

# KONFIGURACJA SAMBY W UBUNTU SERVER 18.04

## I. Instalacja serwera plików - SAMBA.

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo apt install samba
[sudo] password for administrator:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  attr ibverbs-providers libavahi-client3 libavahi-common-data libavahi-common3 libcephfs
  libcups2 libgpgme11 libibverbs1 libjansson4 libldb1 libnl-route-3-200 libnspr4 libnss3
  libpython-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib librados2 libtal
  libtdb1 libtevent0 libwbclient0 python python-crypto python-dnspython python-ldb python
  python-samba python-talloc python-tdb python2.7 python2.7-minimal samba-common samba-co
  samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
Suggested packages:
  cups-common python-doc python-tk python-crypto-doc python-gpgme python2.7-doc binutils
  binfmt-support bind9 bind9utils ctdb ldb-tools ntp | chrony smbldap-tools winbind
  heimdal-clients
The following NEW packages will be installed:
  attr ibverbs-providers libavahi-client3 libavahi-common-data libavahi-common3 libcephfs
  libcups2 libgpgme11 libibverbs1 libjansson4 libldb1 libnl-route-3-200 libnspr4 libnss3
  libpython-stdlib libpython2.7 libpython2.7-minimal libpython2.7-stdlib librados2 libtal
  libtdb1 libtevent0 libwbclient0 python python-crypto python-dnspython python-ldb python
  python-samba python-talloc python-tdb python2.7 python2.7-minimal samba samba-common
  samba-common-bin samba-dsdb-modules samba-libs samba-vfs-modules tdb-tools
0 upgraded, 40 newly installed, 0 to remove and 1 not upgraded.
Need to get 20.1 MB of archives.
After this operation, 97.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

### 1.1

**SAMBA** to oprogramowanie umożliwiające uruchomienie tak zwanego serwera plików na systemie Linux. Umożliwia też utworzenie mieszanego środowiska, w którym mogą działać komputery z systemem operacyjnym Unix, Linux oraz Windows i wzajemnie korzystać ze swoich zasobów. Aby zainstalować sambę należy wydać polecenie **sudo apt install samba**

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
Created symlink /etc/systemd/system/multi-user.target.wants/smbd.service → /lib/systemd/s
service.
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for libc-bin (2.27-3ubuntu1.4) ...
Processing triggers for systemd (237-3ubuntu10.44) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
administrator@serwer110:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-02-20 17:23:56 UTC; 38s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 2318 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 4 (limit: 2317)
    CGroup: /system.slice/smbd.service
            └─2318 /usr/sbin/smbd --foreground --no-process-group
              └─2334 /usr/sbin/smbd --foreground --no-process-group
                └─2339 /usr/sbin/smbd --foreground --no-process-group
                  └─2342 /usr/sbin/smbd --foreground --no-process-group

Feb 20 17:23:56 serwer110 systemd[1]: Starting Samba SMB Daemon...
Feb 20 17:23:56 serwer110 systemd[1]: Started Samba SMB Daemon.
administrator@serwer110:~$ _
```

## 1.2

Po zakończeniu sprawdzamy czy samba działa poleceniem **sudo systemctl status smbd**.  
Widzimy, że jest OK i możemy rozpocząć jej konfigurację

## II. Konfiguracja połączenia anonimowego.

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo mkdir /home/pitbull
administrator@serwer110:~$ sudo chmod 777 /home/pitbull
administrator@serwer110:~$ ls -l /home
total 12
drwxr-xr-x 4 administrator administrator 4096 Feb  2 16:28 administrator
drwxr-xr-x 3 informatykj informatykj 4096 Feb  2 16:48 informatykj
drwxrwxrwx 2 root root 4096 Feb 20 17:41 pitbull
administrator@serwer110:~$ _
```

## 2.1

Najpierw utworzymy folder, który będzie zasobem sieciowym. Tworzymy poleceniem **sudo mkdir /home/pitbull**. Nadamy mu pełne uprawnienia dla wszystkich użytkowników **sudo chmod 777 /home/pitbull**. Po wyświetleniu zawartości folderu **/home** widzimy, że jego właścicielem jest **root** (grupa także). Trzeba to zmienić

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo chown nobody:nogroup /home/pitbull
administrator@serwer110:~$ ls -l /home
total 12
drwxr-xr-x 4 administrator administrator 4096 Feb  2 16:28 administrator
drwxr-xr-x 3 informatuki informatuki 4096 Feb  2 16:48 informatuki
drwxrwxrwx 2 nobody nogroup 4096 Feb 20 17:41 pitbull
administrator@serwer110:~$ _
```

## 2.2

Właścicielem folderu będzie user **nobody**, a grupa **nogroup**. Są to obiekty używane w nieuprzywilejowanych procesach więc możemy spokojnie je zastosować do naszego ćwiczenia.

Zmianę właściciela i grupy możemy wykonać jednym lub dwoma poleceniami:

**sudo chown nobody /home/pitbull**, a następnie **sudo chgrp nogroup /home/pitbull**

A jeśli chcemy wykonać to jednym poleceniem to wpisujemy: **sudo chown nobody:nogroup /home/pitbull**

Następnie sprawdzamy poleceniem **ls -l /home** i widzimy, że jest OK

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf-kopia
administrator@serwer110:~$ cd /etc/samba
administrator@serwer110:/etc/samba$ ls -l
total 32
-rw-r--r-- 1 root root 8 Oct 16 10:50 gdbcommands
-rw-r--r-- 1 root root 9542 Feb 20 17:23 smb.conf
-rw-r--r-- 1 root root 9542 Feb 20 18:07 smb.conf-kopia
drwxr-xr-x 2 root root 4096 Oct 16 10:50 t1s
administrator@serwer110:/etc/samba$ _
```

## 2.3

Plik konfiguracyjny samby to **smb.conf**. Zanim cokolwiek w nim zmienimy, warto zrobić sobie jego kopię. Wykonujemy to poleceniem **sudo cp /etc/samba/smb.conf /etc/samba/smb.conf-kopia**. Przejdziemy do tego katalogu poleceniem **cd /etc/samba**, a potem sprawdzamy czy nasza kopia została tam utworzona

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
GNU nano 2.9.3 /etc/samba/smb.conf

#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.

#=====Global Settings=====

[global]

## Browsing/Identification ###
# Change this to the workgroup/NT-domain name your Samba server will part of
workgroup = SALA110
# server string is the equivalent of the NT Description field
server string = %h server (Samba, Ubuntu)
# Ten wpis nie jest niezbędny, ale do logowania bez autoryzacji wykorzystamy konto nobody
guest account = nobody
# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable its WINS Server
wins support = no
# WINS Server - Tells the NMBD components of Samba to be a WINS Client
2.4
```

Teraz edytujemy plik konfiguracyjny: **sudo nano /etc/samba/smb.conf**. Dokonamy w nim takich zmian aby dowolny użytkownik mógł korzystać z naszego zasobu bez ograniczeń. W sekcji **global** zmieniamy opcje **workgroup = SALA110**, **server string = %h server**, dodajemy wpis **guest account = nobody**

