


SPRAWOZDANIE NR 1			
Nazwa ćwiczenia	Etap 1: Przygotowanie systemu operacyjnego do projektu		 POLITECHNIKA BYDGOSKA Wydział Telekomunikacji, Informatyki i Elektrotechniki
Przedmiot	Narzędzia komputerowe: Projekt		
Student grupa	Jakub Weiss, Dorian Zasada, Maciej Kwiatkowski, Julian Frischke – Zespół nr. 1		
Data ćwiczeń			Data oddania sprawozdania

1. Cel Ćwiczenia

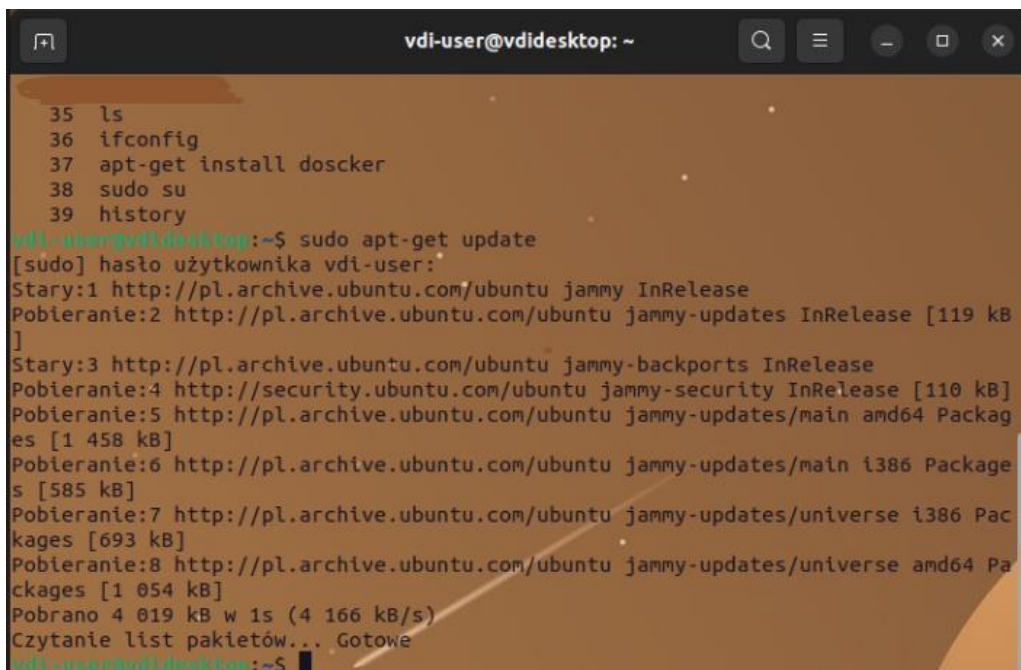
Celem tego etapu jest przygotowanie systemu operacyjnego do realizacji projektu. Na maszynie wirtualnej (lub innym preferowanym rozwiązaniu) musi być zainstalowany system operacyjny Ubuntu wraz z narzędziami docker i docker-compose. Ponadto na maszynach wirtualnych musi być dostępne połączenie ssh. W tym etapie należy skonfigurować 4 maszyny.

2. Przebieg

2.1. Zadanie 1: Skonfigurowanie 4 maszyn z systemem ubuntu & instalacja protokołu komunikacyjnego SSH.

Do stworzenia 4 maszyn wykorzystaliśmy usługę, którą udostępnia nam uczelnia – VDI. VDI to infrastruktura, która pozwala nam na tworzenie oraz łączenie się do tych maszyn z prawie każdego urządzenia.

Po skonfigurowaniu maszyny zaktualizowaliśmy ją – aby nie zwiększać szansy na wystąpienie potencjalnego błędu. Do wykonania tego użyliśmy polecenia „sudo apt-get update”. Polecenie pozwala na aktualizację wszystkich niezbędnych pakietów.



```

vdi-user@vdidesktop: ~
35 ls
36 ifconfig
37 apt-get install doscker
38 sudo su
39 history
vdi-user@vdidesktop:~$ sudo apt-get update
[sudo] hasło użytkownika vdi-user:
Stary:1 http://pl.archive.ubuntu.com/ubuntu jammy InRelease
Pobieranie:2 http://pl.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Stary:3 http://pl.archive.ubuntu.com/ubuntu jammy-backports InRelease
Pobieranie:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Pobieranie:5 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1 458 kB]
Pobieranie:6 http://pl.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [585 kB]
Pobieranie:7 http://pl.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [693 kB]
Pobieranie:8 http://pl.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1 054 kB]
Pobrano 4 019 kB w 1s (4 166 kB/s)
Czytanie list pakietów... Gotowe
vdi-user@vdidesktop:~$

