy ErrorLog i CustomLog jak poniżej:

```
” ErrorLog "|/usr/bin/rotatelogs -t ${APACHE_LOG_DIR}/error.log 1M"
CustomLog "|/usr/bin/rotatelogs -t ${APACHE_LOG_DIR}/access.log 1M" combined
```

Skrót Ctrl+O zapisuje wprowadzone do pliku zmiany na dysk, a skrót Ctrl+X powoduje zamknięcie edytora nano. Po zmianach w konfiguracji konieczne jest jej ponowne załadowanie, przy użyciu polecenia:

```
# service apache2 reload
```

Następnie należy utworzyć katalog /var/www/prod/src/web – zasoby, które zostaną w nim umieszczone będą serwowane przez serwer Apache:

```
# mkdir -p /var/www/prod/src/web
```

Aby przetestować działanie instancji produkcyjnej konieczne jest dodanie pliku index.html o treści widocznej poniżej:

```
# cd /var/www/prod/src/web
echo "WAI - production" > index.html
```

Po tych operacjach należy przejść do przeglądarki internetowej uruchomionej w systemie hosta i odświeżyć stronę pod adresem 192.168.56.10. W przeglądarce powinna być widoczna treść wprowadzona w poprzednim kroku do pliku index.html.

Na potrzeby hosta wytwórczego otworzony zostanie dodatkowy port: 8080. Wymaga to powrotu do katalogu konfiguracyjnego serwera Apache:

```
# cd /etc/apache2
```

Konfiguracja portów serwera znajduje się w pliku ports.conf, który należy otworzyć do edycji i uzupełnić o linię:

```
” Listen 8080
```

W kolejnym kroku konieczne jest utworzenie pliku, który będzie zawierał konfigurację deweloperskiego hosta wirtualnego:

```
# nano sites-available/010-dev.conf
```

Należy go uzupełnić treścią przedstawioną poniżej (konfiguracja zostanie jeszcze uzupełniona, po zainstalowaniu interpretera języka PHP). Zamiast przepisywać treść pliku ręcznie do maszyny wirtualnej można utworzyć plik 010-dev.conf w katalogu współdzielonym i skopiować go do katalogu konfiguracyjnego serwera.

```
” <VirtualHost *:8080>
  DocumentRoot /var/www/dev/src/web
  ErrorLog "|/usr/bin/rotatelog -t ${APACHE_LOG_DIR}/dev_error.log 1M"
  CustomLog "|/usr/bin/rotatelog -t ${APACHE_LOG_DIR}/dev_access.log 1M" combined
</VirtualHost>
```

Aktywowanie tej konfiguracji wymaga utworzenia dowiązania symbolicznego (ang. *symbolic link*) w katalogu sites-enabled. Należy wywołać polecenia:

```
# cd sites-enabled
ln -s ../sites-available/010-dev.conf 010-dev.conf
cd ..
```

Należy załadować nową konfigurację przy użyciu polecenia:

```
# service apache2 reload
```

Następnie należy przejść do katalogu współdzielonego w systemie hosta, **utworzyć w nim podkatalog o nazwie src oraz kolejny podkatalog src/web**, a następnie przy użyciu ulubionego edytora dodać do podkatalogu web plik index.html o treści:

```
” WAI - development
```



Deweloperski host wirtualny serwuje wyłącznie pliki umieszczone w podkatalogu **src/web/** katalogu współdzielonego.

Po tych operacjach należy przejść do przeglądarki internetowej w systemie hosta i wpisać adres maszyny wirtualnej i numer portu środowiska wytwórczego: `http://192.168.56.10:8080`. W przeglądarce powinna być widoczna treść wprowadzona do pliku index.html w poprzednim kroku.

3.2 Instalacja i konfiguracja bazy danych MongoDB

Dane aplikacji będą przechowywane w bazie MongoDB. Jej instalacja w systemie Debian GNU/Linux odbywa się z wykorzystaniem narzędzia apt-get:

```
# apt-get install mongodb-server
```

Po instalacji bazy MongoDB konieczne jest skonfigurowanie mechanizmów kontroli dostępu. Domyślna konfiguracja daje pełne uprawnienia wszystkim klientom lokalnym (domyślnie baza nasłuchuje wyłącznie na interfejsie localhost) nie wymagając podania loginu i hasła. Nie jest to pożądane, gdyż każdy użytkownik mogący zalogować się do systemu uzyskuje pełną kontrolę nad bazą danych. Konieczne jest dodanie dwóch użytkowników:

- administratora uprawnień – zarządza uprawnieniami wszystkich innych użytkowników bazy, sam nie wykonuje żadnych operacji na zapisanych danych,
- użytkownika na potrzeby aplikacji internetowej – posiada uprawnienia do operowania na danych aplikacji, nie zarządza uprawnieniami innych użytkowników.

Dodanie administratora wymaga przygotowania pliku o nazwie admin.js i następującej treści:

```
” db.createUser( { user: "admin",
  pwd: "p@ssw0rd",
  roles: [ "userAdminAnyDatabase" ] } )
```

Plik może zostać przygotowany w systemie hosta i przeniesiony do maszyny wirtualnej poprzez katalog współdzielony. Umieszczony w pliku skrypt doda użytkownika o nazwie admin i hasło p@ssw0rd. Tak przygotowany skrypt należy wywołać przy użyciu poniższego polecenia, po przejściu do katalogu, w którym umieszczono plik admin.js:

```
# mongo admin admin.js
```

Polecenie doda konto administracyjne do bazy o nazwie admin, która stanowi globalną bazę administratorów. Przypisana rola userAdminAnyDatabase pozwoli administratorowi na zarządzanie uprawnieniami użytkowników we wszystkich innych bazach danych w obrębie danej instancji MongoDB. Fakt ten zostanie wykorzystany w dalszej części, w celu dodania użytkownika na potrzeby aplikacji internetowej, która będzie wykorzystywała odrębną bazę danych.

