Połączenie z serwerem z Windowsa: SSH, Windows Terminal i PuTTY

Poziom: podstawy **Dotyczy: OpenSSH (Windows Client), PuTTY** **Czas: ~20 min**

Ta strona pokazuje, jak z laptopa z Windows połączyć się z serwerem linuksowym takim jak `docker01`. Pokazujemy dwie drogi: wbudowanego klienta OpenSSH w Windows Terminal oraz starszy, wciąż spotykany program PuTTY. Wracamy tu, gdy zapomnimy dokładnej komendy albo znaczenia komunikatu o zmienionym odcisku klucza.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera

Fiszka

TYLKO ODCZYT

Sprawdzenie wersji klienta SSH: ssh -V

```
ssh -V
```

Ta komenda mówi, czy w systemie jest w ogóle zainstalowany klient SSH i w jakiej wersji.

> **Pojęcie: SSH.** SSH to szyfrowany sposób logowania się do innego komputera przez sieć i wydawania mu poleceń, tak jakbyśmy siedzieli bezpośrednio przed nim. Usługa `sshd` na serwerze nasłuchuje domyślnie na porcie `22`.

Klient OpenSSH jest częścią systemu bez dodatkowej instalacji od Windows 10 w wersji 1809 (aktualizacja z października 2018) i od Windows Server 2019. Wszystkie stacje robocze i serwery Windows w Zielonej mają już te wersje albo nowsze, więc `ssh` po prostu działa w Windows Terminal.

Przykład:

```
PS C:\Users\tomek> ssh -V
OpenSSH_for_Windows_9.5p1, LibreSSL 3.8.2
```

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **ssh** = Secure Shell: szyfrowane zdalne logowanie

TYLKO ODCZYT

Połączenie z serwerem po SSH: ssh nazwa_usera@NazwaSerwera

```
ssh nazwa_usera@NazwaSerwera
```

(gdzie `nazwa_usera` to konto istniejące na serwerze, a `NazwaSerwera` to jego nazwa DNS albo adres IP)

Po podaniu hasła albo po sprawdzeniu klucza dostajemy znak zachęty serwera zamiast własnego komputera. Od tej chwili każda wpisana komenda wykonuje się na serwerze, nie na naszym laptopie.

Przykład:

```
PS C:\Users\tomek> ssh tomek@docker01
tomek@docker01's password:
Last login: Sun Sep 27 09:58:03 2026 from 10.20.20.55
[tomek@docker01 ~]$
```

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **DNS** = Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP · **IP** = Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection) · **ssh** = Secure Shell: szyfrowane zdalne logowanie

TYLKO ODCZYT

Wygenerowanie pary kluczy SSH: ssh-keygen -t ed25519

```
ssh-keygen -t ed25519
```

> **Pojęcie: klucz SSH.** Klucz SSH to para dwóch powiązanych ze sobą plików: klucza prywatnego, który zostaje tylko na naszym komputerze i nigdy nikomu go nie pokazujemy, oraz klucza publicznego, który możemy bezpiecznie skopiować na dowolny serwer. Serwer wpuszcza nas, gdy nasz klucz prywatny pasuje do klucza publicznego zapisanego wcześniej w jego pliku `authorized_keys`.

Przykład:

```
PS C:\Users\tomek> ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\tomek/.ssh/id_ed25519):
Your identification has been saved in C:\Users\tomek/.ssh/id_ed25519
Your public key has been saved in C:\Users\tomek/.ssh/id_ed25519.pub
```

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **ssh-keygen** = wygeneruj parę kluczy SSH albo sprawdź odcisk klucza

ZMIENIA SYSTEM

Skopiowanie klucza publicznego na serwer bez ssh-copy-id: type oraz ssh

```
type $env:USERPROFILE\.ssh\id_ed25519.pub | ssh nazwa_usera@NazwaSerwera "mkdir -p
~/.ssh && chmod 700 ~/.ssh && cat >> ~/.ssh/authorized_keys && chmod 600
~/.ssh/authorized_keys"
```

Na Linuksie do tego zadania służy gotowe polecenie `ssh-copy-id`, którego Windows nie ma. Ten jeden wiersz robi dokładnie to samo ręcznie: wypisuje nasz klucz publiczny i przesyła go przez SSH prosto do pliku `authorized_keys` na serwerze.

