

9. Prezentacja działania aplikacji

9.1. Uruchomienie aplikacji SMMC

9.1.1. Przygotowanie serwera głównego

Ze względu na wymagania środowiskowe [ŚD03], serwer główny musi być obsługiwany przez system z rodziny Linux. Następnie należy wykonać następujące kroki:

- Utworzenie folderu roboczego

```
[parallels@kali-linux-2021-3] ~]$ mkdir SMMC  
[parallels@kali-linux-2021-3] ~]$ cd SMMC  
[parallels@kali-linux-2021-3] ~/SMMC]$
```

Rys. 9.1 Utworzenie folderu roboczego na serwerze głównym

W tym miejscu będą znajdować się wszystkie pliki odpowiedzialne za działanie aplikacji po stronie serwera monitorującego.

- Instalacja języka programowania Python3

```
[parallels@kali-linux-2021-3] ~/SMMC]$ sudo apt-get install python3  
[sudo] password for parallels:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3 is already the newest version (3.9.2-3).  
python3 set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Rys. 9.2 Instalacja języka programowania Python3

Większość współczesnych dystrybucji systemu Linux posiada zainstalowany język Python3 w standardzie. Warto jednak wykonać ten krok aby mieć pewność co do instalacji.

- Klonowanie kodu z GitHub do lokalizacji lokalnej

```
[parallels@kali-linux-2021-3] ~/SMMC]$ git clone https://github.com/SMMCproject/SMMC.git  
Cloning into 'SMMC'...  
Username for 'https://github.com': s16740@jwstk.edu.pl  
Password for 'https://s16740@jwstk.edu.pl@github.com':  
remote: Enumerating objects: 691, done.  
remote: Counting objects: 100% (691/691), done.  
remote: Compressing objects: 100% (304/304), done.  
remote: Total 691 (delta 435), reused 620 (delta 366), pack-reused 0  
Receiving objects: 100% (691/691), 2.51 MiB | 8.55 MiB/s, done.  
Resolving deltas: 100% (435/435), done.
```

Rys. 9.3 Kopiowanie kodu z repozytorium GitHub

Przez charakterystykę projektu SMMC, do skopiowania kodu źródłowego, wymagane jest zalogowanie się upoważnionym kontem serwisu GitHub.

- Instalowanie środowiska wirtualnego dla Python3

```
[parallels@kali-linux-2021-3] ~/SMMC]$ sudo apt-get install python3-venv  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
libc-devtools libgdal29 libodbc1 libodbccr2 libpython3.9-dev libqhull8.0 libyara8 odbcinst odbcinstdebian2 python-mpltoolkits.basemap-data python3-pyproj python3-psycopg2 python3.9-dev
```

Rys. 9.4 Instalowanie środowiska wirtualnego dla Python3

- Tworzenie środowiska wirtualnego dla aplikacji SMMC

```
[parallels@kali-linux-2021-3]~/SMMC]$ python3 -m venv venv
[parallels@kali-linux-2021-3]~/SMMC]$ ls
SMMC venv
```

Rys. 9.5 Tworzenie środowiska wirtualnego dla aplikacji SMMC

Efektem tego kroku jest utworzenie kolejnego katalogu, w tym wypadku nazywa się on „venv”. W tym miejscu znajdują się kluczowe, dla poprawnego działania środowiska wirtualnego, pliki binarne.

- Aktywowanie środowiska wirtualnego

```
[parallels@kali-linux-2021-3]~/SMMC]$ source venv/bin/activate
(venv) [parallels@kali-linux-2021-3]~/SMMC$
```

Rys. 9.6 Aktywowanie środowiska wirtualnego

Po aktywowaniu środowiska wirtualnego, poruszamy się w przestrzeni „venv” – można to rozpoznać po stosownym napisie obok nazwy użytkownika.

- Instalowanie modułów niezbędnych do poprawnego działania SMMC

```
(venv) [parallels@kali-linux-2021-3]~/SMMC]$ pip install -r SMMC/requirements.txt
Collecting ansible==5.6.0
  Downloading ansible-5.6.0.tar.gz (35.5 MB)
    Preparing metadata (setup.py) ... done
    Collecting ansible-core==2.12.4
      Downloading ansible-core-2.12.4.tar.gz (7.8 MB)
        Preparing metadata (setup.py) ... done
    Collecting Flask==2.0.2
      Downloading Flask-2.0.2-py3-none-any.whl (95 kB)
        Preparing metadata (setup.py) ... done
    Collecting Flask-Login==0.5.0
      Downloading Flask-Login-0.5.0-py2.py3-none-any.whl (16 kB)
    Collecting FlaskAlchemy==2.5.1
      Downloading Flask_SQLAlchemy-2.5.1-py2.py3-none-any.whl (17 kB)
    Collecting Jinja2==3.0.3
      Downloading Jinja2-3.0.3-py3-none-any.whl (133 kB)
        Preparing metadata (setup.py) ... done
    Collecting multiprocessing==0.70.12.2
      Downloading multiprocessing-0.70.12.2-py39-none-any.whl (128 kB)
        Preparing metadata (setup.py) ... done
    Collecting psutil==5.9.0
      Downloading psutil-5.9.0.tar.gz (478 kB)
        Preparing metadata (setup.py) ... done
    Collecting pygal==3.0.0
      Downloading pygal-3.0.0-py2.py3-none-any.whl (129 kB)
        Preparing metadata (setup.py) ... done
    Collecting PyYAML==6.0
      Downloading PyYAML-6.0-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (733 kB)
        Preparing metadata (setup.py) ... done
    Collecting SQLAlchemy==1.4.29
      Downloading SQLAlchemy-1.4.29-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (1.6 MB)
        Preparing metadata (setup.py) ... done
    Collecting Werkzeug==2.0.2
      Downloading Werkzeug-2.0.2-py3-none-any.whl (288 kB)
        Preparing metadata (setup.py) ... done
    Collecting cryptography
      Downloading cryptography-37.0.2-cp36abi3-manylinux_2_17_aarch64.manylinux2014_aarch64.manylinux_2_24_aarch64.whl (3.6 MB)
```

Rys. 9.7 Instalowanie modułów niezbędnych dla aplikacji SMMC

Wymagane moduły do poprawnego działania aplikacji SMMC są zapisane w pliku requirements.txt, który znajduje się w folderze głównym SMMC (SMMC/requirements.txt).

- Uruchomienie aplikacji SMMC