Aby wykorzystać mechanizm uwierzytelniania, należy zmodyfikować plik konfiguracyjny bazy MongoDB: /etc/mongodb.conf. Konieczne jest otwarcie tego pliku do edycji (np. w edytorze nano) i odkomentowanie (poprzez usunięcie znaku # z początku wiersza) linii zawierającej ustawienie:

```
” auth = true
```

Nowa konfiguracja zostanie wczytana w czasie restartu bazy MongoDB:

```
# service mongod restart
```

Po tych zmianach, wszystkie operacje na bazie danych będą wymagały uwierzytelnienia użytkownika o odpowiednich uprawnieniach. Następnym krokiem jest dodanie użytkownika na potrzeby aplikacji internetowej. Posłuży do tego kolejny skrypt (plik o nazwie wai.js):

```
” db.createUser( { user: "wai_web",  
                  pwd: "w@i_w3b",  
                  roles: [ "readWrite", "dbAdmin" ] } )
```

Wywołanie skryptu wymaga podania loginu i hasła dodanego wcześniej administratora uprawnień (uwaga: w poniższym poleceniu po opcji -p nie występuje znak spacji):

```
# mongo -u admin -pp@ssw0rd --authenticationDatabase admin wai wai.js
```

Po tych operacjach dodana zostanie baza danych o nazwie wai oraz użytkownik o loginie wai_web (z hasłem w@i_w3b), który posiada uprawnienia do operowania na danych we wspomnianej bazie. Login i hasło użytkownika wai_web należy wykorzystać w aplikacji, tworzonej w ramach projektu.

3.3 Instalacja i konfiguracja interpretera PHP

W kolejnym kroku zainstalowany zostanie interpreter języka PHP wraz ze standardowymi bibliotekami. Konieczne jest również zainstalowanie sterownika dla bazy MongoDB (nie wchodzi w skład biblioteki standardowej), który pozwoli na wykonywanie operacji na bazie danych z poziomu skryptów PHP. Zainstalowana zostanie również biblioteka PHP GD, która zawiera funkcje pozwalające na przetwarzanie plików graficznych. Należy wywołać poniższe polecenia:

```
# apt-get install php php-mongodb php-gd  
service apache2 restart
```


W czasie instalacji pakietu php, automatycznie aktywowany zostanie moduł mod_php serwera Apache. Dzięki temu serwer będzie wywoływał interpreter PHP dla skryptów w tym języku i serwował

wyniki ich wykonania do przeglądarki internetowej użytkownika. Aby zweryfikować, czy konfiguracja ta przebiegła pomyślnie należy utworzyć nowy plik o nazwie `info.php` w podkatalogu `web` katalogu współdzielonego z maszyną wirtualną i wypełnić go następującą treścią:

```
<?php  
phpinfo();
```

Po wpisaniu w przeglądarce w systemie hosta adresu `http://192.168.56.10:8080/info.php` powinna zostać wyświetlona strona z informacjami o konfiguracji interpretera PHP (rys. 3). Znajdują się tam m.in. informacje o wersji interpretera, systemie operacyjnym i konfiguracji zainstalowanych rozszerzeń i bibliotek. W szczególności strona informacyjna powinna zawierać sekcję związaną ze sterownikiem do bazy danych MongoDB (rys. 4).

Rys. 3 Strona informacyjna `phpinfo()`.

PHP Version 7.0.19-1	
	
System	Linux debian 4.9.0-3-686 #1 SMP Debian 4.9.30-2+deb9u3 (2017-08-06) i686
Build Date	May 11 2017 14:04:47
Server API	Apache 2.0 Handler

Rys. 4 Informacje na temat sterownika do bazy danych MongoDB.

mongodb	
MongoDB support	enabled
MongoDB extension version	1.2.3
MongoDB extension stability	stable
libbson bundled version	1.5.3
libmongoc bundled version	1.5.3
libmongoc SSL	enabled

Ostatni etap konfiguracji polega na dostosowaniu ustawień deweloperskiego hosta wirtualnego pod kątem wytwarzania aplikacji. Wykorzystane zostaną opcje ułatwiające diagnozowanie problemów, które występują w czasie pracy z kodem witryny. Konieczne jest zmodyfikowanie zawartości pliku konfiguracyjnego `/etc/apache2/sites-available/010-dev.conf` do postaci widocznej poniżej:

```
<VirtualHost *:8080>  
    DocumentRoot /var/www/dev/src/web  
    <Directory /var/www/dev/src/web>  
        AllowOverride All  
    </Directory>  
  
    ErrorLog "|/usr/bin/rotatelogs -t ${APACHE_LOG_DIR}/dev_error.log 1M"  
    CustomLog "|/usr/bin/rotatelogs -t ${APACHE_LOG_DIR}/dev_access.log 1M" combined  
  
    php_value display_errors On  
    php_value display_startup_errors On  
    php_value error_reporting -1  
    php_value html_errors On  
    php_value track_errors On  
    php_value output_buffering Off  
    php_value short_open_tag Off  
</VirtualHost>
```