```
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin
```

```
# Nasz folder sieciowy
```

```
[pitbull]
  path = /home/pitbull
  comment = pitbull-guest
  writable = yes
  browseable = yes
  guest ok = yes
```

## 2.5

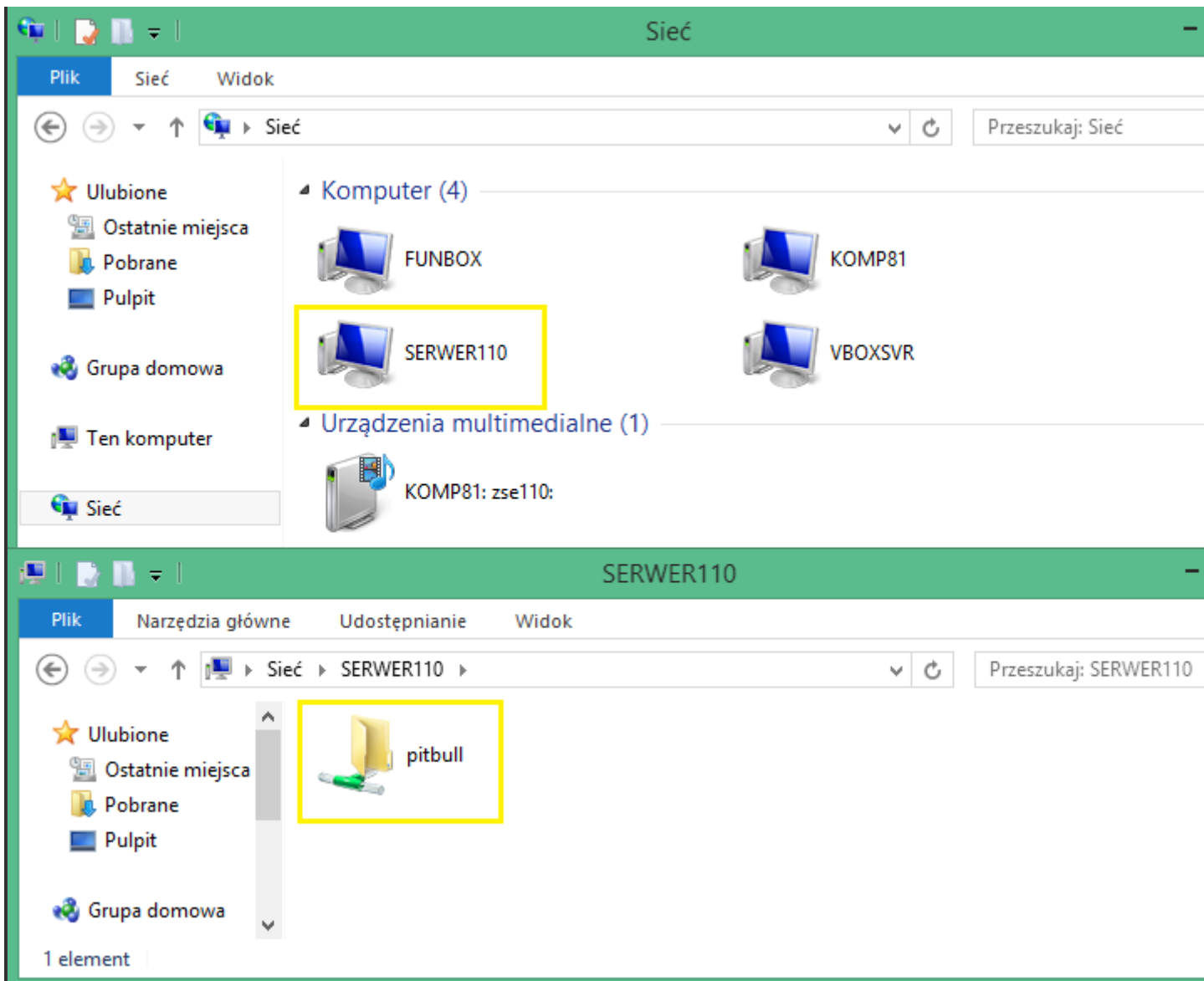
Następnie przechodzimy na koniec pliku i dodajemy nasz folder sieciowy z konkretnymi parametrami dokonując wpisów jak wyżej. Zapisujemy zmiany i zamykamy plik

```
administrator@serwer110:/etc/samba$ sudo systemctl restart smb
administrator@serwer110:/etc/samba$ sudo systemctl status smb
• smb.service - Samba SMB Daemon
  Loaded: loaded (/lib/systemd/system/smb.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2021-02-20 19:02:56 UTC; 2s ago
    Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 3593 (smbd)
  Status: "smbd: ready to serve connections..."
    Tasks: 4 (limit: 2317)
  CGroup: /system.slice/smb.service
          └─3593 /usr/sbin/smbd --foreground --no-process-group
             └─3617 /usr/sbin/smbd --foreground --no-process-group
                └─3618 /usr/sbin/smbd --foreground --no-process-group
                   └─3619 /usr/sbin/smbd --foreground --no-process-group

