

Zdalne zarządzanie - Serwer OpenSSH

Wprowadzenie

Serwer OpenSSH to potężny zbiór narzędzi do zdalnego zarządzania i przesyłania danych między komputerami w sieci.

OpenSSH to bezpłatna wersja rodziny narzędzi protokołu Secure Shell (SSH) do zdalnego sterowania lub przesyłania plików między komputerami. Tradycyjne narzędzia używane do realizacji tych funkcji, takie jak *telnet* lub *rcp*, są niebezpieczne i gdy są używane, przesyłają hasło użytkownika w postaci jawnego tekstu. OpenSSH zapewnia demona serwera i narzędzia klienckie, aby ułatwić bezpieczne szyfrowane operacje zdalnego sterowania i przesyłania plików, skutecznie zastępując starsze narzędzia.

Składnik serwera OpenSSH, *ssh*, stale nasłuchuje połączeń klientów z dowolnego narzędzia klienta. Kiedy pojawia się żądanie połączenia, *sshd* konfiguruje prawidłowe połączenie w zależności od rodzaju połączenia narzędzia klienta. Na przykład, jeśli komputer zdalny łączy się z aplikacją klienta *ssh*, serwer OpenSSH konfiguruje sesję zdalnego sterowania po uwierzytelnieniu. Jeśli zdalny użytkownik łączy się z serwerem OpenSSH za pomocą *scp*, demon serwera OpenSSH inicjuje bezpieczną kopię plików między serwerem a klientem po uwierzytelnieniu. OpenSSH może korzystać z wielu metod uwierzytelniania, w tym zwykłego hasła, klucza publicznego i protokołu *Kerberos*.

Instalacja serwera OpenSSH

Przed przystąpieniem do jakiegokolwiek instalacji należy zaktualizować system:

```
sudo apt update
```

Instalacja aplikacji klienta i serwera OpenSSH jest prosta.

Aby zainstalować aplikację serwera OpenSSH i powiązane pliki pomocnicze, użyj w wierszu polecenia polecenia:

```
sudo apt -y install openssh-server
```

Pakiet *openssh-server* można również wybrać do instalacji podczas procesu instalacji Server Edition.

Konfiguracja SSH - uwierzytelnianie za pomocą hasła

Możesz skonfigurować domyślne zachowanie aplikacji serwera OpenSSH, *sshd*, edytując plik */etc/ssh/sshd_config*. Aby uzyskać informacje o dyrektywach konfiguracyjnych użytych w tym pliku, możesz wyświetlić odpowiednią stronę podręcznika poleceniem:

```
man sshd_config
```

W pliku konfiguracyjnym *sshd* znajduje się wiele dyrektyw kontrolujących takie rzeczy, jak ustawienia komunikacji i tryby uwierzytelniania. Poniżej przedstawiono przykłady dyrektyw konfiguracyjnych, które można zmienić, edytując plik */etc/ssh/sshd_config*.

Przed edycją pliku konfiguracyjnego można wykonać kopię oryginalnego pliku i zabezpieczyć go przed zapisem, aby mieć oryginalne ustawienia, jako odniesienie, albo do ponownego użycia w razie potrzeby.

Skopiuj plik */etc/ssh/sshd_config* i chroń go przed zapisem za pomocą następujących poleceń wydanych w wierszu polecenia:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
```

```
sudo chmod a-w /etc/ssh/sshd_config.original
```

Przykłady dyrektyw konfiguracyjnych, które możesz zmienić:

1. Aby ustawić OpenSSH na nasłuchiwanie na porcie TCP 2222 zamiast domyślnego portu TCP 22, zmień dyrektywę Port na 2222:

Port 2222 - sprawdzamy nie zmieniamy - pozostajemy na porcie domyślnym

```
GNU nano 4.8 /etc/ssh/sshd_config Zmodyfikowany
# possible, but leave them commented. Uncommented options override the
# default value.