```

└─(venv)─(parallels㉿kali-linux-2021-3)─[~/SMMC]
$ cd SMMC

└─(venv)─(parallels㉿kali-linux-2021-3)─[~/SMMC/SMMC]
$ python3 main.py
* Serving Flask app 'mainwebsite' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.0.14:5050/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 683-253-257

```

Rys. 9.8 Uruchomienie aplikacji SMMC

Jest to ostatnia komenda potrzebna do uruchomienia SMMC na serwerze monitorującym. Od teraz w terminalu będą zapisywane logi aplikacji, a strona internetowa, jest już aktywna pod adresem podanym w logach.



Rys. 9.9 Strona główna poprawnie uruchomionego SMMC

9.1.2. Przygotowanie klientów Linux

9.1.2.1. Tworzenie konta administratora

Do spełnienia wymagania [ŚD02] należy przygotować odpowiednie konto użytkownika z uprawnieniami „root”, w tym przykładzie w systemie Kali Linux. Komendy mogą się różnić między dystrybucjami Linux. W tym celu należy wykonać następujące kroki:

```

parallels㉿kali-linux-2021-3:~)
$ sudo adduser admin
[sudo] password for parallels:
Adding user `admin' ... Adding new group `admin' (1001) ...
Adding new user `admin' (1001) with group `admin' ...
Creating home directory `/home/admin' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for admin
Enter the new value, or press ENTER for the default
  Full Name []:
    Room Number []:
      Work Phone []:
        Home Phone []:
          Other []
Is the information correct? [Y/n] Y

```

Rys. 9.10 Dodawanie nowego użytkownika w systemie Kali Linux

Następnie nowego użytkownika należy dodać do tzw. „grupy sudo” i przypisać do środowiska Shell.

```
[└─(parallels㉿kali-linux-2021-3)-[~]
└─$ sudo usermod -a -G sudo admin
└─(parallels㉿kali-linux-2021-3)-[~]
└─$ sudo chsh -s /bin/zsh admin
```

Rys. 9.11 Dodanie użytkownika do grupy sudo i przypisanie Shell

Teraz możliwe powinno być zalogowanie jako nowy użytkownik, można to potwierdzić następującymi komendami;

```
[└─(parallels㉿kali-linux-2021-3)-[~]
└─$ su admin
Password:
└─(admin㉿kali-linux-2021-3)-[/home/parallels]
└─$ whoami
admin
└─(admin㉿kali-linux-2021-3)-[/home/parallels]
└─$ █
```

Rys. 9.12 Potwierdzenie działania nowego konta użytkownika

9.1.2.2. Konfiguracja połączeń SSH

Do komunikacji Ansible ↔ Linux wykorzystywany jest protokół SSH [ŚD05], który wymaga wcześniejszej konfiguracji dla użytkownika zarządzającego serwerami. W tym scenariuszu klucz SSH powinien zostać wygenerowany na urządzeniu, które potem będzie serwerem monitorującym.

```
[adamtomporowski@MacBook-Pro-Adam ~ % ssh-keygen -t ed25519 -C "ansible"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/Users/adamtomporowski/.ssh/id_ed25519): /Users/adamtomporowski/.ssh/ansible2
[Enter passphrase (empty for no passphrase):
[Enter same passphrase again:
Your identification has been saved in /Users/adamtomporowski/.ssh/ansible2
Your public key has been saved in /Users/adamtomporowski/.ssh/ansible2.pub
The key fingerprint is:
SHA256:w2J2Q+k/3pV7vztFAmV0qmZ8JQaF+3N6BziI9j4euMc ansible
The key's randomart image is:
++-[ED25519 256]-
|       .==.|
|       .oo o |
|       o o+.|
|       + ..o.o.|
|       + S. .=o.o |
|       o oo=.oo.+o|
|       ..o+ .o=|
|       oE+ ..o+|
|       .++... .=B|
+---[SHA256]---+
adamtomporowski@MacBook-Pro-Adam ~ % █
```

Rys. 9.13 Generowanie klucza SSH

Bardzo ważne jest, aby nie definiować tzw. „passphrase”, dlatego by Ansible mógł wykonywać zadania automatycznie.

Kolejnym krokiem konfiguracji jest przekopiowanie wygenerowanego klucza, na serwer, który ma być objęty dostępem zdalnym.

```
[adamtomporowski@MacBook-Pro-Adam monitoring % ssh-copy-id -i ~/.ssh/ansible.pub admin@192.168.0.132
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/adamtomporowski/.ssh/ansible.pub"
/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 prompted now it is to install the new keys
[admin@192.168.0.132's password:

Number of key(s) added:      1

Now try logging into the machine, with:  "ssh 'admin@192.168.0.132'"
and check to make sure that only the key(s) you wanted were added.

adamtomporowski@MacBook-Pro-Adam monitoring %
```

Rys. 9.14 Kopiowanie klucza SSH na serwer zdalny

W tym momencie serwer zdalny będzie akceptował połączenia SSH, w tym również te sterowane przez Ansible. Można to zweryfikować w następujący sposób:

```
[adamtomporowski@MacBook-Pro-Adam ~ % ssh admin@192.168.0.132
[admin@192.168.0.132's password:
Linux kali-linux-2021-3 5.14.0-kali2-arm64 #1 SMP Debian 5.14.9-2kali1 (2021-10-04) aarch64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jun  5 10:18:06 2022 from 192.168.0.59
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
→ https://www.kali.org/docs/general-use/python3-transition/
(Run: "touch ~/.hushlogin" to hide this message)
(admin@kali-linux-2021-3) -[~]
$
```

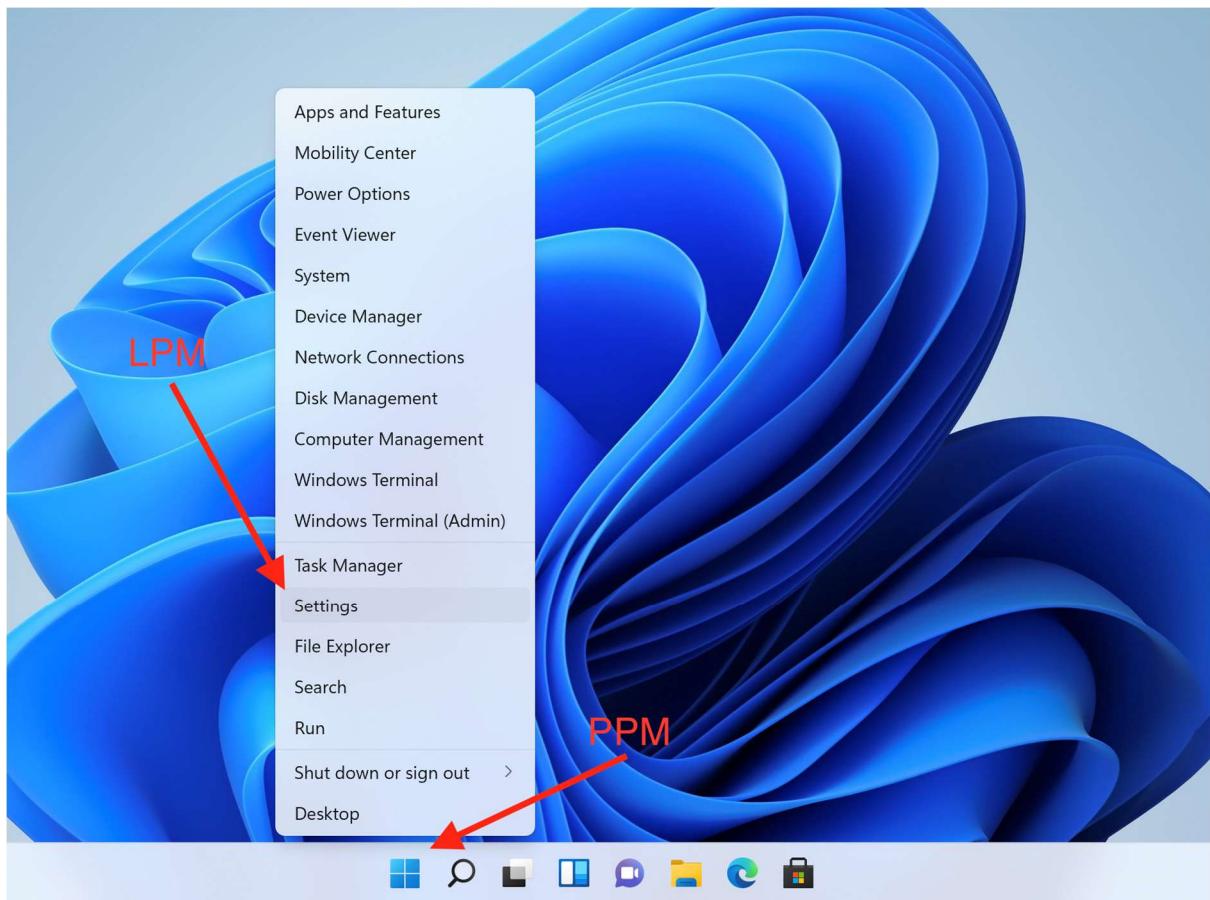
Rys. 9.15 Potwierdzenie poprawności konfiguracji SSH

9.1.3. Przygotowanie klientów Windows