Feb 20 19:02:56 serwer110 systemd[1]: Starting Samba SMB Daemon...
Feb 20 19:02:56 serwer110 systemd[1]: Started Samba SMB Daemon.
administrator@serwer110:/etc/samba$ _
```

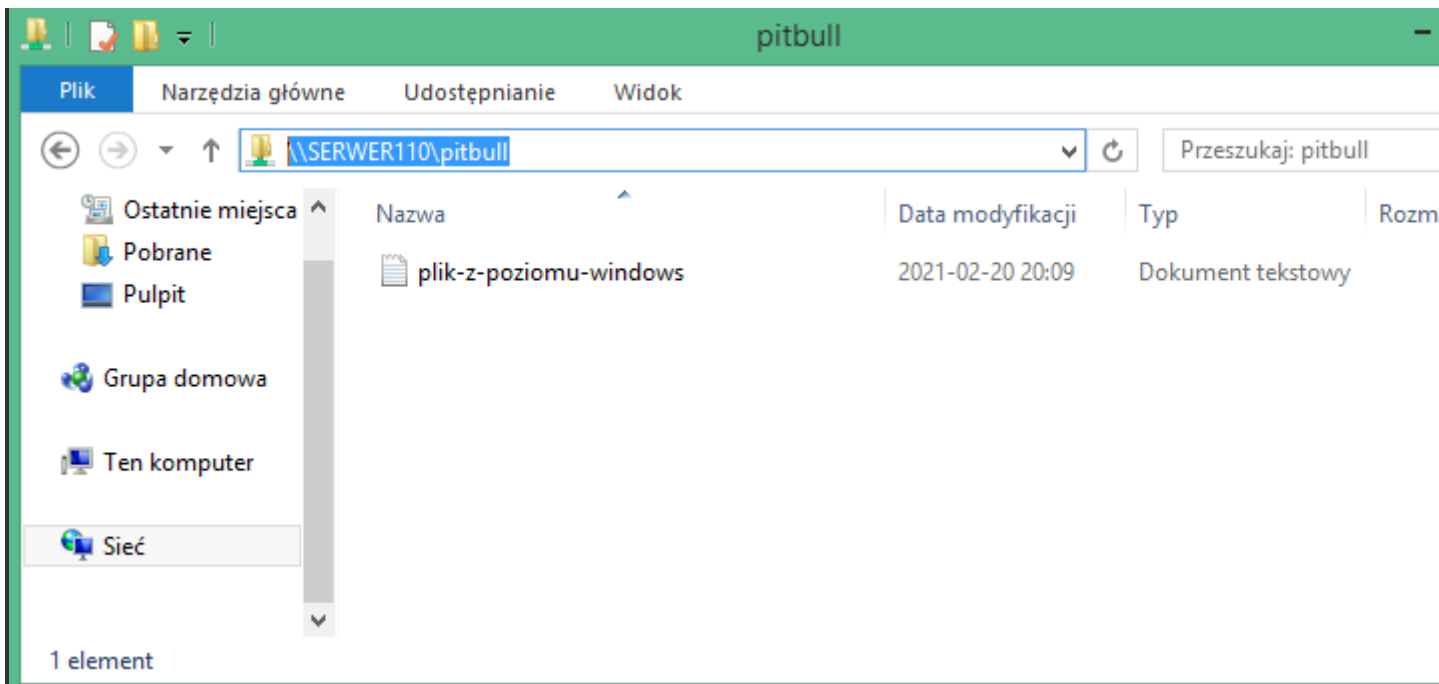
## 2.6

Następnie restartujemy naszą usługę poleceniem **sudo systemctl restart smb** oraz sprawdzamy jej status. Działa!



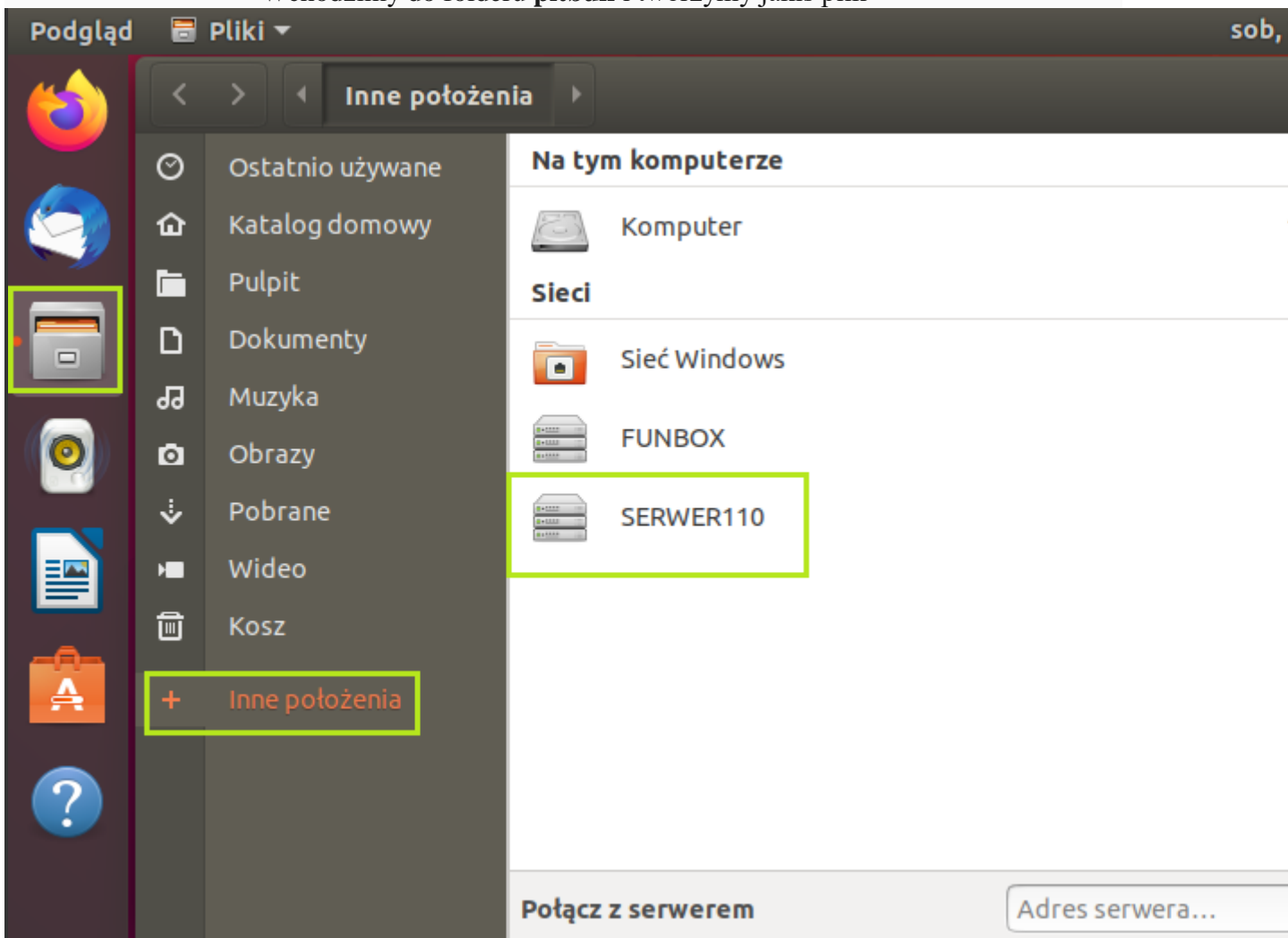
2.7

Teraz sprawdzamy z poziomu klienta Windows. Widzimy nasz serwer, ale gdyby go nie było, to w pasku eksploratora wpisujemy `\\10.80.80.1` i dostęp do folderu powinien się pojawić



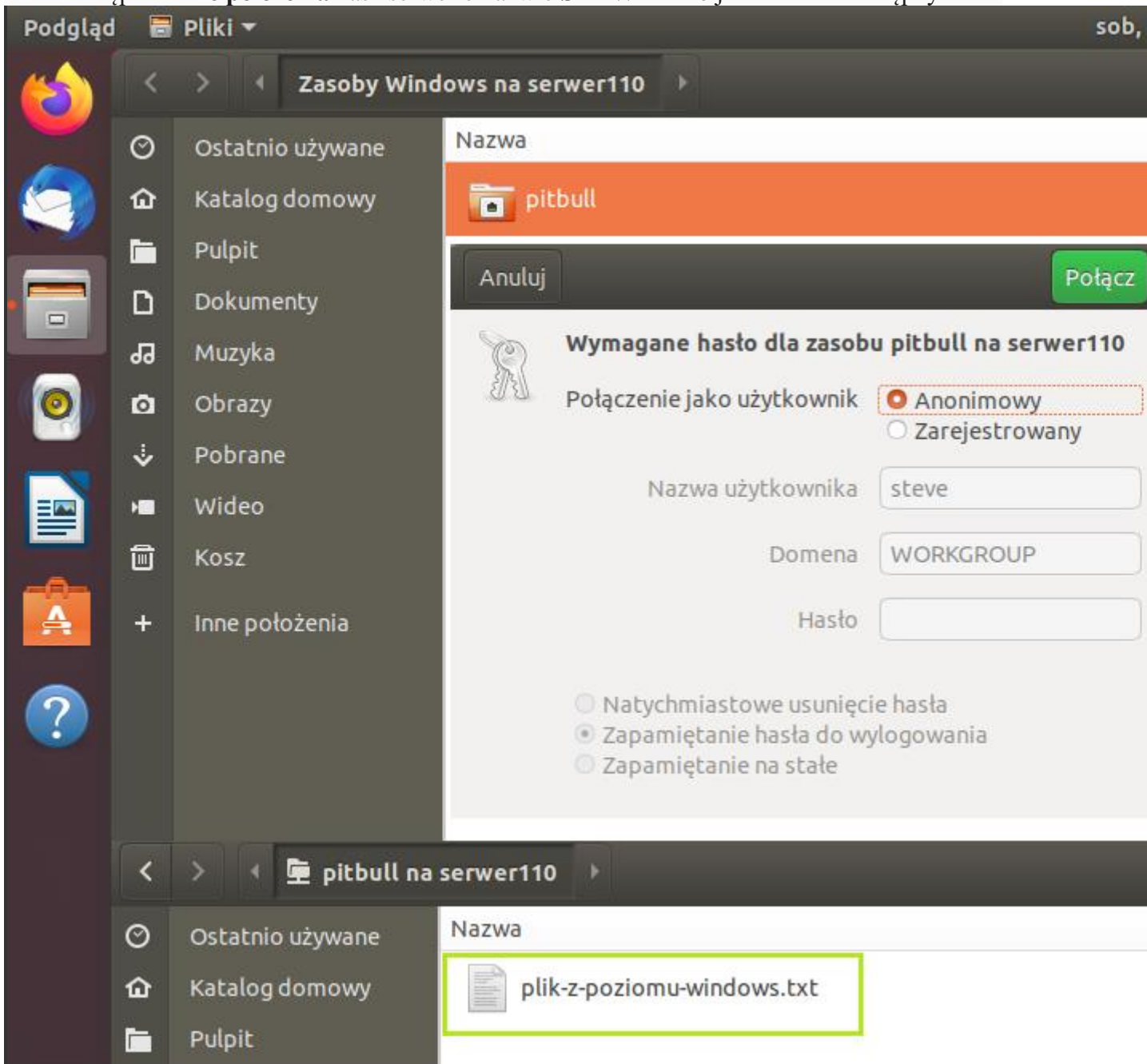
2.8

Wchodzimy do folderu **pitbull** i tworzymy jakiś plik



2.9

Podobnie postępujemy na drugim kliencie (z Ubuntu), tutaj po wybraniu opcji **Pliki**, a następnie **Inne położenia** nasz serwer o nazwie **SERWER110** jest również dostępny



2.10

Klikamy w **SERWER110**, następnie w folder **pitbull**, wybieramy dostęp anonimowy, klikamy **Połącz** i mamy dostęp do zasobu sieciowego

### III. Konfiguracja połączenia autoryzowanego dla jednego użytkownika.