```

Po zakończeniu aktualizacji, nadszedł czas na zainstalowanie protokołu SSH, który był niezbędny, aby połączyć się z danymi maszynami. (SSH to specjalny protokół, który pozwala na zdalne oraz szyfrowane połączenie z serwerem przy pomocy połączenia terminalowego) komunikacyjny. Do wykonania tego użyliśmy polecenia „sudo apt-get install ssh”.

```
vd1-user@vdi desktop: ~
Pobrano 4 019 kB w 1s (4 166 kB/s)
Czytanie list pakietów... Gotowe
vd1-user@vdi desktop:~$ sudo apt-get install ssh
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
ssh is already the newest version (1:8.9p1-3ubuntu0.6).
Następujące pakiety zostały zainstalowane automatycznie i nie są już więcej wymagane:
 chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
 intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
 libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
 libcodec2-1.0 libdavid5 libflite1 libgme0 libgsm1
 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm15 libmfx1
 libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libstrat0-0
 libsrtp1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0
 libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1
 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvb1-common libzvb10
 mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us va-driver-all
 vdpau-driver-all
Aby je usunąć należy użyć "sudo apt autoremove".
0 aktualizowanych, 0 nowo instalowanych, 0 usuwanych i 228 nieaktualizowanych.
vd1-user@vdi desktop:~$
```

Zainstalowaliśmy SSH wcześniej, tutaj prezentujemy tylko przykład użycia komendy jeśli takowy protokół był już wcześniej zainstalowany.

Następnie przenieśliśmy się na konto administracyjne „root”, aby mieć dostęp do wszystkich niezbędnych nam rzeczy i przeszliśmy do katalogu /etc/ssh aby edytować plik konfiguracyjny SSH.

```
root@vdi desktop:/etc/ssh# ls
moduli          sshd_config.d      ssh_host_ed25519_key.pub
ssh_config      ssh_host_ecdsa_key  ssh_host_rsa_key
sshd_config.d   ssh_host_ecdsa_key.pub  ssh_host_rsa_key.pub
sshd_config     ssh_host_ed25519_key  ssh_import_id
root@vdi desktop:/etc/ssh# nano sshd_config
```

Komendą „nano” otwieramy plik konfiguracyjny w edytorze tekstu, aby móc ustalić port dla protokołu SSH (domyślnie, ustawiamy port 22, ponieważ jest on publiczny)

```
GNU nano 6.2          sshd_config *
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
[Wczytano 122 linie]
^G Pomoc  ^O Zapisz  ^W Wyszukaj ^K Wytnij  ^T Wykonaj ^C Lokalizacja
^X Wyidź  ^R Wczyt.plik ^\ Zastap  ^U Wklej   ^J Wyjustuj ^_ Do linii
```

Usuujemy „#” przy słowie „Port”, aby ustawić port na 22.