Przykład:

```
PS C:\Users\tomek> type $env:USERPROFILE\.ssh\id_ed25519.pub | ssh tomek@docker01 "mkdir
-p ~/.ssh && chmod 700 ~/.ssh && cat >> ~/.ssh/authorized_keys && chmod 600
```

```
~/.ssh/authorized_keys"
tomek@docker01's password:
```

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **type** = pokaż, czym jest dane słowo dla powłoki: poleceniem, aliasem czy funkcją

TYLKO ODCZYT

Zapisanie skrótu do serwera: plik ~/.ssh/config

```
Host docker01
    HostName 10.20.10.42
    User tomek
    IdentityFile ~/.ssh/id_ed25519
```

Plik `config` w katalogu `.ssh` pozwala zapisać alias dla serwera razem z jego adresem, kontem i kluczem, więc nie trzeba ich za każdym razem wpisywać. Plik zakładamy w `C:\Users\tomek\.ssh\config` dowolnym edytorem tekstu.

Przykład:

```
PS C:\Users\tomek> ssh docker01
Last login: Sun Sep 27 10:12:44 2026 from 10.20.20.55
[tomek@docker01 ~]$
```

TYLKO ODCZYT

Skopiowanie pliku na serwer: scp

```
scp .\raport.xlsx nazwa_usera@NazwaSerwera:/home/nazwa_usera/
```

Komenda `scp` kopiuje plik tą samą szyfrowaną drogą co zwykłe połączenie SSH, bez osobnego programu do transferu plików. Ten sam alias z pliku `config` działa też tutaj zamiast pełnej nazwy serwera.

Przykład:

```
PS C:\Users\tomek> scp .\raport.xlsx tomek@docker01:/home/tomek/
raport.xlsx                   100%   48KB   2.1MB/s   00:00
```

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **scp** = secure copy: kopiowanie plików przez SSH

TYLKO ODCZYT

Połączenie pulpitem zdalnym z serwerem Windows: mstsc

```
mstsc /v:NazwaSerwera
```

SSH loguje nas do wiersza poleceń, ale serwery Windows takie jak `dc01` czy `sql01` obsługujemy zwykle przez pełny pulpit zdalny. Polecenie `mstsc` otwiera to samo okno Podłączania pulpitu zdalnego, które znamy z menu Start, tylko od razu z podanym adresem serwera.

Przykład:

```
PS C:\Users\tomek> mstsc /v:dc01
```

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **mstsc** = Microsoft Terminal Services Client: program otwierający pulpit zdalny (RDP) do innego Windows

KLIKAMY W INTERFEJSIE

Kliknięcia w PuTTY i PuTTYgen: interfejs

```
PuTTY: pole Host Name (or IP address) → NazwaSerwera → Port 22 → Connection type SSH → Open
PuTTYgen: Generate → poruszamy myszką nad pustym polem → Save private key → Save public key
```

PuTTY to starszy, samodzielny program do SSH, wciąż spotykany na starszych stacjach w firmach, które nie zdążyły przejść na Windows Terminal. PuTTYgen generuje klucze we własnym formacie, opisanym w sekcji niżej.

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera

Jak to działa

Rozwiń: jak to działa pod spodem

> **Pojęcie: SSH i port 22.** SSH to protokół szyfrowanego zdalnego logowania, a jego usługa `sshd` na serwerze nasłuchuje domyślnie na porcie `22`. Cała droga połączenia z laptopa Tomka do serwera `docker01` przechodzi przez sieć firmową i zaporę `fw01`, zanim usługa sshd w ogóle zobaczy nasze żądanie.



Droga połączenia SSH od laptopa Tomka przez sieć i zaporę do powłoki na docker01 (kliknięcie otwiera pełny rozmiar)

Przy pierwszym połączeniu z nowym serwerem klient SSH pokazuje odcisk klucza tego serwera i pyta, czy mu ufamy. Za pierwszym razem to normalne pytanie, na które odpowiadamy `yes`, bo świeżo postawiony serwer jeszcze nie ma zapisanego odcisku po naszej stronie.

> **Pojęcie: odcisk klucza (fingerprint).** Odcisk klucza to krótki ciąg znaków jednoznacznie identyfikujący klucz danego serwera, zapisywany potem lokalnie w pliku `known_hosts`. Przy każdym kolejnym połączeniu klient po cichu porównuje aktualny odcisk z tym zapisanym, bez pytania nas o nic.

Zupełnie inaczej wygląda sytuacja, gdy klient nagle krzyczy dużymi literami „REMOTE HOST IDENTIFICATION HAS CHANGED". Ten komunikat pojawia się tylko wtedy, gdy odcisk klucza zmienił się od ostatniego połączenia, co zwykle oznacza przeinstalowany system na serwerze. Zanim wpiszemy cokolwiek dalej, warto zapytać kolegę z zespołu, czy faktycznie ktoś świadomie postawił ten serwer od nowa, bo to samo ostrzeżenie pokazuje też próba podszycia się pod serwer.