```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo adduser bondjs
[sudo] password for administrator:
Adding user `bondjs' ...
Adding new group `bondjs' (1002) ...
Adding new user `bondjs' (1002) with group `bondjs' ...
Creating home directory `/home/bondjs' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bondjs
Enter the new value, or press ENTER for the default
  Full Name []: James Bond
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
administrator@serwer110:~$ _
```

**3.1** Aby istniała autoryzacja użytkownika w sambie, należy założyć konto w systemie oraz w sambie. Istnieje synchronizacja tych dwóch baz danych użytkowników, którą również należy przeprowadzić. Tworzymy konto **bondjs**

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo smbpasswd -a bondjs
New SMB password:
Retype new SMB password:
Added user bondjs.
administrator@serwer110:~$
```

**3.2** Następnie synchronizujemy naszego usera z sambą poleceniem **sudo smbpasswd -a bondjs**. Hasło nalepiej dać takie jak obowiązujące w systemie dla tego usera

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo mkdir /home/auto
administrator@serwer110:~$ sudo chmod 777 /home/auto
administrator@serwer110:~$ ls -l /home
total 20
drwxr-xr-x 4 administrator administrator 4096 Feb  2 16:28 administrator
drwxrwxrwx 2 root root 4096 Feb 20 20:42 auto
drwxr-xr-x 2 bondjs bondjs 4096 Feb 20 20:37 bondjs
drwxr-xr-x 3 informatykj informatykj 4096 Feb  2 16:48 informatykj
drwxrwxrwx 2 nobody nogroup 4096 Feb 20 19:09 pitbull
administrator@serwer110:~$
```

### 3.3

Utworzymy folder, który będzie zasobem sieciowym z dostępem autoryzowanym. Tworzymy poleceniem **sudo mkdir /home/auto**. Nadamy mu pełne uprawnienia dla wszystkich użytkowników **sudo chmod 777 /home/auto**. Po wyświetleniu zawartości folderu **/home** widzimy, że jego właścicielem jest **root** (grupa także). Możemy to zostawić

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
GNU nano 2.9.3 /etc/samba/smb.conf

# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

# Nasz folder sieciowy

[pitbull]
  path = /home/pitbull
  comment = pitbull-guest
  writable = yes
  browseable = yes
  guest ok = yes

# Kolejny folder sieciowy z dostępem autoryzowanym dla użytkownika bondjs

[auto]
  path = /home/auto
  comment = auto-bondjs
  writable = yes
  browseable = yes
  guest ok = no
  valid users = bondjs
```

### 3.4

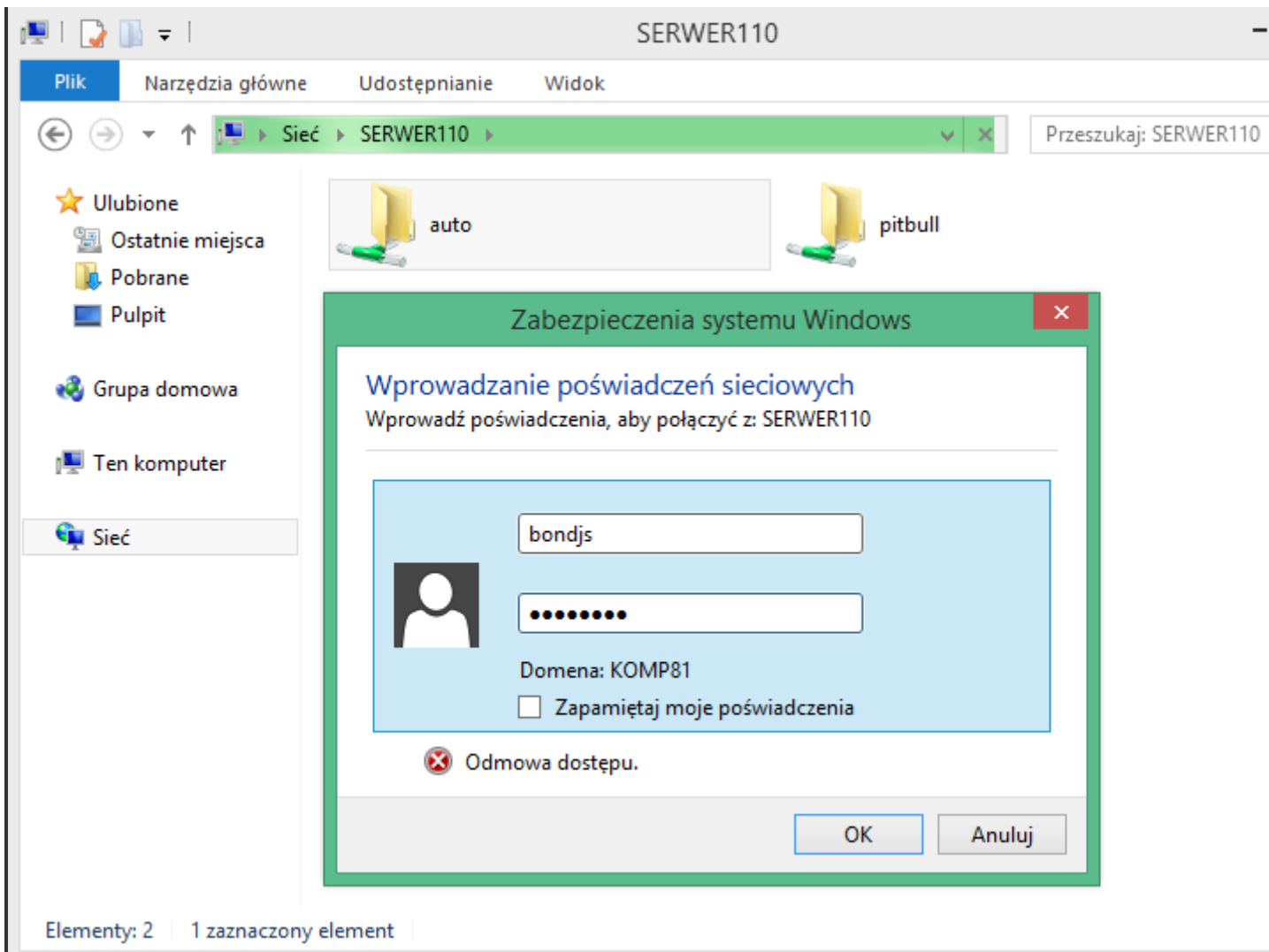
Teraz edytujemy plik konfiguracyjny: **sudo nano /etc/samba/smb.conf** i tam dokonamy takich zmian aby mieć dostęp autoryzowany. Przechodzimy na koniec pliku i dodajemy nasz nowy folder sieciowy z konkretnymi parametrami dokonując wpisów jak wyżej. Zapisujemy zmiany i zamykamy plik

```
administrator@serwer110:~$ sudo systemctl restart smbd
administrator@serwer110:~$ sudo systemctl status smbd
• smbd.service - Samba SMB Daemon
  Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2021-02-20 20:53:24 UTC; 8s ago
    Docs: man:smbd(8)
          man:samba(7)
          man:smb.conf(5)
 Main PID: 3754 (smbd)
  Status: "smbd: ready to serve connections..."
   Tasks: 4 (limit: 2317)
  CGroup: /system.slice/smbd.service
          └─3754 /usr/sbin/smbd --foreground --no-process-group
             └─3776 /usr/sbin/smbd --foreground --no-process-group
                └─3777 /usr/sbin/smbd --foreground --no-process-group
                   └─3778 /usr/sbin/smbd --foreground --no-process-group

Feb 20 20:53:24 serwer110 systemd[1]: Starting Samba SMB Daemon...
Feb 20 20:53:24 serwer110 systemd[1]: Started Samba SMB Daemon.
administrator@serwer110:~$ _
```

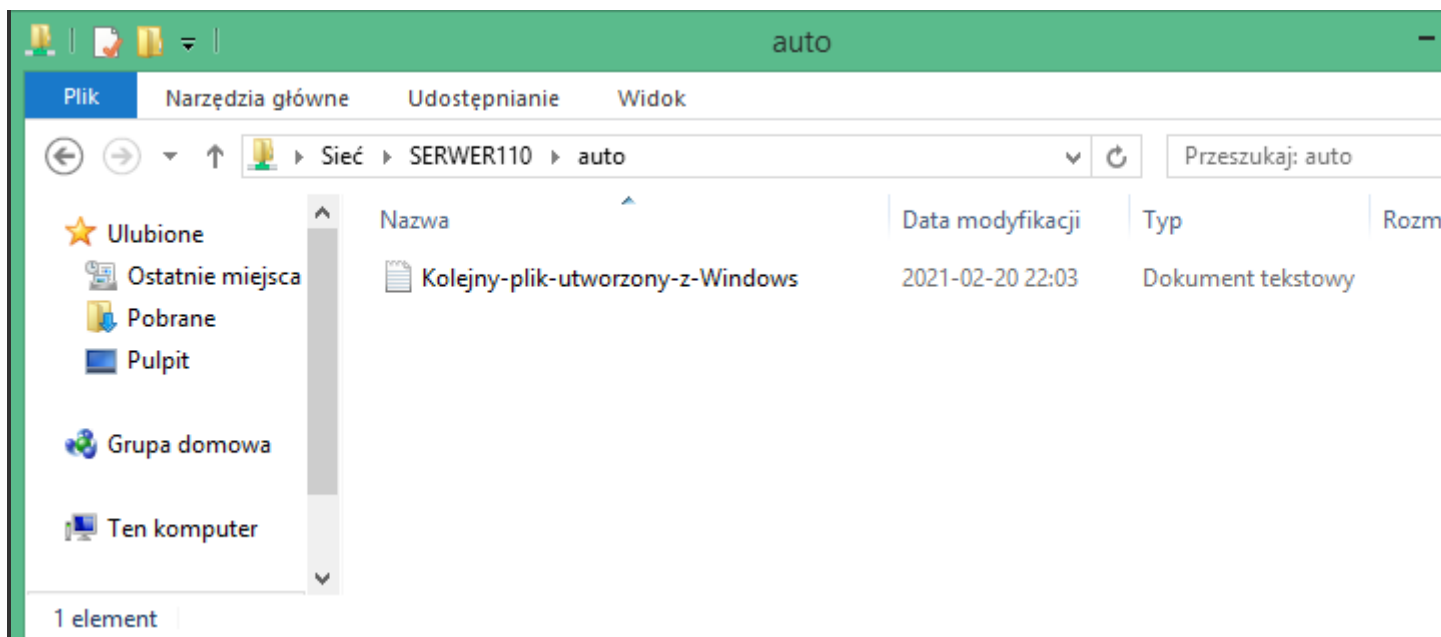
### 3.5

Następnie restartujemy naszą usługę poleceniem **sudo systemctl restart smbd** oraz sprawdzamy jego status. Działa!