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

Port 22
#AddressFamily any
```

2. Aby *sshd* zezwalał na poświadczenia logowania oparte na kluczu publicznym, wystarczy dodać lub zmodyfikować wiersz:

PubkeyAuthentication yes

Jeśli linia jest już obecna, upewnij się, że nie została skomentowana (#).

```
GNU nano 4.8 /etc/ssh/sshd_config Zmodyfikowany
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
```

3. Aby twój serwer OpenSSH wyświetlał zawartość pliku */etc/issue.net* jako banner przed logowaniem, po prostu dodaj lub zmodyfikuj linię:

Baner /etc/issue.net

W pliku */etc/ssh/sshd_config*.

```
GNU nano 4.8 /etc/ssh/sshd_config Zmodyfikowany
#VersionAddendum none

# no default banner path
Banner /etc/issue.net

# Allow client to pass locale environment variables
AcceptEnv LANG LC *
```

Np. W pliku */etc/issue.net* dopisujemy linię „Witam – serwer ks23-xyy”

```
us-3n00@ks23-3n00:/$ cat /etc/issue.net
Ubuntu 20.04.3 LTS
Witam - serwer ks23-3n00
```

4. Uwierzytelnianie hasła dla Open SSH Server na Ubuntu jest domyślnie włączone, więc można się zalogować bez zmiany jakichkolwiek ustawień. Co więcej, konto root jest domyślnie zabronione. Uwierzytelnianie za pomocą hasła „PermitRootLogin prohibit-password”, więc ustawienie domyślne jest dobre do użycia. Ale jeśli chcemy zabronić wszystkim logowania do roota, zmieniamy na *PermitRootLogin no*

Po wprowadzeniu zmian w pliku */etc/ssh/sshd_config* zapisz plik i zrestartuj aplikację serwera *sshd*, aby wprowadzić zmiany za pomocą następującego polecenia w wierszu polecenia:

```
sudo systemctl restart sshd.service
```

Sprawdzamy status

```
sudo systemctl status sshd.service
```

UWAGA

Dostępnych jest wiele innych dyrektyw konfiguracyjnych dla *sshd*, które zmieniają zachowanie aplikacji serwera w zależności od potrzeb. Pamiętaj jednak, że jeśli jedyną metodą dostępu do serwera jest *ssh* i popełnisz błąd podczas konfigurowania *sshd* za pomocą pliku */etc/ssh/sshd_config*, może się okazać, że zostałeś zablokowany na serwerze po ponownym uruchomieniu. Ponadto, jeśli dostarczona zostanie niepoprawna dyrektywa konfiguracyjna, serwer *sshd* może odmówić uruchomienia, dlatego należy zachować szczególną ostrożność podczas edytowania tego pliku na serwerze zdalnym.

Ubuntu desktop -Instalacja klienta OpenSSH


Aby zainstalować aplikacje klienckie OpenSSH w systemie Ubuntu, użyj tego polecenia w wierszu polecenia:

```
sudo apt-get -y install openssh-client
```

W zasadzie po zainstalowaniu aplikacji klienckiej, możemy połączyć się z serwerem, wydając polecenie wg wzorca **ssh [„nazwa użytkownika”@nazwa hosta lub adres IP]**. Konto „nazwa użytkownika” musi istnieć na serwerze.

W naszym przypadku zalogujemy się na koncie usxxyy wpisując:

```
ssh usxxyy@172.22.y.1
```



```

us-3n00@ks23-3n00: ~
ud-3n00@k1d23-3n00:/$ ssh us-3n00@172.22.0.1
Ubuntu 20.04.3 LTS
Witam - serwer ks23-3n00
us-3n00@172.22.0.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-164-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Oct 16 19:37:53 UTC 2023