9.1.3.1. Tworzenie konta administratora

Do spełnienia wymagania [ŚD02] należy przygotować odpowiednie konto administratora, w tym przykładzie w systemie Windows 11. Kroki będą podobne we wszystkich współczesnych systemach Windows. W tym celu należy wykonać kolejne czynności:

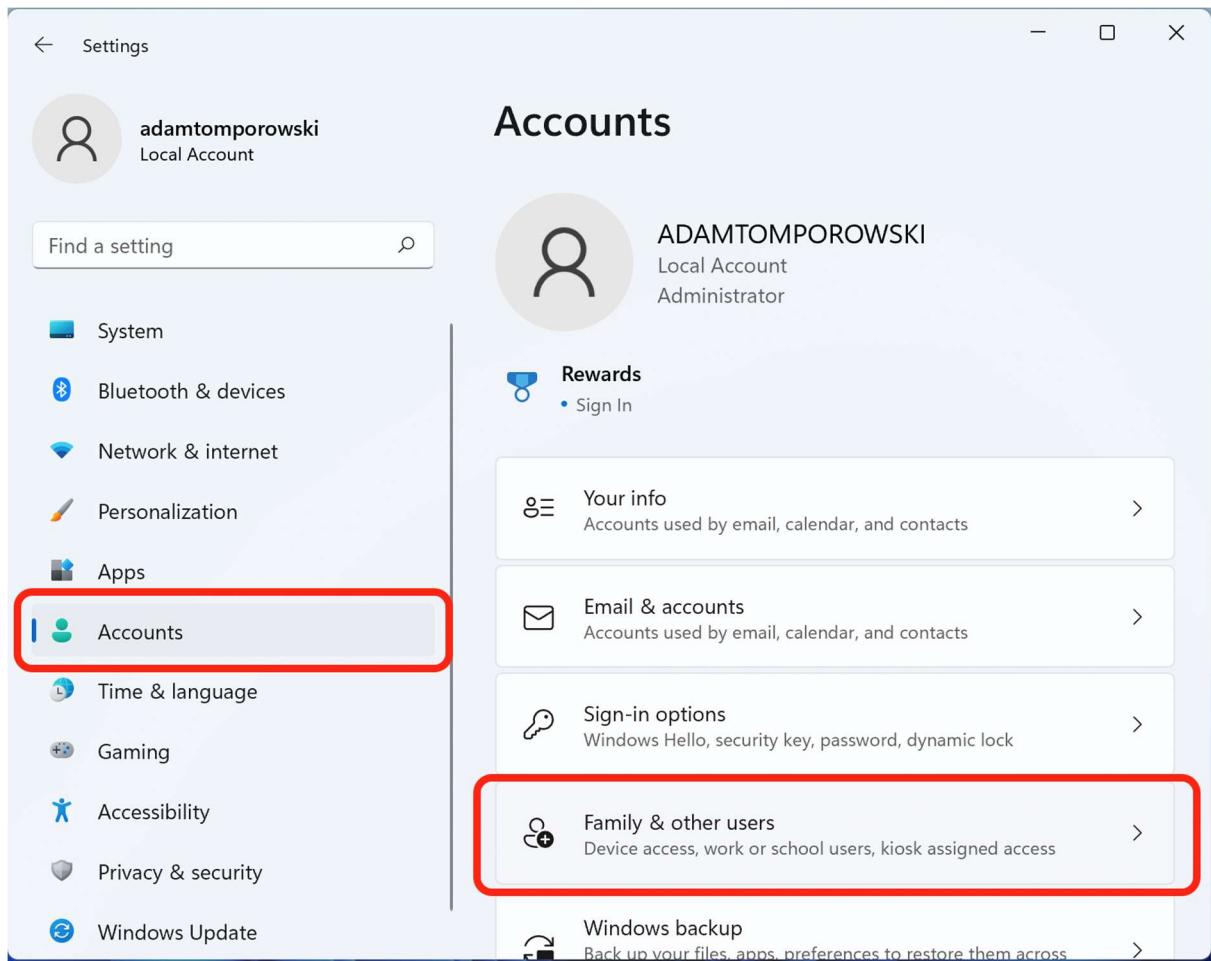
- Uruchomienie ustawień systemu



Rys. 9.16 Uruchomienie ustawień w systemie Windows 11

Kliknięcie prawym przyciskiem myszy na ikonie menu Start uruchomi menu kontekstowe, z którego należy wybrać pozycje „Settings”.

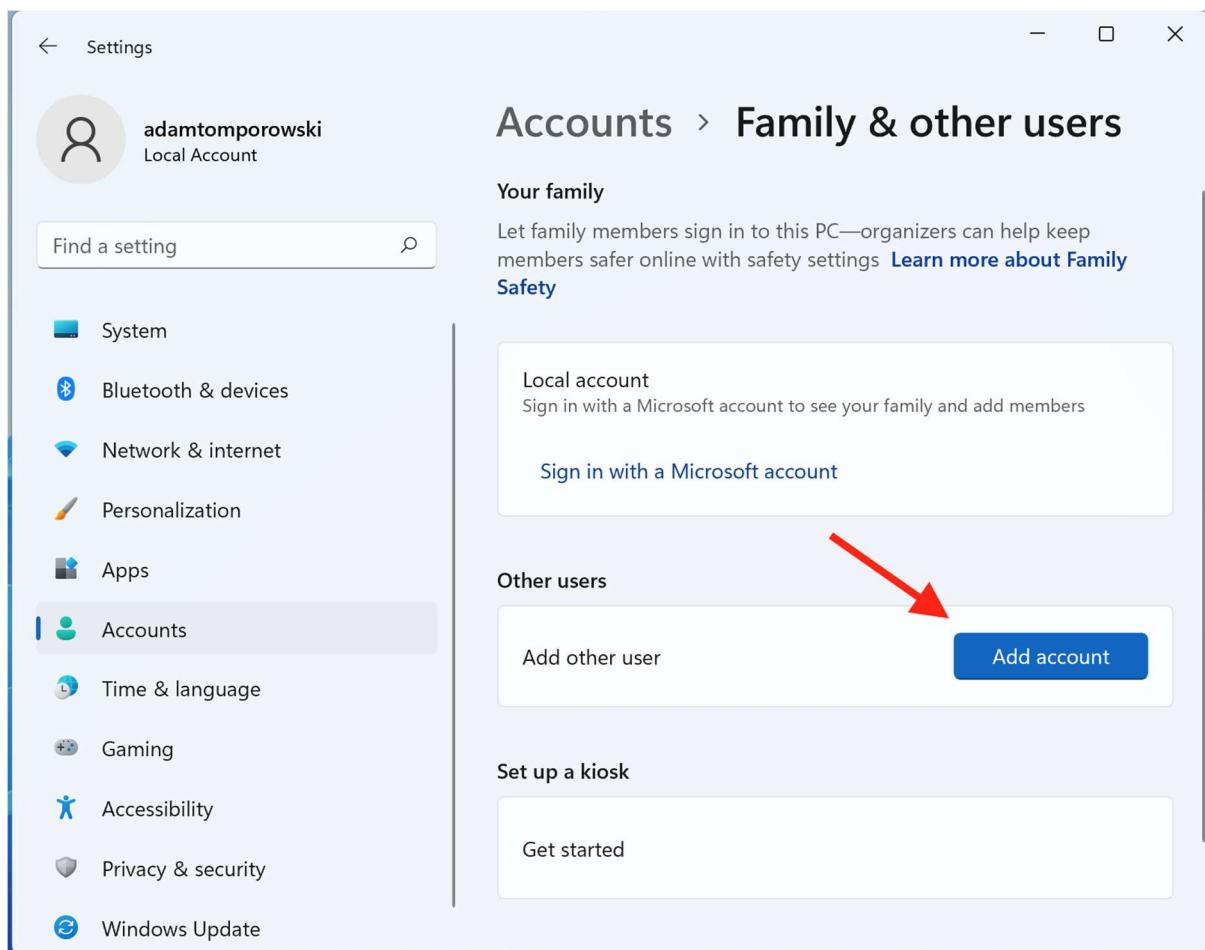
- Uruchomienie ustawień kont użytkowników