Nowe ustawienia zostaną uwzględnione po ponownym wczytaniu konfiguracji:

```
# service apache2 reload
```

Na koniec zainstalowane zostanie narzędzie Composer do zarządzania zależnościami projektów wykorzystujących język PHP. Dodatkowy pakiet unzip jest opcjonalną zależnością pakietu composer:

```
# apt-get install composer unzip
```

3.4 Praca nad projektem

Maszyna przygotowana według powyższych instrukcji jest gotowa do realizacji projektu. Na etapie wytwarzania aplikacji można korzystać z hosta wirtualnego dev (<http://192.168.56.10:8080>), który serwuje pliki z katalogu współdzielonego z systemem hosta. Dzięki temu kod źródłowy projektu może być edytowany przy użyciu ulubionych narzędzi zainstalowanych w systemie hosta, a zmiany są natychmiast widoczne dla serwera uruchomionego w systemie gościa. W razie wystąpienia błędów w czasie działania skryptów PHP komunikaty diagnostyczne są wyświetlane bezpośrednio w oknie przeglądarki – zgodnie z ustawieniami w pliku `010-dev.conf`.

Na etapie rozliczania projektu konieczne jest użycie hosta wirtualnego z konfiguracją produkcyjną (<http://192.168.56.10>). Aby przenieść pliki aplikacji internetowej z katalogu współdzielonego do katalogu hosta produkcyjnego, należy wykonać poniższe polecenia:

```
# rm -r /var/www/prod/*  
cp -r /var/www/dev/* /var/www/prod/
```

Pierwsze z nich – `rm` – usuwa dotychczasową zawartość hosta produkcyjnego (uwaga: nie pyta o potwierdzenie, nie ma możliwości cofnięcia), drugie – `cp` – kopiuje pliki. Po wykonaniu tych poleceń, oba hosty wirtualne będą serwowały takie same zasoby.

Diagnozowanie problemów w środowisku produkcyjnym przebiega inaczej niż w przypadku hosta deweloperskiego. Produkcyjny serwer nie powinien wyświetlać komunikatów błędów wykonania skryptów PHP na wynikowej stronie, gdyż stałyby się one w ten sposób widoczne dla użytkowników końcowych. Zamiast tego komunikaty błędów zapisywane są w logach serwera, które można śledzić wykorzystując polecenie:

```
# tail -F /var/log/apache/access.log
```

Nowe komunikaty będą wyświetlane na bieżąco na konsoli maszyny wirtualnej. Aby przerwać śledzenie logu błędów, należy użyć kombinacji klawiszy `Ctrl+C`.

4. Przygotowanie maszyny wirtualnej do transportu

W celu rozliczenia kolejnych etapów projektu, konieczne jest przeniesienie przygotowanej maszyny wirtualnej na komputer laboratoryjny. Poniżej przedstawiono kroki pozwalające znacząco przyspieszyć ten proces.



Szybsze przygotowanie stanowiska oznacza więcej czasu na udzielenie odpowiedzi na pytania odnośnie projektu. Warto przetestować opisaną poniżej procedurę na komputerach laboratoryjnych przed terminem oddawania projektu (na jednych z wcześniejszych zajęć projektowych). Pozwoli to uniknąć ewentualnych problemów z uruchomieniem projektu w dniu jego rozliczania.

4.1 Rozmiar obrazu dysku maszyny wirtualnej

Przed przeniesieniem maszyny wirtualnej warto zadbać o zminimalizowanie rozmiaru obrazu dysku. Pozwoli to znacząco skrócić czas potrzebny na skopiowanie maszyny na pendrive lub zewnętrzny dysk, a następnie na dysk komputera laboratoryjnego. Jest to szczególnie istotne w przypadku urządzeń, które obsługują jedynie wolniejszy standard USB 2.0 (wiele pendrive'ów).

Aby usunąć zbędne pliki z dysku maszyny wirtualnej, należy wykorzystać skrypt pomocniczy `clean.sh`, który został umieszczony w katalogu domowym użytkownika root (ścieżka: `/root/`). Jego wywołanie w systemie gościa przebiega następująco:

```
# /root/clean.sh
```

Skrypt usuwa pliki tymczasowe i cache informacji o wykorzystywanych repozytoriach pakietów – przed późniejszymi operacjami na pakietach konieczne będzie ponowne wywołanie polecenia `apt-get update`.

4.2 Eksport maszyny wirtualnej wraz z ustawieniami

Plik VDI zawiera jedynie obraz dysku maszyny wirtualnej. Pozostałe ustawienia (rozmiar pamięci operacyjnej, konfiguracja kart sieciowych itd.) zapisane są w osobnym pliku o rozszerzeniu `vbox`. Program VirtualBox pozwala na wyeksportowanie zarówno obrazu dysku jak i ustawień maszyny wirtualnej do pojedynczego pliku w formacie OVF (ang. *Open Virtualization Format*). Wykorzystanie tej możliwości przyspieszy uruchomienie maszyny na komputerze laboratoryjnym, gdyż nie będzie konieczne ustawianie wszystkich opcji konfiguracyjnych maszyny – zamiast tego zostaną one załadowane z pliku OVF.



Eksport do formatu OVF jest zalecanym sposobem przenoszenia maszyny wirtualnej ze względu na łatwość procedury oraz automatyczne kompaktowanie i kompresję obrazu dysku.