System load:  0.0               Users logged in:      1
Usage of /:   41.1% of 18.53GB   IPv4 address for enp0s3: 10.0.2.15
Memory usage: 37%              IPv4 address for enp0s8: 172.22.0.1
Swap usage:   0%               IPv4 address for enp0s8: 172.22.0.2
Processes:   187

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

146 aktualizacji można zastosować natychmiast.
Aby wyświetlić te dodatkowe aktualizacje, należy wprowadzić w terminalu: apt list --upgradable

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Oct 16 19:36:55 2023 from 172.22.0.101
us-3n00@ks23-3n00:~$

```

Od tej chwili możemy z hosta wykonać wszelkie polecenia na zdalnym serwerze, co widać po zgłoszeniu. Jesteśmy na serwerze

Windows 10 klient – zarządzanie z wiersza poleceń

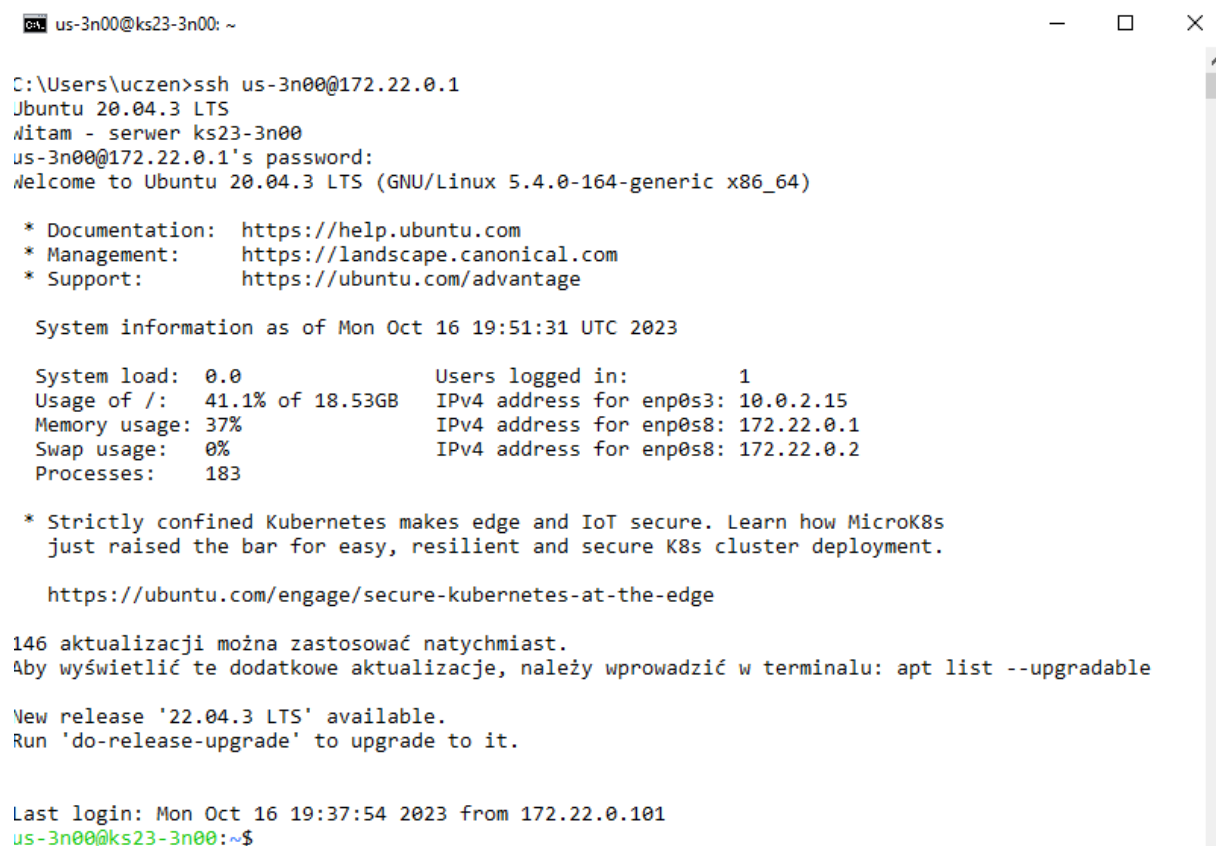
W windowsie nie potrzebujemy instalować dodatkowych aplikacji klienckich. W wierszu poleceń klienta znajdującego się w sieci serwera, wpisujemy:

```
ssh usxxyy@172.22.y.1
```

gdzie

xx to klasa

yy to nr z dziennika



```

C:\Users\uczen>ssh us-3n00@172.22.0.1
Ubuntu 20.04.3 LTS
Witam - serwer ks23-3n00
us-3n00@172.22.0.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-164-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Oct 16 19:51:31 UTC 2023

System load:  0.0           Users logged in:      1
Usage of /:   41.1% of 18.53GB IPv4 address for enp0s3: 10.0.2.15
Memory usage: 37%          IPv4 address for enp0s8: 172.22.0.1
Swap usage:   0%           IPv4 address for enp0s8: 172.22.0.2
Processes:    183

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

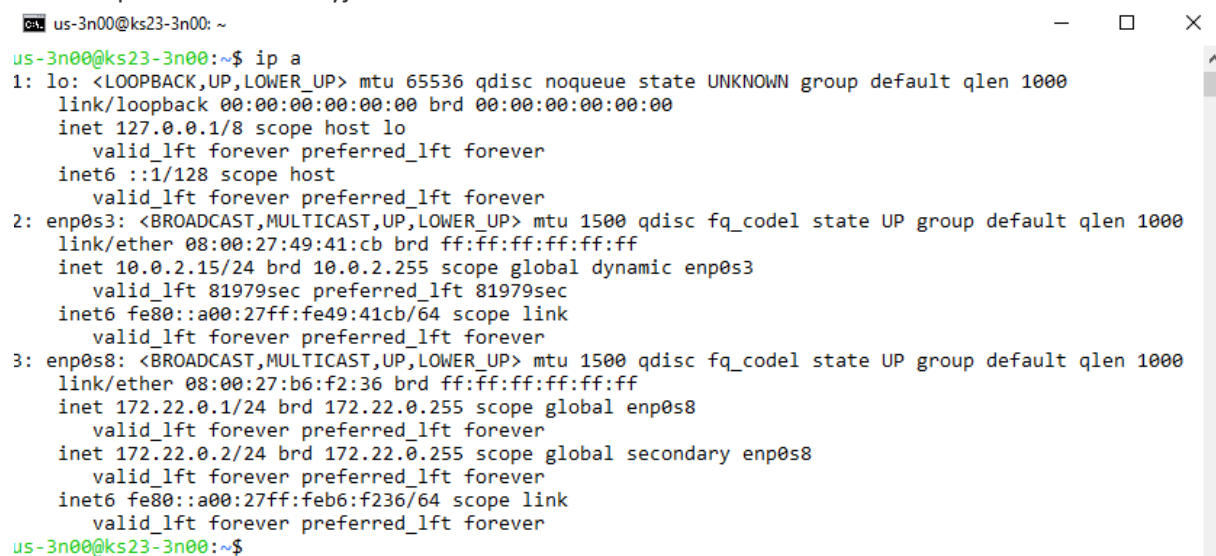
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

146 aktualizacji można zastosować natychmiast.
Aby wyświetlić te dodatkowe aktualizacje, należy wprowadzić w terminalu: apt list --upgradable