Rys. 9.17 Uruchomienie ustawień kont użytkowników

Z nowo otwartego okna należy wybrać kolejno „Accounts”, a potem „Family & other users”.

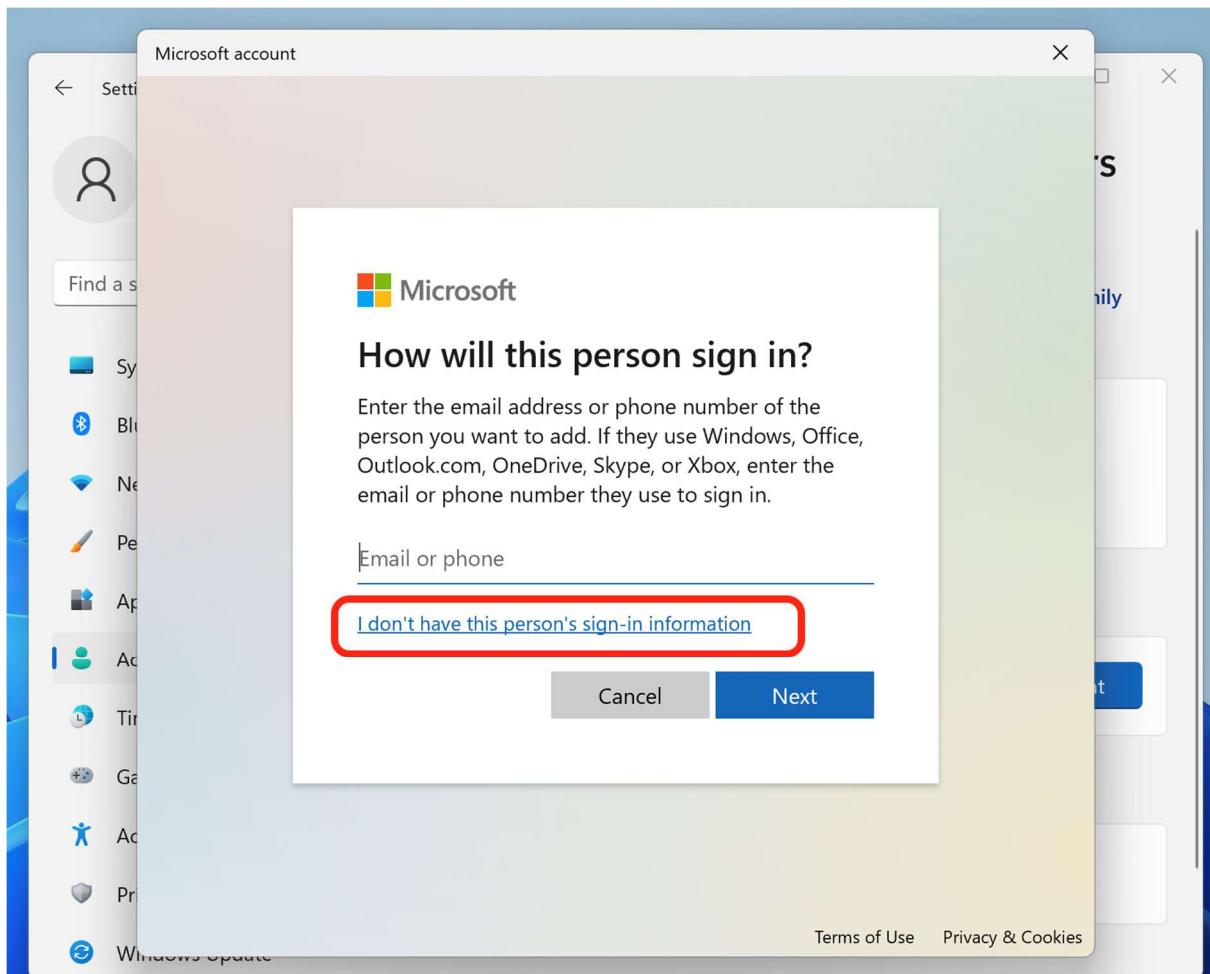
- Tworzenie konta użytkownika



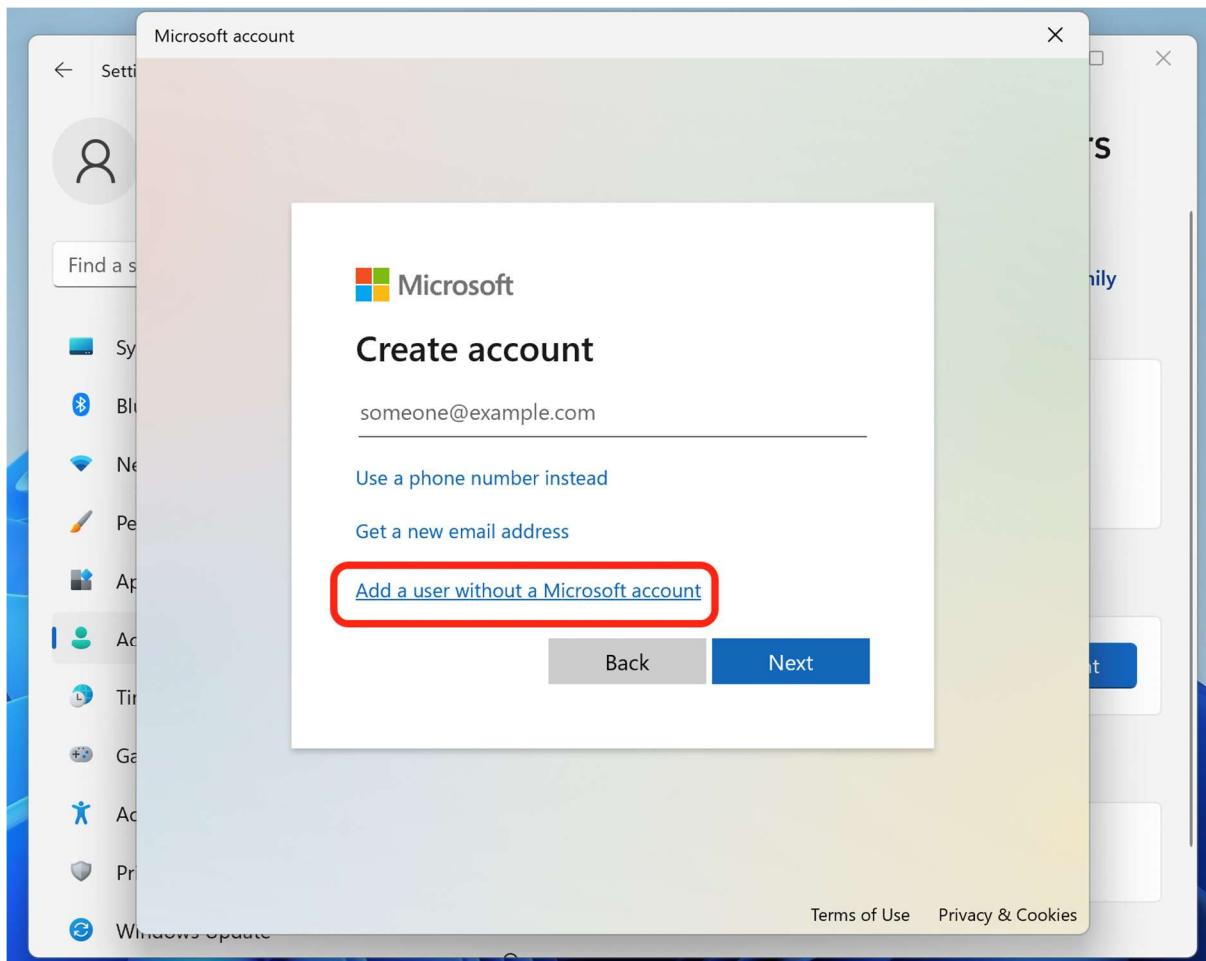
Rys. 9.18 Tworzenie konta użytkownika w Windows 11

W zakładce „Family & other users” należy wybrać przycisk „Add account”, wskazany wyżej czerwoną sztrelką.

Otwarte zostanie kolejne okno, w którym trzeba wybrać hiperłącze pod formularzem, następnie wskazać potrzebę utworzenia konta lokalnego.



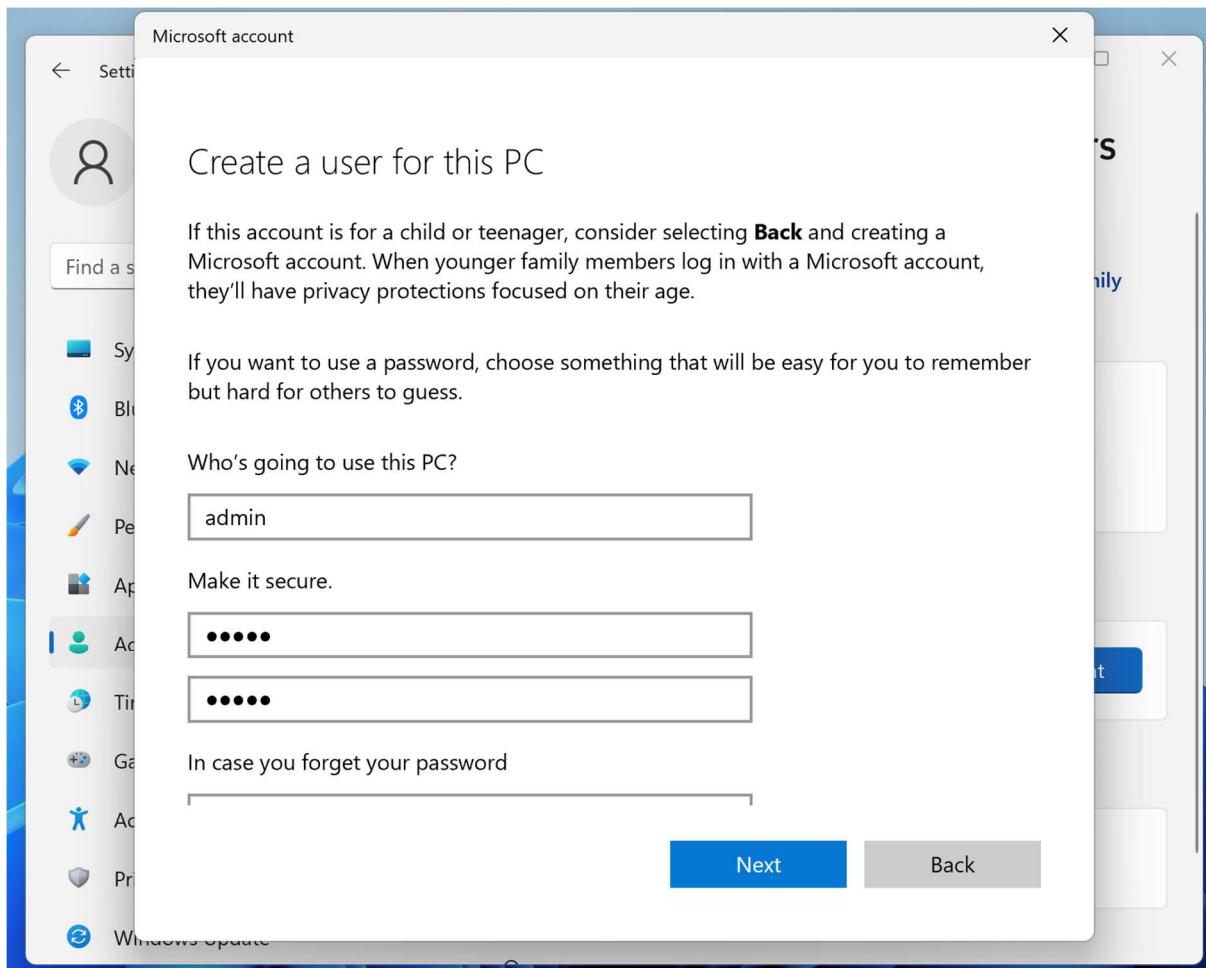