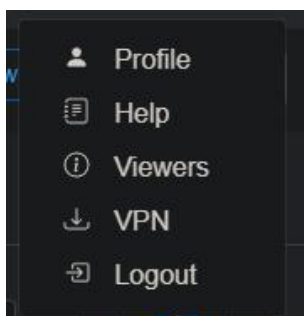
Następnie instalujemy narzędzie net-tools pozwalające sprawdzić nam adres ip karty sieciowej. Poleceniem „ifconfig” wywołujemy wszystkie informacje na temat kart sieciowych. Tym sposobem dowiadujemy się o adresie naszej maszyny: 10.2.140.195. (oczywiście jest to jedna z 4 maszyn, tak samo działaliśmy przy 3 pozostałych).

```
ens4: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1366
    inet 10.2.140.195 netmask 255.255.0.0 broadcast 10.2.255.255
    inet6 fe80::57b1:6c6e:dceb:9520 prefixlen 64 scopeid 0x20<link>
    ether 52:54:00:18:89:c1 txqueuelen 1000 (Ethernet)
    RX packets 18021 bytes 787282 (787.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 21973 bytes 975688 (975.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

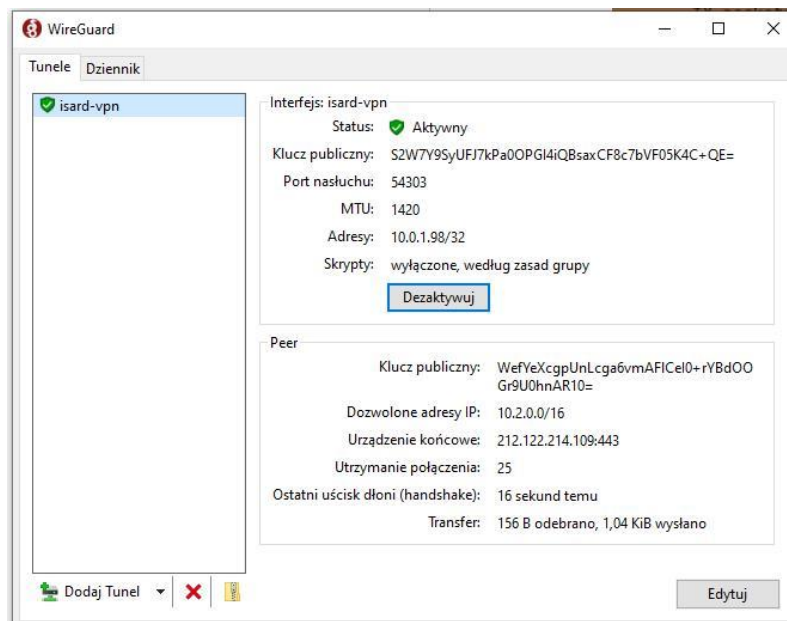
2.2. Zadanie 2: Konfiguracja VPN'a

Na ten moment schodzimy z maszyny i zajmujemy się naszym urządzeniem, z którego będziemy się łączyć do maszyny. Instalujemy VPN'a WireGuard, który pozwoli nam na utworzenie tunelu do naszych maszyn.

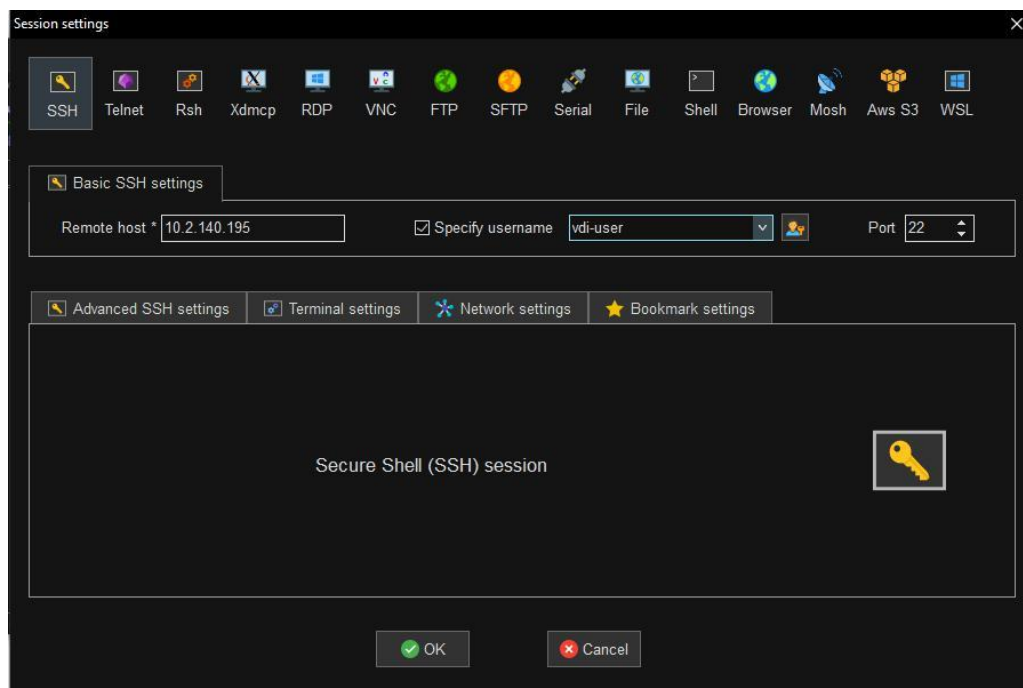
Po zainstalowaniu VPN'a, przechodzimy na stronę VDI a następnie klikamy w nasze imię i nazwisko, aby pobrać plik do naszego vpn'a – tak zwany tunel.



Importujemy plik do programu, powinno to wyglądać mniej więcej w ten sposób:

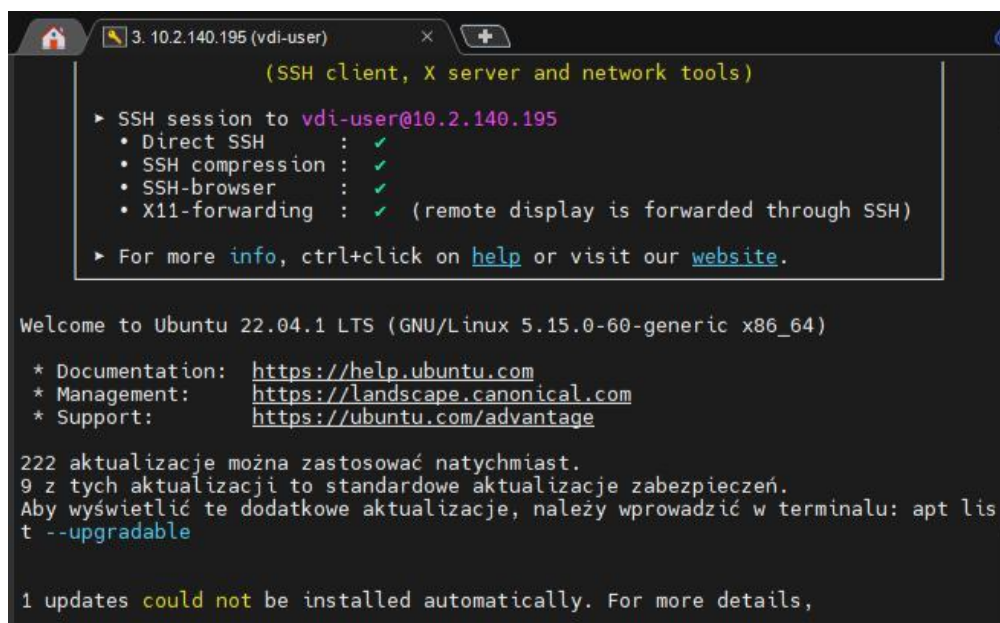


Teraz zostało nam tylko połączenie się do maszyny po SSH – my w tym celu wykorzystujemy program „MobaXterm”, który pozwala nam na połączenie się do paru maszyn na raz. (lecz można użyć do tego również programu Putty oraz zwykłego windowsowego terminala).



Wpisujemy adres ip, naszą nazwę użytkownika oraz wybieramy port.
I tym oto sposobem udało się nam połączyć do naszej maszyny po SSH.

! nasza nazwa użytkownika będzie musiała być vdi-user, ponieważ inaczej się nie zalogujemy. Nie mamy innych użytkowników stworzonych na serwerze.



2.3. Zadanie 3: Instalacja Docker'a oraz Docker-compose

Na koniec bułka z masłem – instalacja docker'a oraz docker-compose.