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Oct 16 19:37:54 2023 from 172.22.0.101
us-3n00@ks23-3n00:~$
  
```

Jak widać po zgłoszeniu i nagłówku okna jesteśmy na serwerze Ubuntu i możemy na nim wykonywać wszelkie prace administracyjne.

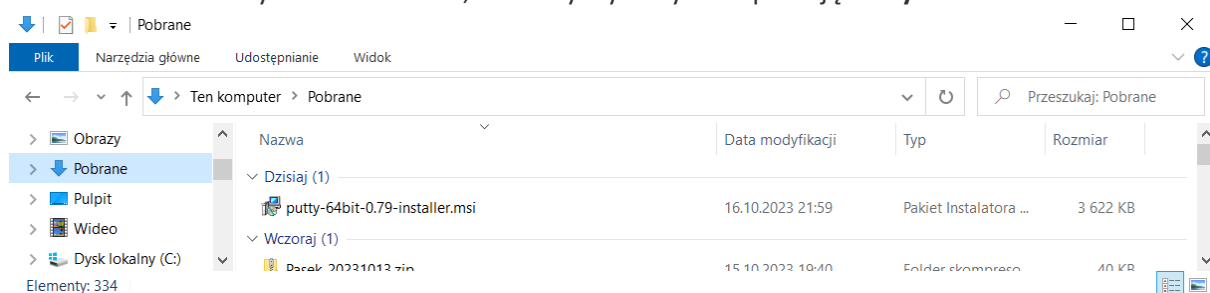


```

us-3n00@ks23-3n00:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:49:41:cb brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 81979sec preferred_lft 81979sec
    inet6 fe80::a00:27ff:fe49:41cb/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b6:f2:36 brd ff:ff:ff:ff:ff:ff
    inet 172.22.0.1/24 brd 172.22.0.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet 172.22.0.2/24 brd 172.22.0.255 scope global secondary enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb6:f236/64 scope link
        valid_lft forever preferred_lft forever
us-3n00@ks23-3n00:~$
  
```

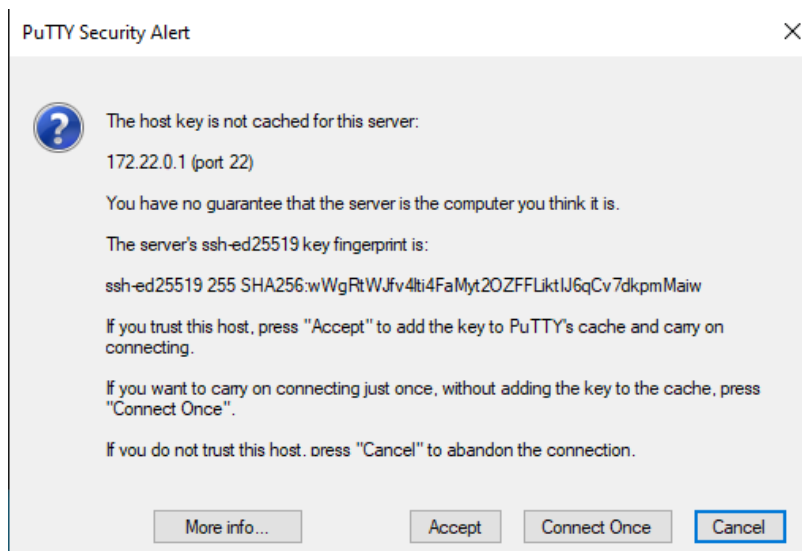
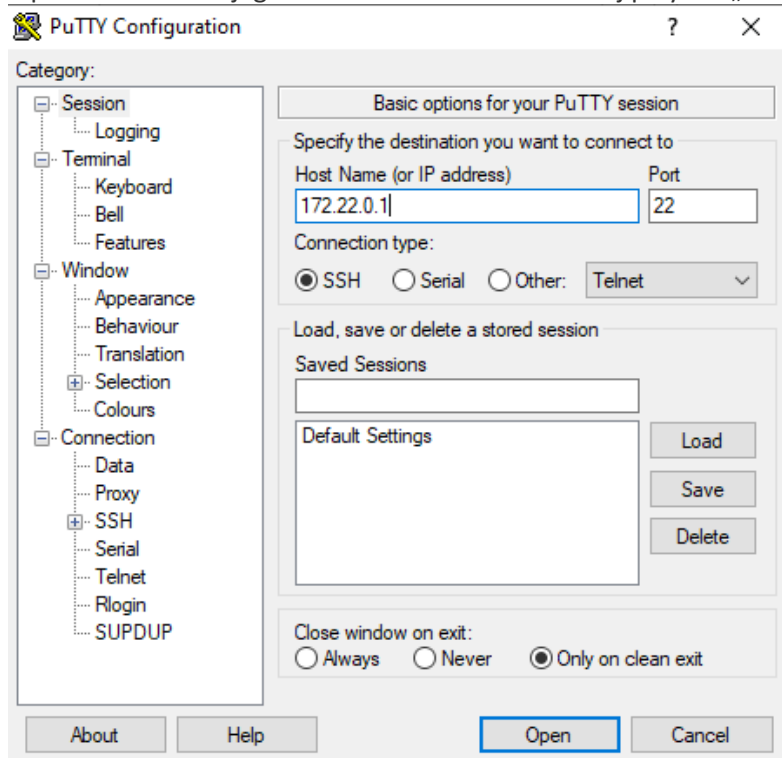
Windows 10 klient – zarządzanie poprzez aplikację Putty

Jako klienta SSH dla systemu Windows, możemy wykorzystać aplikację **Putty**.



Ten przykład pokazuje konfigurację połączenia w **Putty**:

Wpisz adres IP swojego serwera zaznacz SSH i kliknij przycisk „Otwórz”.



Akceptujemy połączenie

Po połączeniu z serwerem i uwierzytelnieniu, z wykorzystaniem SSH, możliwe jest praca na komputerze zdalnym.