Rys. 9.19 Wskazanie na tworzenie konta lokalnego, krok 1



Rys. 9.20 Wskazanie na tworzenie konta lokalnego, krok 2

- Konfiguracja nowego konta użytkownika

W aktualnym oknie należy podać login i hasło dla tworzonego konta użytkownika. Ze względu na wymaganie [ŚD02] dane te muszą być takie same jak dla kont wykorzystywanych na innych maszynach objętych systemem SMMC.

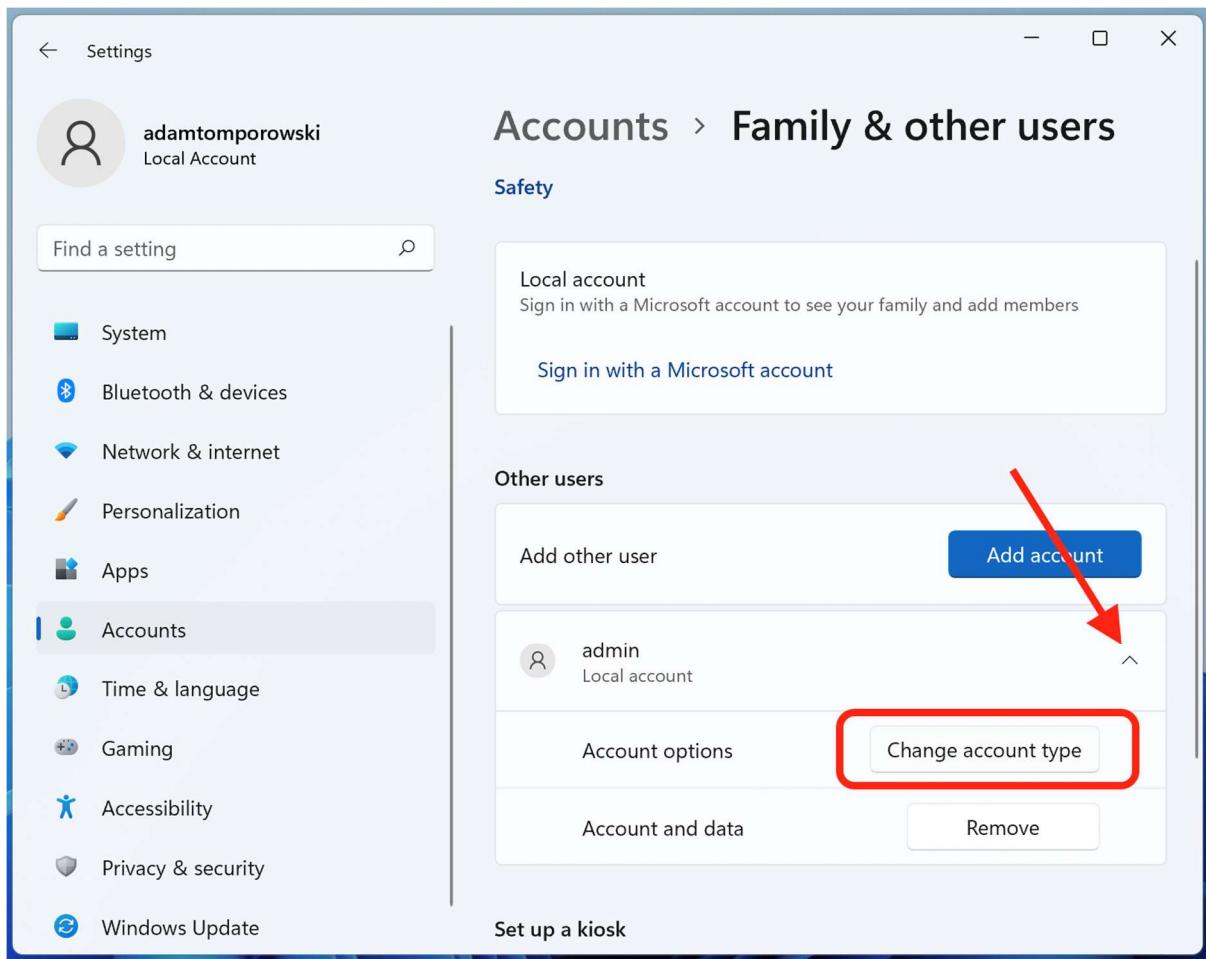


Rys. 9.21 Konfigurowanie nowego konta użytkownika

Po upewnieniu się co do poprawności loginu i hasła zapisujemy konfigurację klikając „Next”.

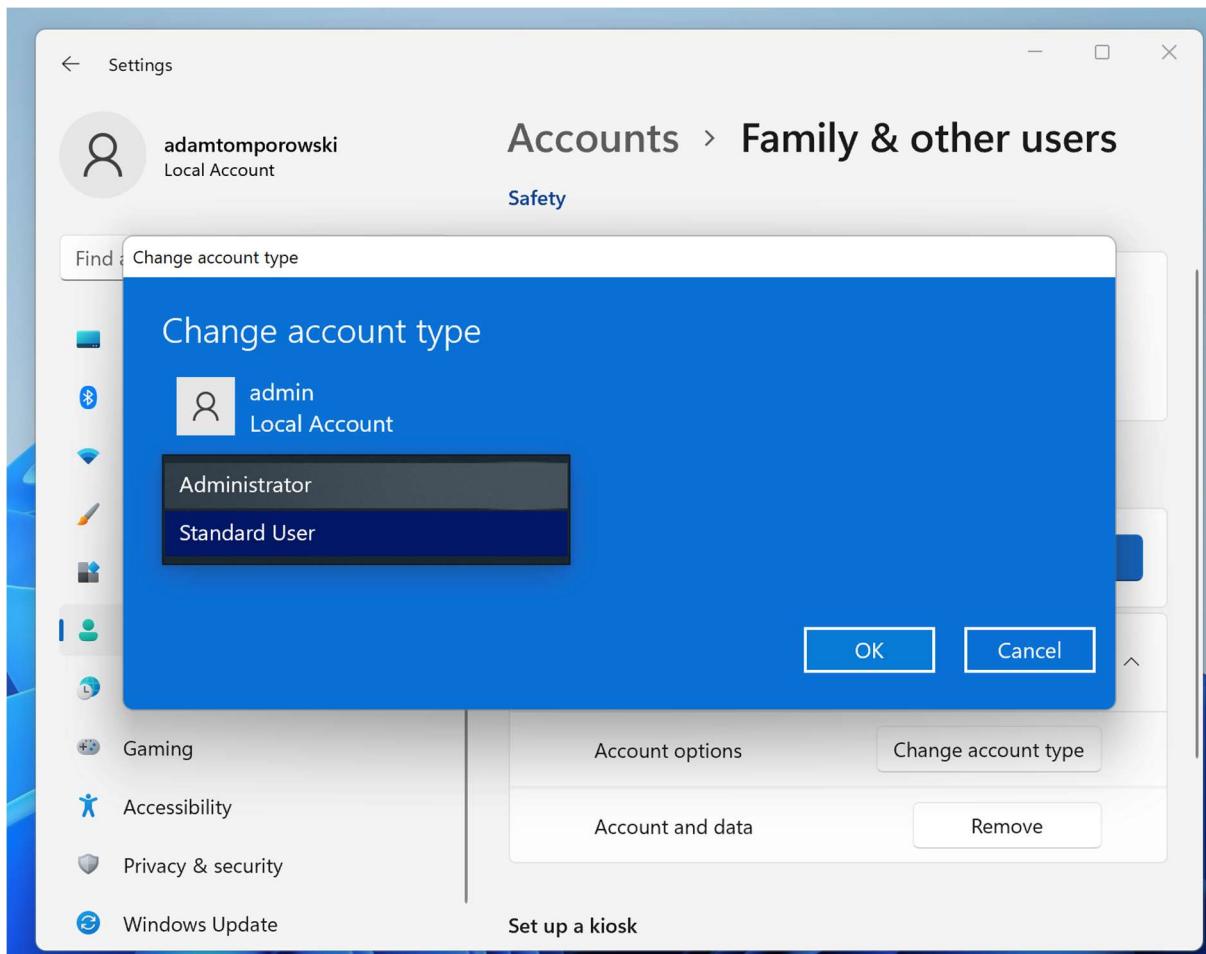