Format OVF nie jest charakterystyczny dla programu VirtualBox. Jest to format uniwersalny, a jego obsługa jest zaimplementowana w wielu różnych menadżerach maszyn wirtualnych. Pozwala on na przenoszenie maszyn pomiędzy rozwiązaniami różnych dostawców. Przykładowo, maszyna wirtualna przygotowana na komputerze deweloperskim przy użyciu programu VirtualBox po wyeksportowaniu do formatu OVF może zostać przeniesiona na serwer produkcyjny wykorzystujący platformę VMware vSphere.

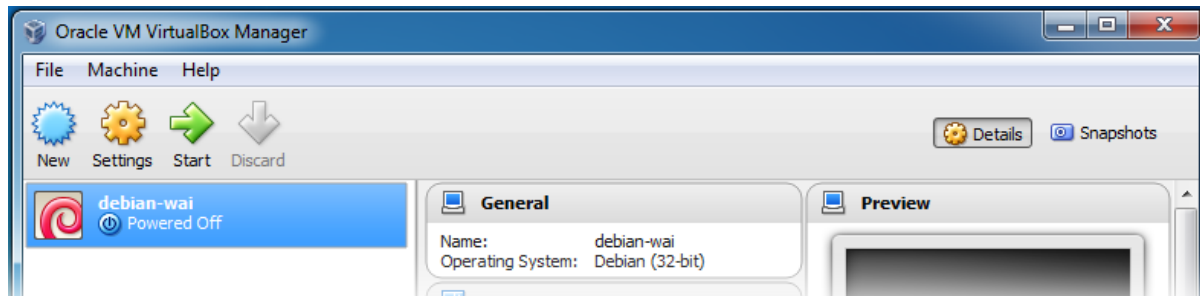
4.2.1 Eksport maszyny do pliku w formacie OVFA



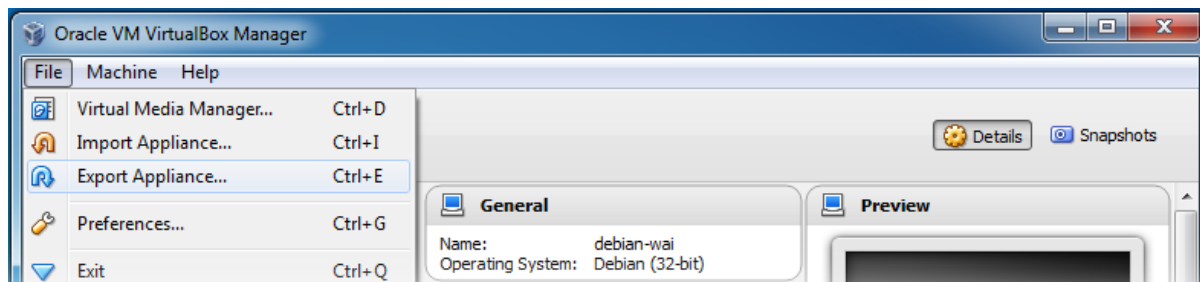
Wyeksportowana maszyna nie obejmuje zawartości katalogu współdzielonego. Pliki, które się w nim znajdują, pozostają zapisane w systemie plików systemu hosta. Pliki aplikacji internetowej należy przenieść do katalogu produkcyjnego hosta wirtualnego `/var/www/prod/` przed rozpoczęciem eksportu.

Przed rozpoczęciem eksportu należy wyłączyć maszynę wirtualną. Konieczne jest również usunięcie z konfiguracji maszyny katalogów współdzielonych, gdyż zawierają one ścieżki do lokalnych katalogów w systemie hosta. Po przeniesieniu wyeksportowanej maszyny na inny komputer, ścieżki te będą najpewniej niepoprawne. Po tych czynnościach wstępnych należy wykonać przedstawione poniżej kroki, w celu wyeksportowania maszyny.

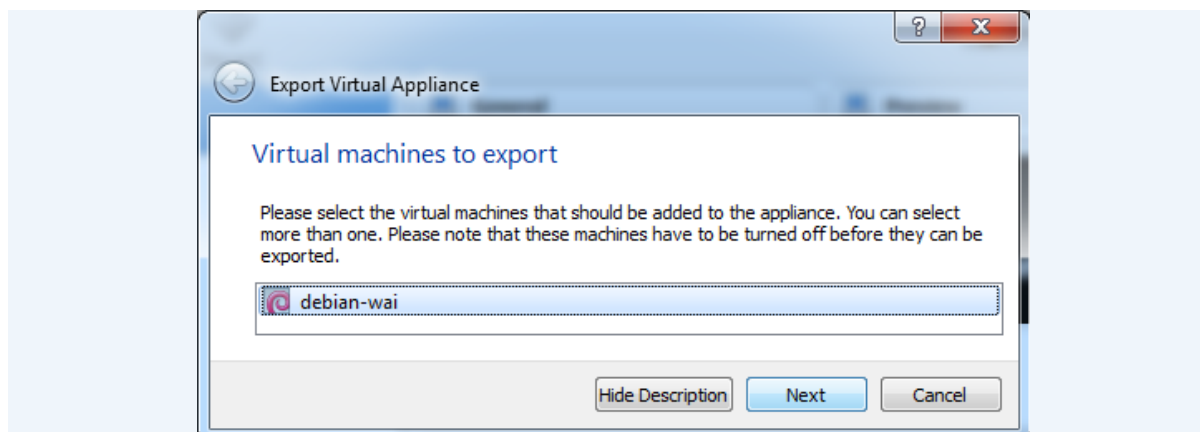
Krok 0 Główne okno programu VirtualBox z pojedynczą maszyną wirtualną do wyeksportowania:



Krok 1 Należy wybrać pozycję *Export Appliance* z menu *File*:

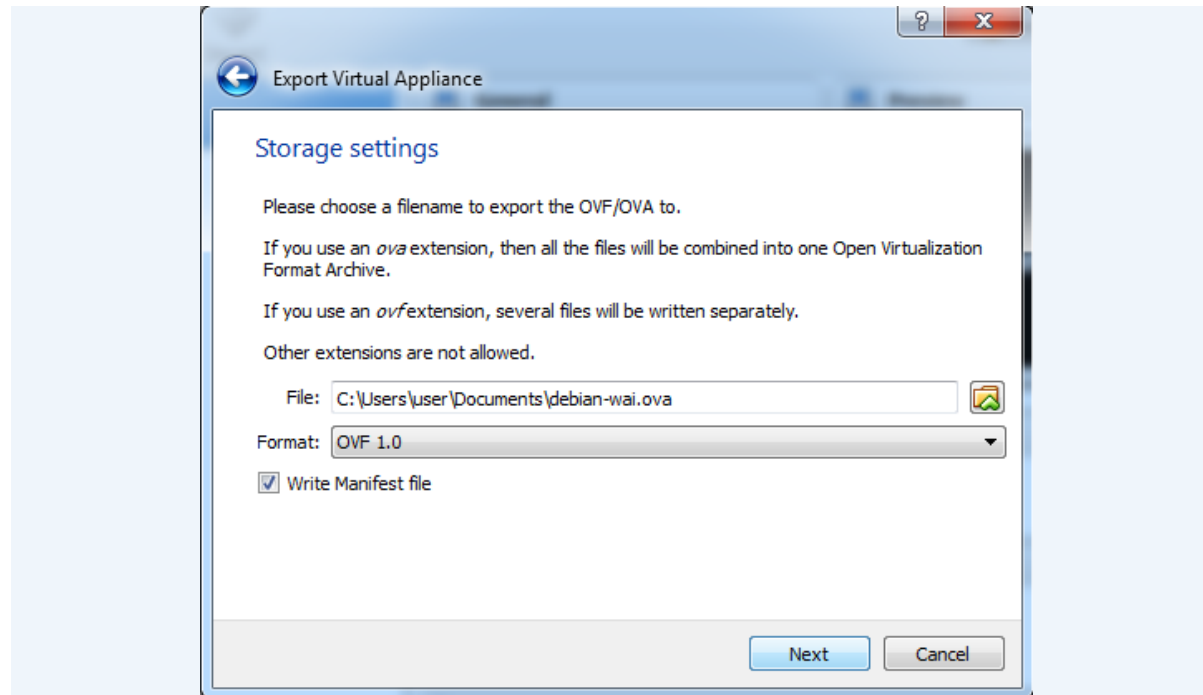


Krok 2 Należy wybrać z listy maszynę wirtualną, która ma zostać wyeksportowana i zatwierdzić wybór przyciskiem *Next*:

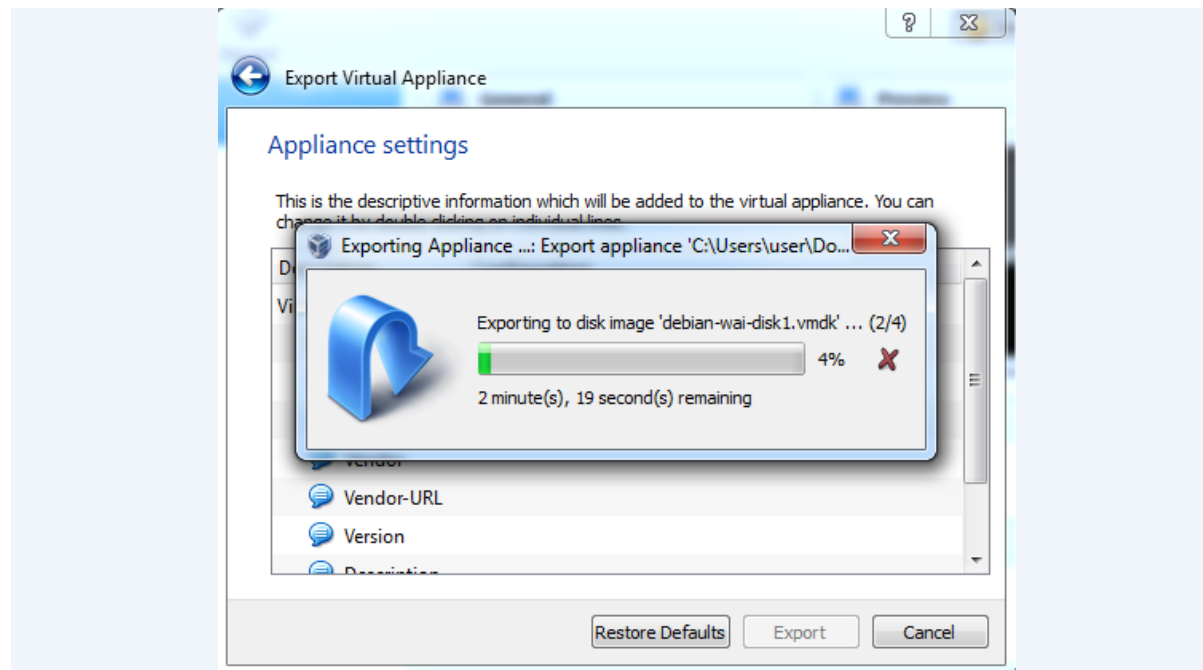


Krok 3 Konieczne jest określenie nazwy pliku, który będzie zawierał wyeksportowaną maszynę oraz lokalizacji na dysku, w której zostanie on zapisany. Wybranie rozszerzenia ova dla pliku spowoduje umieszczenie obrazu dysku i wszystkich ustawień maszyny w pojedynczym pliku OVFA (ang. *Open Virtualization Format Archive*). Zaznaczenie opcji *Write Manifest file* oznacza, że w czasie eksportu do pliku ova dołączone zostaną sumy kontrolne poszczególnych składników archiwum. Dzięki temu możliwe będzie zweryfikowanie poprawności obrazu w czasie późniejszego importu.