```

us-3n00@ks23-3n00: ~
login as: us-3n00
Pre-authentication banner message from server:
| Ubuntu 20.04.3 LTS
| Witam - serwer ks23-3n00
End of banner message from server
us-3n00@172.22.0.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-164-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Oct 16 20:22:04 UTC 2023

System load:  0.0               Users logged in:      1
Usage of /:   41.1% of 18.53GB  IPv4 address for enp0s3: 10.0.2.15
Memory usage: 37%              IPv4 address for enp0s8: 172.22.0.1
Swap usage:   0%               IPv4 address for enp0s8: 172.22.0.2
Processes:   189

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

146 aktualizacji można zastosować natychmiast.
Aby wyświetlić te dodatkowe aktualizacje, należy wprowadzić w terminalu: apt list --upgradable

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Oct 16 19:54:44 2023 from 172.22.0.102
us-3n00@ks23-3n00:~$
us-3n00@ks23-3n00: ~
us-3n00@ks23-3n00:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:49:41:cb brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 79551sec preferred_lft 79551sec
    inet6 fe80::a00:27ff:fe49:41cb/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b6:f2:36 brd ff:ff:ff:ff:ff:ff
    inet 172.22.0.1/24 brd 172.22.0.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet 172.22.0.2/24 brd 172.22.0.255 scope global secondary enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb6:f236/64 scope link
        valid_lft forever preferred_lft forever
us-3n00@ks23-3n00:~$

```


Konfiguracja SSH - uwierzytelnianie za pomocą klucza SSH

Klucze SSH

Klucze SSH umożliwiają uwierzytelnianie między dwoma hostami bez potrzeby podawania hasła. Uwierzytelnianie za pomocą klucza SSH wykorzystuje dwa klucze, klucz *prywatny* i /lub klucz *publiczny*.

Aby wygenerować klucze, na desktopie, w wierszu polecenia wpisz:

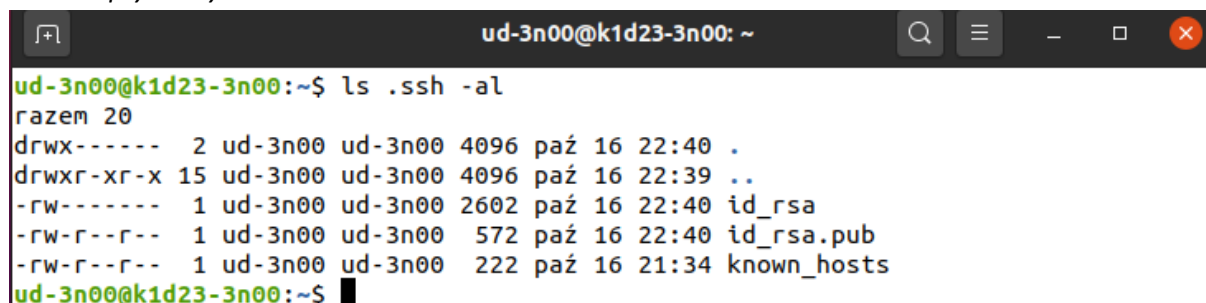
```
ssh-keygen -t rsa
```

Spowoduje to wygenerowanie kluczy przy użyciu *algorytmu RSA*. Podczas procesu pojawi się monit o utworzenie klucza oraz podanie hasła. W obu przypadkach, po prostu naciśnij *Enter*.