```
vdi-user@vdidesktop:~$ sudo apt-get install docker
[sudo] hasło użytkownika vdi-user:
Czytanie list pakietów... Gotowe
Budowanie drzewa zależności... Gotowe
Odczyt informacji o stanie... Gotowe
docker is already the newest version (1.5-2).
Następujące pakiety zostały zainstalowane automatycznie i nie są już więcej wymagane:
 chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
 intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
 libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
 libcodec2-1.0 libdavid5 libflite1 libgme0 libgsm1
 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm15 libmfx1
 libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0
 libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0
 libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1
 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0
 mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us va-driver-all
 vdpau-driver-all
Aby je usunąć należy użyć "sudo apt autoremove".
0 aktualizowanych, 0 nowo instalowanych, 0 usuwanych i 228 nieaktualizowanych.
vdi-user@vdidesktop:~$
```

Do wykonania użyliśmy polecenia „sudo apt-get install docker” które zainstalowało nam nasz program. Tak samo zrobiliśmy z docker-compose, dopisaliśmy tylko zawartość po myślniku.

3. Wnioski

Pierwszy etap projektu nie był zbyt wymagający. Jedyne problemy, które napotkaliśmy to rozterki jakiego rozwiązania użyć. Myśleliśmy nad postawieniem 4 wirtualnych maszyn na komputerze jednego z nas, lecz po głębokich namysłach zrezygnowaliśmy z tego pomysłu. Byłoby to mało praktyczne rozwiązanie, gdyż jeden z nas nie zawsze musiałby mieć włączony komputer co skutkowałoby ewentualnym brakiem dostępu do naszych maszyn a także pożerałoby to trochę zasobów a mając w zanadru alternatywę, którą była szkolna infrastruktura VDI zdecydowaliśmy się na nią. Poza tym ten etap pozwolił nam rozwinąć umiejętności komunikacyjne, docieranie do siebie i ogólnie pojętą pracę w zespole. Pomaganie kolegom z drużyny oraz wspólne naradzanie się. Jest to bardzo ważny element w jakiegokolwiek pracy, więc cieszymy się, że jak na razie w projekcie są poruszane też umiejętności miękkie a nie same umiejętności twarde.