Jest to szczególnie istotne dla obrazów dystrybuowanych przez Internet, gdzie prawdopodobieństwo uszkodzenia pliku w czasie transferu jest znacznie większe niż przy przenoszeniu maszyny na pamięć USB.



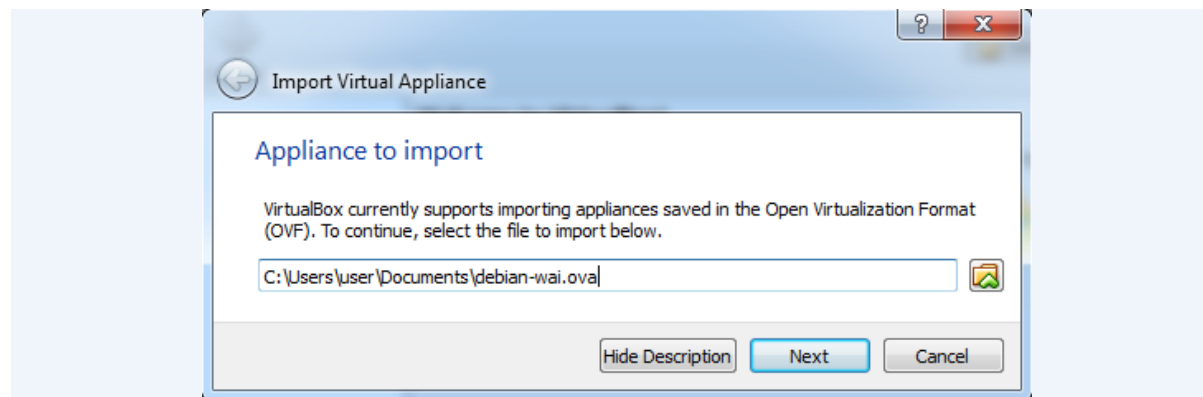
Krok 4 W kolejnym kroku widoczne jest podsumowanie informacji o maszynie wirtualnej. Wybranie przycisku *Export* spowoduje utworzenie pliku ova na dysku. Postęp procesu można śledzić w oknie dialogowym.



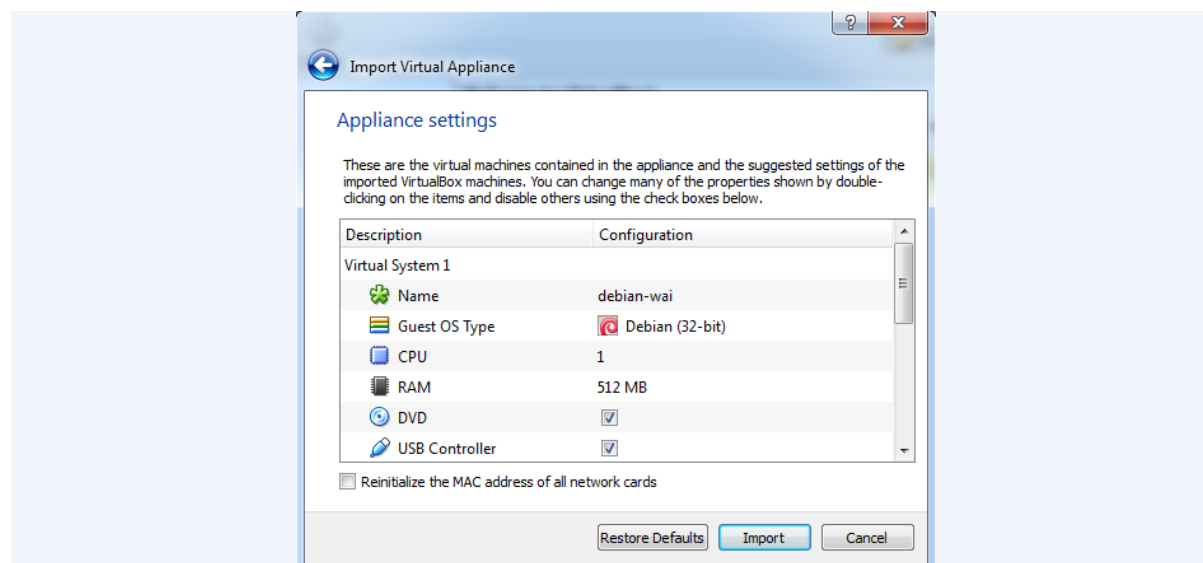
4.2.2 Import pliku OVFA

Przygotowany w poprzednim kroku plik w formacie OVFA można zaimportować na innym komputerze, zgodnie z opisem poniżej.