```
ud-3n00@k1d23-3n00: /
ud-3n00@k1d23-3n00:/$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ud-3n00/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ud-3n00/.ssh/id_rsa
Your public key has been saved in /home/ud-3n00/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:gipDyQxnOuBDsNiWf2hIeiB6b6+7+oRUjGW4neY19rI ud-3n00@k1d23-3n00
The key's randomart image is:
+---[RSA 3072]-----+
|. .o
|oo.*
|B.Ooo.
|XOo++o+
|=B=o=oooS
|. .=.....
|o o + o
| o o .E
| .o=+.
+----[SHA256]-----+
ud-3n00@k1d23-3n00:/$
```

Domyślnie klucz *publiczny* jest zapisywany w pliku `~/.ssh/id_rsa.pub`, natomiast `~/.ssh/id_rsa` jest kluczem *prywatnym*.



```
ud-3n00@k1d23-3n00: ~
ud-3n00@k1d23-3n00:~$ ls .ssh -al
razem 20
drwx----- 2 ud-3n00 ud-3n00 4096 paź 16 22:40 .
drwxr-xr-x 15 ud-3n00 ud-3n00 4096 paź 16 22:39 ..
-rw----- 1 ud-3n00 ud-3n00 2602 paź 16 22:40 id_rsa
-rw-r--r-- 1 ud-3n00 ud-3n00 572 paź 16 22:40 id_rsa.pub
-rw-r--r-- 1 ud-3n00 ud-3n00 222 paź 16 21:34 known_hosts
ud-3n00@k1d23-3n00:~$
```

Teraz skopiuj plik `id_rsa` na zdalny host (serwer), wprowadzając polecenie wg wzorca - **ssh-copy-id username@remotehost**

```
ssh-copy-id usxxyy@172.22.0.1
```

gdzie `xx` to klasa a `yy` nr z dziennika

```

ud-3n00@k1d23-3n00: ~
ud-3n00@k1d23-3n00:~$ ssh-copy-id us-3n00@172.22.0.1
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
Ubuntu 20.04.3 LTS
Witam - serwer ks23-3n00
us-3n00@172.22.0.1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'us-3n00@172.22.0.1'"
and check to make sure that only the key(s) you wanted were added.

ud-3n00@k1d23-3n00:~$

```

co dołączy klucz rsa do ~/.ssh/authorized_keys na serwerze