Logowanie kluczem jest bezpieczniejsze i wygodniejsze niż logowanie hasłem, bo klucza nie da się zgadnąć ani podejrzeć znad ramienia. Hasło wystarcza do jednorazowego dostępu albo do kont serwisowych bez pliku `authorized_keys`, ale w codziennej pracy administratora klucz jest normą. Wiele serwerów w Zielonej ma logowanie hasłem w ogóle wyłączone, żeby ktoś z zewnątrz nie mógł go zgadywać automatem.

> **Pojęcie: plik .ppk.** PuTTY nie rozumie standardowego formatu klucza OpenSSH i wymaga własnego formatu `.ppk`, generowanego programem PuTTYgen. Klucz

wygenerowany komendą `ssh-keygen` trzeba najpierw wczytać do PuTTYgen i zapisać ponownie jako `.ppk`, zanim PuTTY zechce go użyć.

Z życia wzięte

Z ŻYCIA WZIĘTE

Tomek pierwszy raz w życiu łączy się z serwerem przez SSH, celuje w `docker01`. Widzi pytanie o odcisk klucza i komunikat, którego nigdy nie widział, więc zamyka okno i zgłasza „awarię łączności". Pokazujemy mu, że to zwykłe pierwsze połączenie, wpisujemy `yes` razem i logujemy się bez przeszkód. Dwa tygodnie później Tomek sam tłumaczy nowej osobie różnicę między tym pytaniem a prawdziwym ostrzeżeniem o zmienionym kluczu.

Częste błędy

Objaw	Przyczyna	Co robimy
`Connection refused`	usługa SSH na serwerze nie działa albo nasłuchuje na innym porcie	sprawdzamy `systemctl status sshd` na serwerze albo pytamy administratora serwera
`Connection timed out`	serwer nie odpowiada wcale, zwykle zapora albo zła nazwa serwera	sprawdzamy `ping NazwaSerwera` i reguły na `fw01`
`Permission denied (publickey)`	serwer nie zna naszego klucza publicznego albo wymaga hasła, którego nie podaliśmy	sprawdzamy, czy klucz trafił do `authorized_keys`, próbujemy `ssh -v` dla szczegółów
`Host key verification failed`	zapisany wcześniej odcisk klucza serwera różni się od tego, co dostajemy teraz	potwierdzamy z zespołem, że serwer faktycznie zmienił klucz, dopiero potem usuwamy stary wpis z `known_hosts`
PuTTY nie widzi klucza wygenerowanego przez ssh-keygen	PuTTY rozumie tylko format `.ppk`, nie standardowy plik OpenSSH	wczytujemy klucz do PuTTYgen i zapisujemy jako `.ppk`

Ćwiczenia

1. Łączymy się pierwszy raz z nowo postawionym serwerem `k3s01` i klient pyta, czy ufamy jego odciskowi klucza. Czy to powód do niepokoju?

Odpowiedź

Nie, to normalne pytanie przy pierwszym połączeniu z serwerem, którego odcisku klucza jeszcze nie znamy. Niepokoić powinien dopiero komunikat „REMOTE HOST IDENTIFICATION HAS CHANGED" przy serwerze, z którym łączyliśmy się już wcześniej.

1. Kolega prosi o skopiowanie klucza publicznego na serwer, ale pracuje na Windows i nie ma `ssh-copy-id`. Jak inaczej zrobi to samo?

Odpowiedź

Może wypisać zawartość swojego pliku `id_ed25519.pub` komendą `type` i przekazać ją przez potok do `ssh`, które dopisze klucz do pliku `authorized_keys` na serwerze. To dokładnie to samo, co robi `ssh-copy-id` na Linuksie, tylko krok po kroku.

1. Klucz wygenerowany przez `ssh-keygen` nie działa w PuTTY. Dlaczego i co trzeba zrobić?

Odpowiedź

PuTTY rozumie tylko własny format klucza `.ppk`, a nie standardowy plik OpenSSH. Trzeba wczytać klucz do programu PuTTYgen i zapisać go ponownie jako plik `.ppk`.

Revision #29
Created 2026-09-28 05:52:37 UTC by Tomasz Konieczny
Updated 2026-10-26 14:18:05 UTC by Tomasz Konieczny