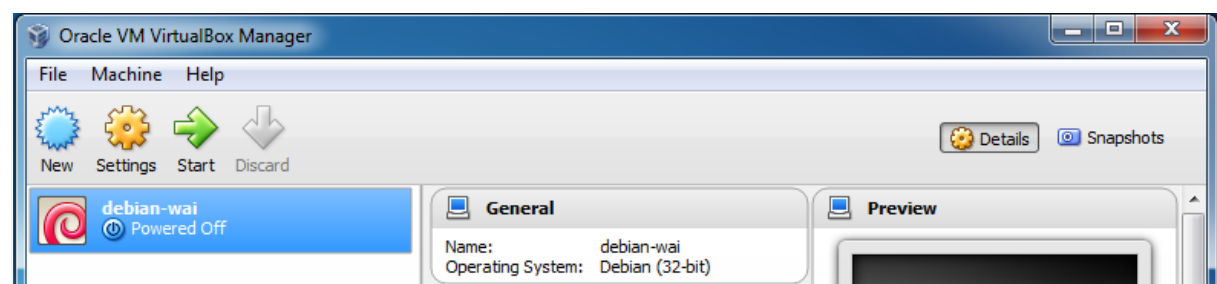
- Krok 1** Należy wybrać pozycję *Import Appliance* z menu *File*, a następnie wskazać lokalizację pliku o rozszerzeniu ova na dysku i zatwierdzić wybór przyciskiem *Next*.



- Krok 2** Wyświetlona zostanie konfiguracja maszyny wirtualnej. Widoczne ustawienia mogą zostać na tym etapie zmodyfikowane. Przykładowo, zmieniona może zostać nazwa maszyny, jeśli oryginalna jest zajęta przez inną maszynę wirtualną, istniejącą już w obrębie danego menadżera maszyn. Na zakończenie należy wybrać przycisk *Import*.



- Krok 3** Zaimportowana maszyna będzie widoczna w głównym oknie programu VirtualBox:



Dodatek A. Zalety maszyn wirtualnych i ich rola w branży IT

Na rynku istnieje wiele rozwiązań, umożliwiających pracę z maszynami wirtualnymi, zarówno darmowych jak i o charakterze komercyjnym. Wśród najpopularniejszych wymienić można: Oracle VirtualBox, VMware Workstation, VMware vSphere, Microsoft Hyper-V, Xen oraz QEMU. Rozwiązania te typowo umożliwiają dostosowanie podstawowych cech wirtualizowanych urządzeń, np.: rozmiar pamięci operacyjnej, rozmiar pamięci trwałej, liczba rdzeni procesora, liczba kart sieciowych i sposób ich podłączenia. Pozwala to na swobodną konfigurację maszyny wirtualnej, w celu dostosowania jej do planowanych zastosowań. Konfiguracja odbywa się za pośrednictwem aplikacji nadzorcy maszyn (ang. *hypervisor*), która odpowiada za ich uruchamianie i dostarczenie im wirtualnych komponentów.

Wykorzystywanie maszyn wirtualnych jest obecnie powszechne w branży IT, co wynika z ich wielu zalet. Po pierwsze, maszyny wirtualne pozwalają na separację różnych środowisk wykonawczych – każda z maszyn działa w sposób niezależny od innych. Każda może być uruchamiana, restartowana bądź przenoszona niezależnie od pozostałych. Dzięki temu, uruchamiając poszczególne aplikacje internetowe na różnych maszynach wirtualnych, możliwe jest wykonywanie operacji administracyjnych dla jednej z nich bez zakłócania dostępności pozostałych. Jest to szczególnie istotne w przypadku aplikacji obsługujących duży ruch i witryn typu e-commerce, gdzie nawet kilkuminutowa niedostępność usług przekłada się na realne straty finansowe.

Separacja środowisk wykonawczych ułatwia również konfigurację każdego z nich z osobna. Historycznie typowe środowiska aplikacji internetowych opisywane były akronimem LAMP, który wskazywał składające się na nie komponenty: Linux + Apache + MySQL + PHP (system operacyjny + serwer HTTP + relacyjna baza danych + język skryptowy). Taki *stos technologiczny* (ang. *technology stack*) stanowił wspólny mianownik dla wielu aplikacji i pozwalał na ich równoczesne hostowanie w obrębie pojedynczej maszyny.

Współczesne aplikacje internetowe często wymagają dodatkowych komponentów, np.: bazy danych typu NoSQL, bazy z wyszukiwaniem pełnotekstowym, brokera komunikatów, brokera zadań wsadowych, mechanizmu cachowania, serwera raportów itd. Każda aplikacja może wymagać innego stosu technologicznego. Próba konfiguracji wszystkich wymaganych komponentów dla kilku różnych aplikacji w obrębie jednego systemu operacyjnego może zakończyć się konfliktami w dostępie do zasobów pomiędzy różnymi usługami (np. numery portów). W skrajnych przypadkach wystąpić mogą trudne do wykrycia i zdiagnozowania problemy, wynikające z niekompatybilności między poszczególnymi komponentami.

Wykorzystanie maszyn wirtualnych rozwiązuje powyższe problemy dzięki separacji środowisk. Każda aplikacja może zostać uruchomiona w obrębie własnej maszyny wirtualnej, w której zainstalowane zostały wyłącznie usługi konieczne do działania witryny. Warto zwrócić uwagę, że stos technologiczny poszczególnych aplikacji może różnić się już na poziomie systemu operacyjnego. Maszyny wirtualne ponownie pokazują swoją przydatność, umożliwiając uruchomienie wielu różnych systemów operacyjnych na jednej maszynie fizycznej, niezależnie od tego jaki system pełni rolę hosta.