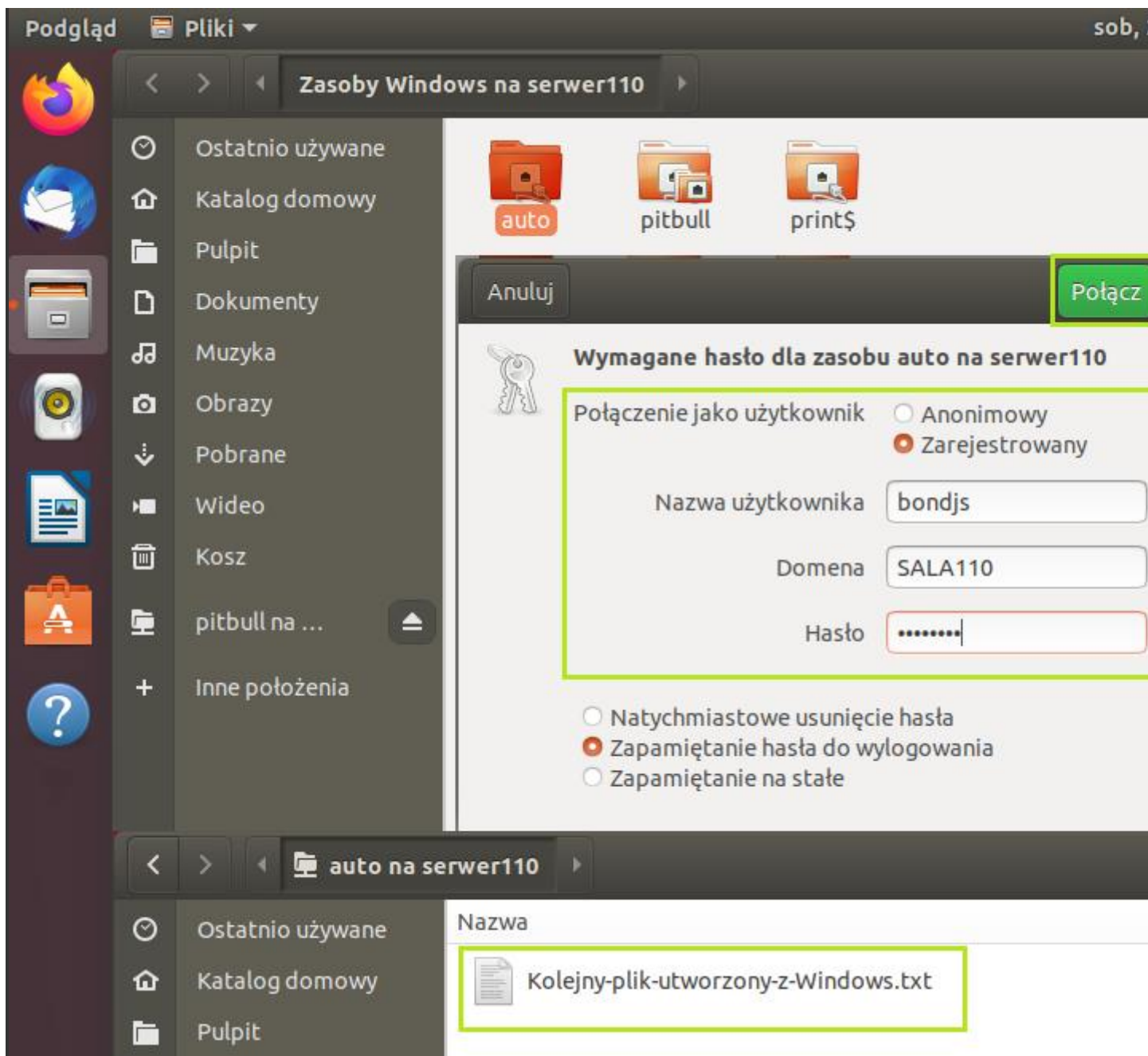
### 3.6

Ponownie sprawdzamy z poziomu klienta Windows. Nasz serwer jest widoczny, przechodzimy do niego i klikamy w folder **auto**. Oczywiście zostajemy poproszenie o autoryzację



3.7

Wchodzimy do folderu **auto** i tworzymy plik. Wszystko działa jak należy



3.8

Podobnie postępujemy na drugim kliencie (z Ubuntu). Klikamy w folder **auto**, wybieramy dostęp autoryzowany, podajemy login, grupę roboczą, hasło i klikamy **Połącz** i mamy dostęp do zasobu sieciowego z utworzonym plikiem

```
administrator@serwer110:~$ cd /home
administrator@serwer110:/home$ ls -l auto
total 0
-rwxr--r-- 1 bondjs bondjs 0 Feb 20 21:03 Kolejny-plik-utworzony-z-Windows.txt
administrator@serwer110:/home$ ls -l pitbull
total 0
-rwxr--r-- 1 nobody nogroup 0 Feb 20 19:09 plik-z-poziomu-windows.txt
administrator@serwer110:/home$
```

### 3.9

Oczywiście z poziomu serwera też możemy sprawdzić zawartość tych folderów. Przechodzimy do katalogu `/home` i używając polecenia `ls -l nazwa_folderu` wyświetlamy ich zawartość

## IV. Konfiguracja połączenia autoryzowanego dla kilku użytkowników.

```
administrator@serwer110:~$ sudo adduser bondj2s
[sudo] password for administrator:
Adding user `bondj2s' ...
Adding new group `bondj2s' (1003) ...
Adding new user `bondj2s' (1003) with group `bondj2s' ...
Creating home directory `/home/bondj2s' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bondj2s
Enter the new value, or press ENTER for the default
  Full Name []: James Bondj2
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
administrator@serwer110:~$ sudo adduser bondj3s
Adding user `bondj3s' ...
Adding new group `bondj3s' (1004) ...
Adding new user `bondj3s' (1004) with group `bondj3s' ...
Creating home directory `/home/bondj3s' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for bondj3s
Enter the new value, or press ENTER for the default
  Full Name []: James Bondj3
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
administrator@serwer110:~$
```

#### 4.1

Tym razem tworzymy przynajmniej dwa konta w systemie znanym już poleceniem **sudo adduser nazwa\_użytkownika**. Utworzymy konta **bondj2s** i **bondj3s**



```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo smbpasswd -a bondj2s
New SMB password:
Retype new SMB password:
Added user bondj2s.
administrator@serwer110:~$ sudo smbpasswd -a bondj3s
New SMB password:
Retype new SMB password:
Added user bondj3s.
administrator@serwer110:~$ _
```

#### 4.2

Następnie synchronizujemy ich z sambą poleceniem **sudo smbpasswd -a nazwa\_użytkownika**

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo addgroup wataha
Adding group `wataha' (GID 1005) ...
Done.
administrator@serwer110:~$ sudo usermod -G wataha -a bondj2s
administrator@serwer110:~$ sudo usermod -G wataha -a bondj3s
administrator@serwer110:~$
```

#### 4.3

Tworzymy również grupę **wataha**, a następnie przypisujemy do niej naszych użytkowników poleceniem **sudo usermod -G wataha -a nazwa\_użytkownika**

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
GNU nano 2.9.3 /etc/group
video:x:44:
sasl:x:45:
plugdev:x:46:administrator
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
input:x:104:
crontab:x:105:
syslog:x:106:
messagebus:x:107:
lxd:x:108:administrator
mlocate:x:109:
uudd:x:110:
ssh:x:111:
landscape:x:112:
administrator:x:1000:
informatykj:x:1001:
rdma:x:113:
sambashare:x:114:
bondjs:x:1002:
bondj2s:x:1003:
bondj3s:x:1004:
wataha:x:1005:bondj2s,bondj3s
```