```

us-3n00@ks23-3n00: ~
us-3n00@ks23-3n00:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCWkFmLQR7c5FBrT+Y1tALnadEnGPLSkJNg6e2+TQnA
Z3EtCVS3MrANPjdpGWQZwZjdjH6T8Ie8CACH9FiGXhUGqOWGoOpbANuHei0qnzzBWFa8ZXLtWSFUgyVv
2WNsYSM9nvRx20cItICLTj7xv5bgVs7loB5Acvj54NNNs0rrFiKwcxeKIC14dY7WAnn/uZQkbftcW4re
BHZLPPRPOcVUnQtZG38K4uCTf428IYjjLWOGF0ycJhhExHMBIESN5jJpEhlb+5jQtMBEMAIBQfFrA9a+
6KVkGiT0thHjL6KzwINP9g3aC1+V6y01q0JBtFwsvIoxeNNcvjLS0dskZKkKFFZHT1bpoZnx/aKtgjC4
mDR57cKM0d4lhrjKAwM9l+I9QFv2zHM3hZwvvrkbbd4LUTvJtx9n0fwIzAVF8TDW1LAo5Ibohcn7Nlb0
KagUjZ/bdnJel1bZUA5MC7cQtz084xVAlNysMMom9ANbVvdOhCCIG3vaAvrbFS+X0+Pr+jk= ud-3n00
@k1d23-3n00
us-3n00@ks23-3n00:~$

```

i pozwoli na połączenie zdalne z hosta (k3d23-3n00) do serwera bez hasła, jako użytkownik us3n00 z konta ud3n00.

```

us-3n00@ks23-3n00: ~
ud-3n00@k1d23-3n00:~$ ssh us-3n00@172.22.0.1
Ubuntu 20.04.3 LTS
Witam - serwer ks23-3n00
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-164-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Oct 16 20:57:22 UTC 2023

System load:  0.0               Users logged in:      1
Usage of /:   41.1% of 18.53GB   IPv4 address for enp0s3: 10.0.2.15
Memory usage: 37%              IPv4 address for enp0s8: 172.22.0.1
Swap usage:   0%                IPv4 address for enp0s8: 172.22.0.2
Processes:   190

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

146 aktualizacji można zastosować natychmiast.
Aby wyświetlić te dodatkowe aktualizacje, należy wprowadzić w terminalu: apt lis
t --upgradable

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Oct 16 20:55:41 2023 from 172.22.0.1
us-3n00@ks23-3n00:~$

```


Na koniec sprawdzamy uprawnienia do pliku **authorized_keys**, tylko uwierzytelniony użytkownik powinien mieć uprawnienia do odczytu i zapisu.

```
us-3n00@ks23-3n00: ~
ls -al .ssh
razem 16
drwx----- 2 us-3n00 us-3n00 4096 paź 16 20:55 .
drwxr-xr-x 15 us-3n00 us-3n00 4096 paź 15 20:21 ..
-rw----- 1 us-3n00 us-3n00 572 paź 16 20:46 authorized_keys
-rw-r--r-- 1 us-3n00 us-3n00 222 paź 16 20:55 known_hosts
us-3n00@ks23-3n00: ~$
```

Moje uprawnienia są prawidłowe. Jeśli uprawnienia są nieprawidłowe, zmień je:

chmod 600 .ssh/authorized_keys

Od teraz możemy łączyć się zdalnie z serwerem bez podawania hasła i realizować zadania administracyjne.

```
ud-3n00@k1d23-3n00: ~
us-3n00@ks23-3n00: ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:49:41:cb brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 78030sec preferred_lft 78030sec
    inet6 fe80::a00:27ff:fe49:41cb/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b6:f2:36 brd ff:ff:ff:ff:ff:ff
    inet 172.22.0.1/24 brd 172.22.0.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet 172.22.0.2/24 brd 172.22.0.255 scope global secondary enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb6:f236/64 scope link
        valid_lft forever preferred_lft forever
us-3n00@ks23-3n00: ~$ exit
wylogowanie
Connection to 172.22.0.1 closed.
ud-3n00@k1d23-3n00: ~$
```