Kolejny aspekt związany jest z bezpieczeństwem. Udana atak na aplikację, działającą w obrębie jednej maszyny wirtualnej, nie ma wpływu na działanie pozostałych maszyn. Dodatkowo mniejsza liczba działających usług w każdej z maszyn ułatwia zarządzanie bezpieczeństwem. Mniejsza liczba usług oznacza mniej potencjalnych luk bezpieczeństwa, z których mogą skorzystać napastnicy.

Kolejna istotna kwestia to łatwość podziału zasobów serwera fizycznego. Każda maszyna wirtualna otrzymuje pewną część zasobów fizycznych (pamięci operacyjnej, mocy procesora, pamięci trwałej

itd. – przykład przygotowania takiej konfiguracji w dalszej części dokumentu). Przydział ten jest kontrolowany przez nadzorcę maszyn wirtualnych (*hypervisor*). Aplikacje uruchomione w obrębie maszyny nie mogą przekroczyć narzucanych przez nadzorcę limitów. Dzięki temu pojedyncza maszyna wirtualna z wadliwą aplikacją (np. posiadającą wycieki pamięci), nie zakłóca prawidłowego działania pozostałych.

Maszyny wirtualne mogą być łatwo powielane i przenoszone między różnymi maszynami fizycznymi w celu ich uruchomienia z użyciem dodatkowych zasobów lub uruchomienia kilku równocześnie działających instancji. Ułatwia to skalowanie aplikacji w poziomie i pozwala szybko reagować na zmienne obciążenie, generowane przez użytkowników. Z tego względu maszyny wirtualne są podstawą wszystkich usług w *chmurze* (ang. *cloud computing*).

Omówione dotychczas zalety przedstawiony były w kontekście środowisk produkcyjnych aplikacji. Wiele z tych zalet ma zastosowanie również na wcześniejszych etapach cyklu życia aplikacji: w środowiskach wytwórczych i testowych. Zastosowanie maszyn wirtualnych pozwala zbliżyć te środowiska do środowiska docelowego, zmniejszając tym samym ryzyko niespodziewanych problemów w czasie wdrażania instancji produkcyjnej.

Wirtualizacja umożliwia również zasymulowanie na komputerze dewelopera złożonych konfiguracji, uwzględniających przykładowo wydzielony serwer bazodanowy, serwer zasobów statycznych i zwielokrotnione instancje serwera aplikacji dla równoważenia obciążenia. Pozwala to – jeszcze przed wdrożeniem produkcyjnym – na zidentyfikowanie problemów, które są niemożliwe do wykrycia, gdy wszystkie komponenty są uruchomione w obrębie jednego systemu.

Wykorzystanie maszyn wirtualnych zwalnia dewelopera z konieczności instalacji całego stosu technologicznego dla aplikacji w systemie hosta. Jest to szczególnie istotne dla deweloperów uczestniczących w więcej niż jednym projekcie i niesie takie same korzyści jak w środowisku produkcyjnym: ułatwia konfigurację i zapobiega konfliktom między różnymi usługami. Pozwala to również na proste zidentyfikowanie zależności projektu, które wprost wynikają z konfiguracji maszyny wirtualnej. Dodatkowo wdrażanie nowych deweloperów do projektu staje się łatwiejsze, ponieważ mogą oni otrzymać gotową maszynę wirtualną, zawierającą kompletne środowisko wykonawcze. Skracą to czas potrzebny do rozpoczęcia pracy nad kodem aplikacji.

Maszyny wirtualne stają się przydatne również w czasie testowania nowych rozwiązań, takich jak zmiana bądź aktualizacja serwera HTTP, dodanie nowych modułów, czy wprowadzenie mechanizmu cachowania danych. Mechanizm migawek (ang. *snapshots*) pozwala łatwo przywrócić maszynę wirtualną do stanu, w którym działała ona poprawnie, w sytuacji gdy nowa konfiguracja zakończy się niepowodzeniem. Dodatkowo, nieudane zmiany w obrębie maszyny wirtualnej nie destabilizują działania systemu hosta.

Nadzorca maszyn wirtualnych wprowadza dodatkowy poziom abstrakcji ponad maszyną fizyczną dla systemu operacyjnego gościa. Może się to odbić na wydajności maszyn wirtualnych. Jednak ze względu na liczne zalety wirtualizacji opracowane zostało wiele rozwiązań, mających minimalizować jej negatywny wpływ na wydajność. Współczesne procesory posiadają rozszerzenia, tworzone z myślą o uruchamianiu na nich maszyn wirtualnych (np. Intel® VT-x, AMD-V™), oferując tym samym sprzętowe wsparcie wirtualizacji dla nadzorców maszyn. Obejmuje ono aspekty związane z samym procesorem (umożliwiając nawet zagnieżdżoną wirtualizację), dostępem do pamięci operacyjnej (zagnieżdżone stronicowanie), obsługą operacji wejścia/wyjścia i wykorzystaniem karty graficznej w systemie gościa. Dzięki temu maszyny wirtualne mogą osiągać wydajność zbliżoną do natywnego działania na maszynie fizycznej.

Wirtualizacja pełni obecnie kluczową rolę zarówno w środowiskach produkcyjnych, jak i wytwórczych aplikacji. Istnieje wiele jej odmian i rozwiązań pośrednich, jednak ich szczegółowe omówienie wykracza poza zakres niniejszego dokumentu, który pełni rolę wprowadzenia do tej tematyki. Szersze omówienie zagadnień związanych z maszynami wirtualnymi będzie tematem innych przedmiotów w toku studiów.