**4.4** Sprawdzamy plik **group** i widzimy, że nasi użytkownicy są przypisani do właściwej grupy

```
# Kolejny folder sieciowy z dostępem autoryzowanym dla użytkownika bondjs
```

```
[auto]
```

```
path = /home/auto  
comment = auto-bondjs  
writable = yes  
browseable = yes  
guest ok = no  
valid users = bondjs
```

```
# Jeszcze jeden folder sieciowy dla grupy wataha
```

```
[wataha]
```

```
path = /home/wataha  
comment = grupa-wataha  
writable = yes  
browseable = yes  
guest ok = no  
valid users = @wataha  
create mask = 0755  
directory mask = 0770
```

#### 4.5

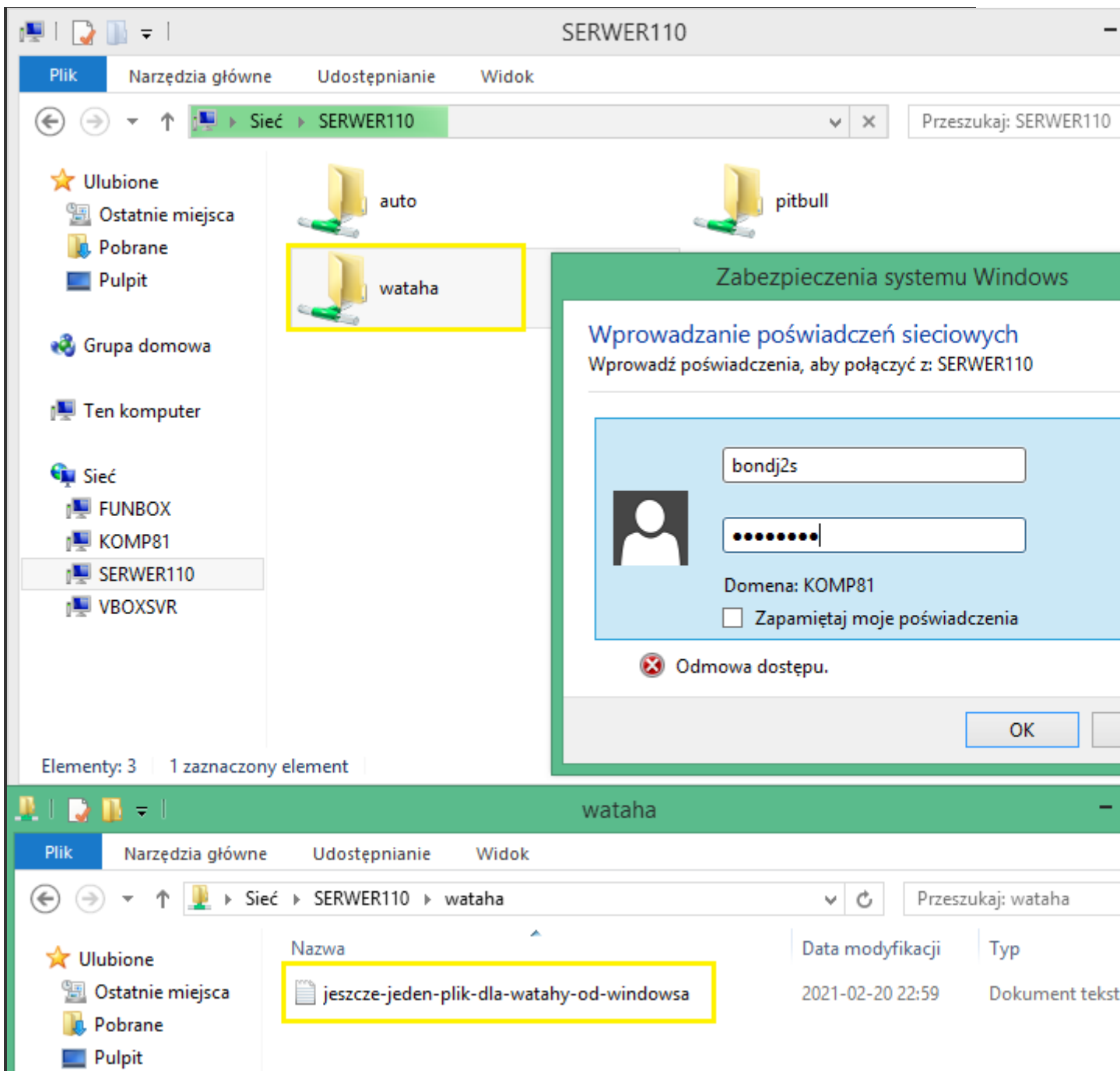
Teraz modyfikujemy plik **smb.conf**. Edytujemy go znanym już poleceniem **sudo nano /etc/samba/smb.conf** i tam dokonamy modyfikacji nadając nazwę udziałowi również **wataha**

```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
administrator@serwer110:~$ sudo mkdir -m 777 /home/wataha
administrator@serwer110:~$ ls -l /home
total 32
drwxr-xr-x 4 administrator administrator 4096 Feb  2 16:28 administrator
drwxrwxrwx 2 root root 4096 Feb 20 21:03 auto
drwxr-xr-x 2 bondj2s bondj2s 4096 Feb 20 21:24 bondj2s
drwxr-xr-x 2 bondj3s bondj3s 4096 Feb 20 21:25 bondj3s
drwxr-xr-x 2 bondjs bondjs 4096 Feb 20 20:37 bondjs
drwxr-xr-x 3 informatykj informatykj 4096 Feb  2 16:48 informatykj
drwxrwxrwx 2 nobody nogroup 4096 Feb 20 19:09 pitbull
drwxrwxrwx 2 root root 4096 Feb 20 21:46 wataha
administrator@serwer110:~$ sudo systemctl restart smbd
administrator@serwer110:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-02-20 21:47:43 UTC; 5s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 3913 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 5 (limit: 2317)
    CGroup: /system.slice/smbd.service
            └─3913 /usr/sbin/smbd --foreground --no-process-group
              └─3935 /usr/sbin/smbd --foreground --no-process-group
                └─3936 /usr/sbin/smbd --foreground --no-process-group
                  └─3939 /usr/sbin/smbd --foreground --no-process-group
                    └─3941 /usr/sbin/smbd --foreground --no-process-group

Feb 20 21:47:43 serwer110 systemd[1]: Stopped Samba SMB Daemon.
Feb 20 21:47:43 serwer110 systemd[1]: Starting Samba SMB Daemon...
Feb 20 21:47:43 serwer110 systemd[1]: Started Samba SMB Daemon.
Feb 20 21:47:44 serwer110 smbd[3941]: pam_unix(samba:session): session opened for user bo
lines 1-20/20 (END)
administrator@serwer110:~$
```

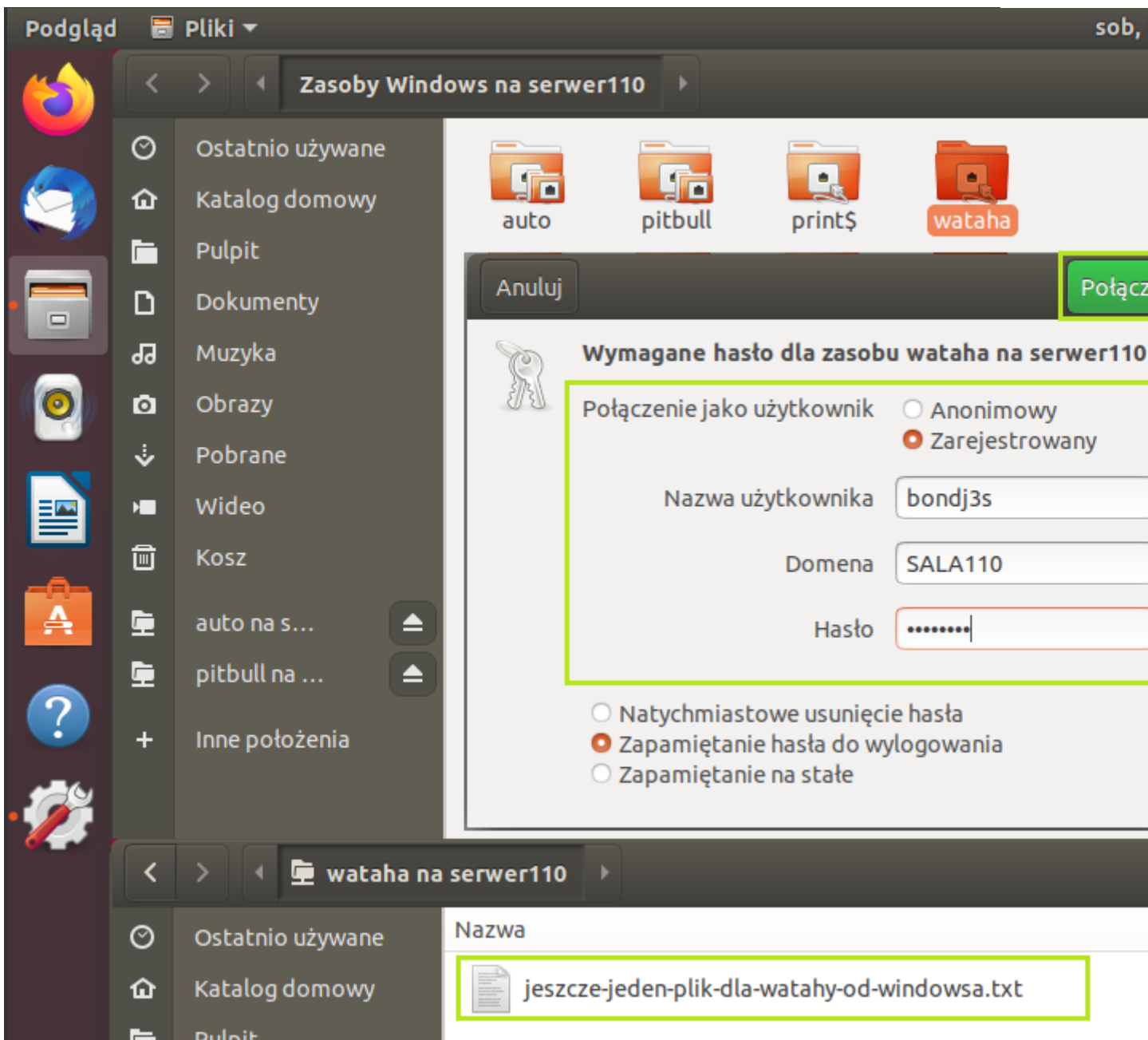
#### 4.6

Musimy jeszcze utworzyć nasz folder. Przy tworzeniu nadamy mu pełne prawa dostępu dla wszystkich użytkowników. Wykonujemy polecenie **sudo mkdir -m 777 /home/wataha**. Sprawdzamy prawa, restartujemy SAMBĘ i widzimy, że jest ok



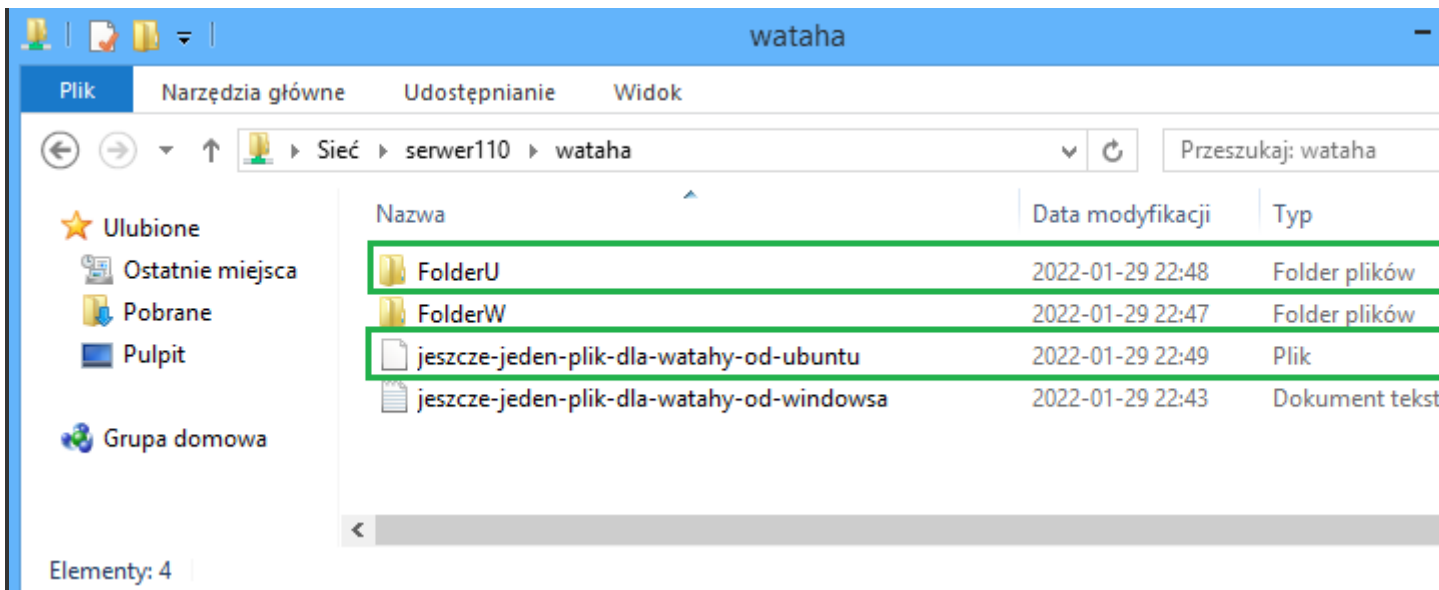
4.7

Sprawdzamy z poziomu klienta Windows. Warto się wylogować z systemu i zalogować ponownie. Zaznaczamy **Sieć** potem **SERWER110**, klikamy w folder **wataha** i autoryzujemy dostęp. Wchodzimy do folderu **wataha** i tworzymy plik. Wszystko działa jak należy



4.8

Podobnie postępujemy na drugim kliencie (z Ubuntu). Klikamy w folder **wataha**, wybieramy dostęp autoryzowany (drugim użytkownikiem), podajemy login, grupę roboczą, hasło i klikamy **Połącz** i mamy dostęp do zasobu sieciowego



#### 4.9

Teraz utworzymy folder i plik z klienta Ubuntu oraz drugi folder z klienta Windows (plik już mamy)

```
administrator@serwer110:/home$ ls -l wataha
total 8
drwxrwx--- 2 bondj3s bondj3s 4096 Jan 29 21:48 FolderU
drwxrwx--- 2 bondj2s bondj2s 4096 Jan 29 21:47 FolderW
-rw-r--r-- 1 bondj3s bondj3s  0 Jan 29 21:49 jeszcze-jeden-plik-dla-watahy-od-ubuntu
-rwxr--r-- 1 bondj2s bondj2s  0 Jan 29 21:43 jeszcze-jeden-plik-dla-watahy-od-windowsa.
administrator@serwer110:/home$ _
```

#### 4.10

Następnie z poziomu serwera sprawdzamy zawartość folderu **wataha**. Widzimy, że właścicielem danych z Ubuntu jest **bondj3s**, a danych z Windowsa **bondj2s**, czyli tak jak się logowaliśmy. Dodatkowo uprawnienia do tworzonych folderów to 770, a do plików 755, a raczej 644 - ponieważ przy tworzonych plikach, SAMBA jakby nie uwzględnia opcji **x** (wykonanie).

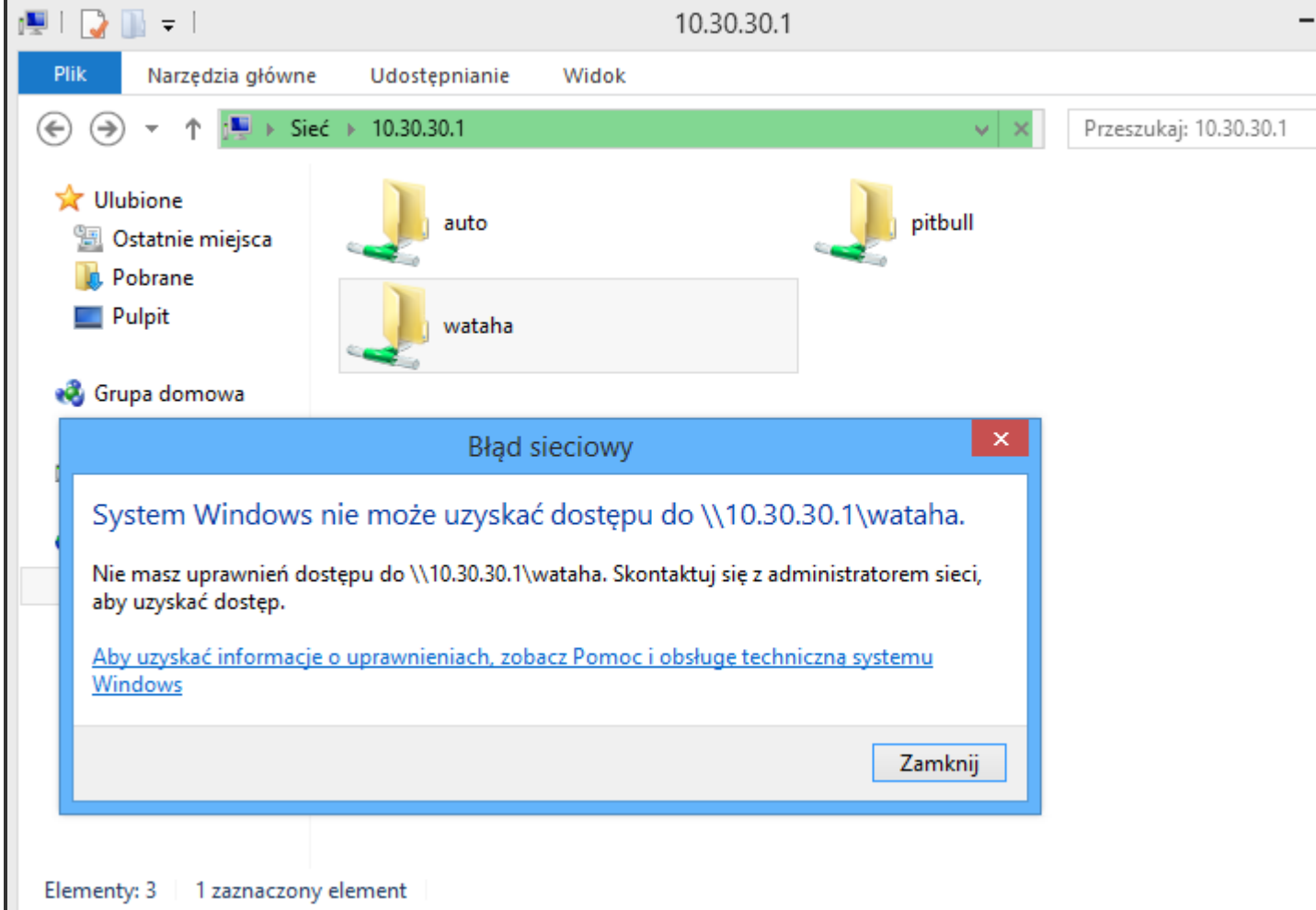
```
Ubuntu-S-18-04 [Uruchomiona] - Oracle VM VirtualBox
Plik Maszyna Widok Wejście Urządzenia Pomoc
GNU nano 2.9.3 /etc/samba/smb.conf

# Jeszcze jeden folder sieciowy dla grupy wataha

[wataha]
path = /home/wataha
comment = grupa-wataha
writable = yes
browseable = yes
guest ok = no
valid users = @wataha
create mask = 0755
directory mask = 0770
invalid users = bondj2s
```

**4.11**

Możemy też na poziomie SAMB-y regulować uprawnieniami. Aby komuś zabronić dostępu (nawet z danej grupy, która posiada uprawnienia do zasobu) wystarczy użyć opcji **invalid users = bondj2s**



**4.12**

I sprawdzamy teraz dostęp użytkownikiem **bondj2s**. Oczywiście dostępu brak

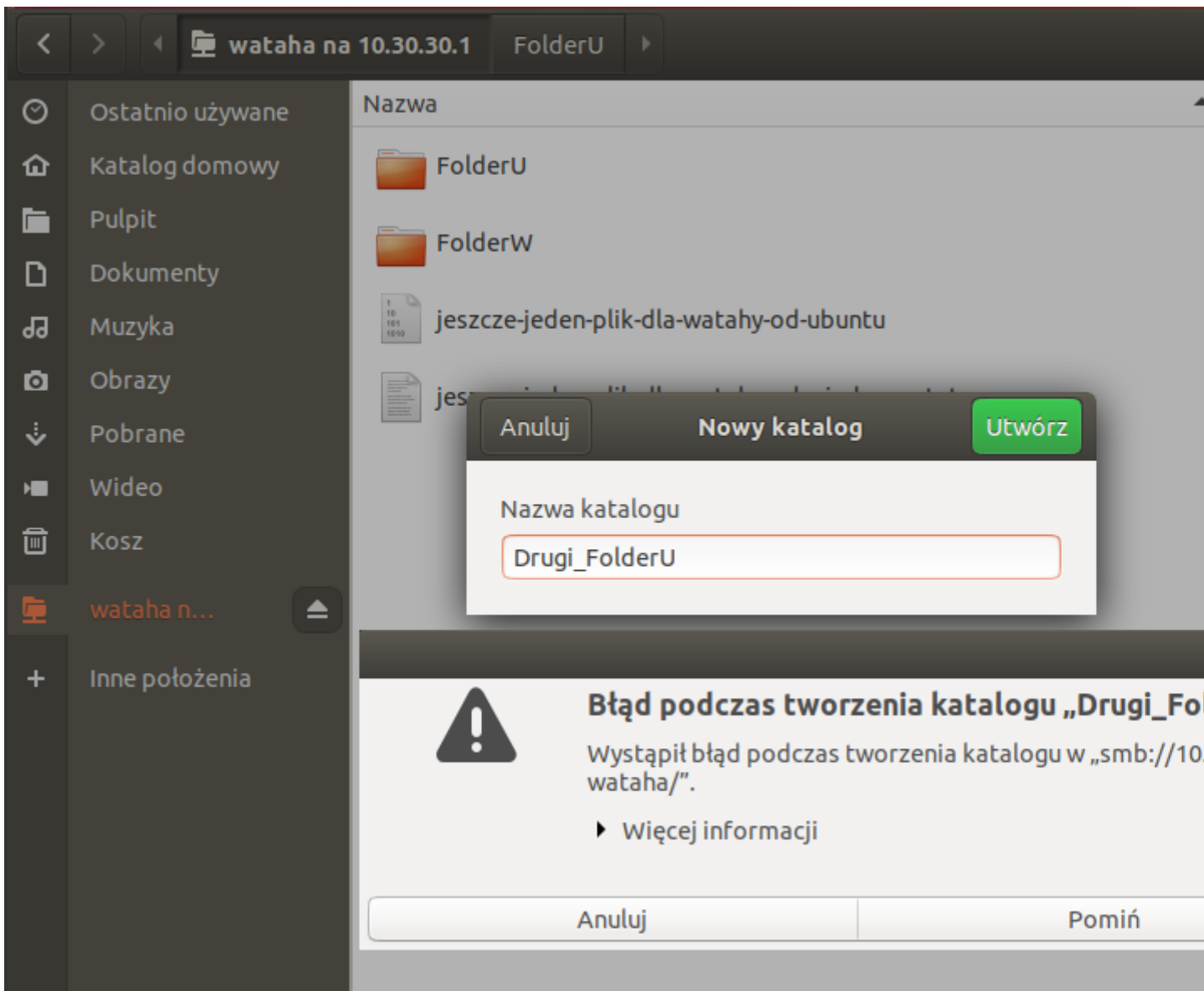


```
# Jeszcze jeden folder sieciowy dla grupy wataha
```

```
[wataha]
  path = /home/wataha
  comment = grupa-wataha
  writable = yes
  browseable = yes
  guest ok = no
  valid users = @wataha
  create mask = 0755
  directory mask = 0770
  invalid users = bondj2s
  read list = bondj3s
```

#### 4.13

Możemy też użyć opcji **write list** lub **read list**, a następnie podać użytkowników, którym chcemy nadać takie uprawnienia. Nadamy więc użytkownikowi **bondj3s** tylko odczyt do naszego zasobu



4.14

I teraz z poziomu klienta Ubuntu, będąc zalogowanym **bondj3s** sprawdzamy czy uda nam się utworzyć jakiś folder. Oczywiście dostajemy odmowę

© 2009- 2018 Copyright **ZSE Rzeszów** | Developed by **Maciej Biel & Ireneusz Paździor**