- Podniesienie uprawnień utworzonego konta

Na tym etapie nowe konto jest już utworzone, należy teraz zmienić jego typ z użytkownika standardowego na administratora.



Rys. 9.22 Podniesienie uprawnień konta użytkownika

W tym celu należy rozwinąć menu kontekstowe przy nowym użytkowniku (czerwona strzałka), a następnie wybrać pozycje „Change account type”.



Rys. 9.23 Wybór rodzaju konta użytkownika

Z nowego okna, za pomocą listy rozwijanej, wybieramy „Administrator” i potwierdzamy nasz wybór klikając „OK”.

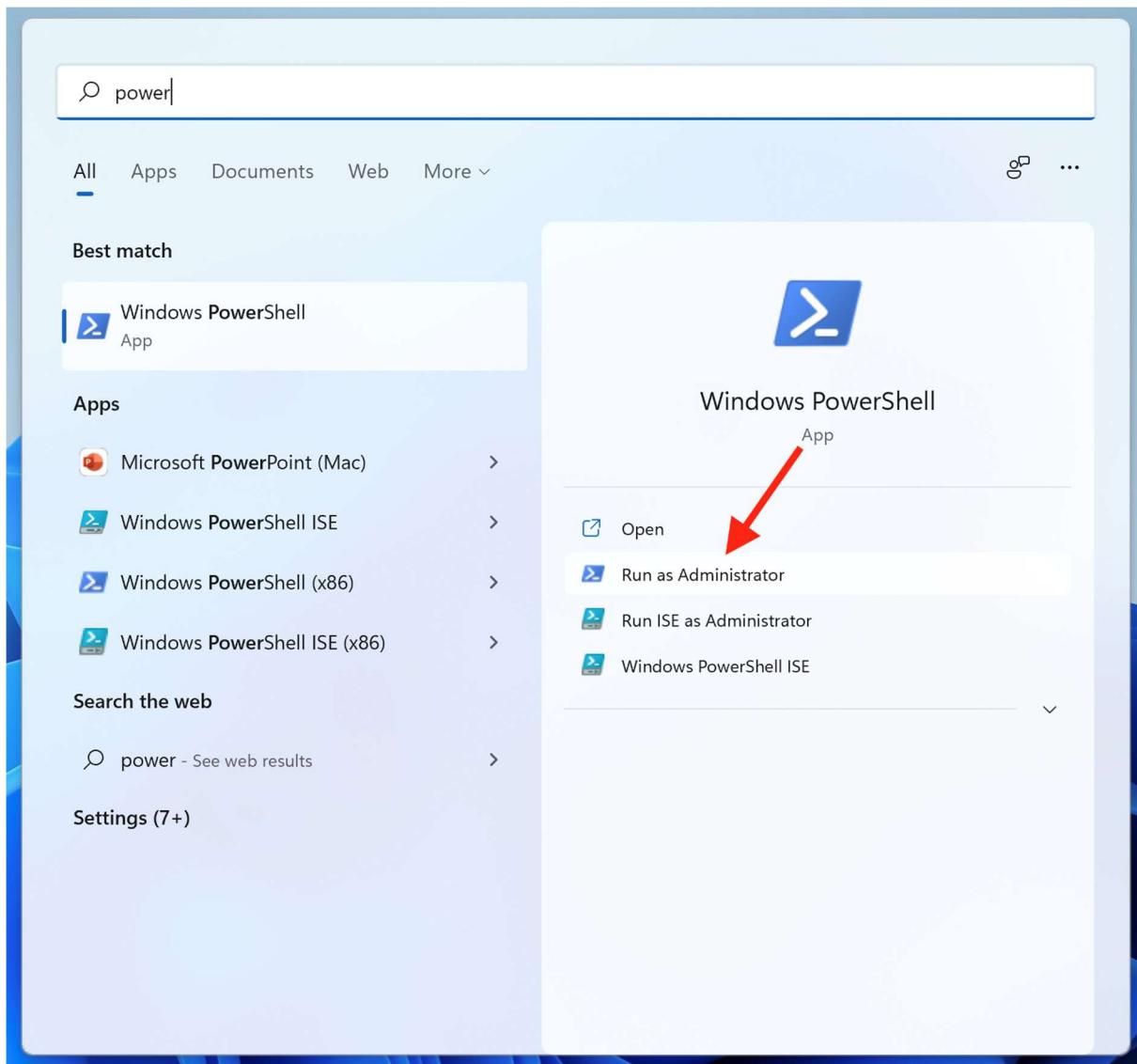
9.1.3.2. Przygotowanie systemu Windows pod konfigurację za pomocą Ansible

Ansible do komunikacji i zarządzania systemami Windows używa usługi WinRM (Ansible project contributors, 2022), jednak by zapewnić poprawną wymianę informacji między urządzeniami, należy spełnić wymagania [ŚD06], [ŚD07] oraz [ŚD08].

Korzystając z dokumentacji udostępnionej przez zespół Red Hat (Ansible, 2021), napisaliśmy skrypt PowerShell, który automatyzuje te czynności. Skrypt został umieszczony w repozytorium aplikacji SMMC (SMMC/mainwebsite/monitoring/windows/enable_ansible.ps1).

W celu uruchomienia skryptu należy wykonać kolejne kroki:

- Uruchomienie konsoli PowerShell jako administrator

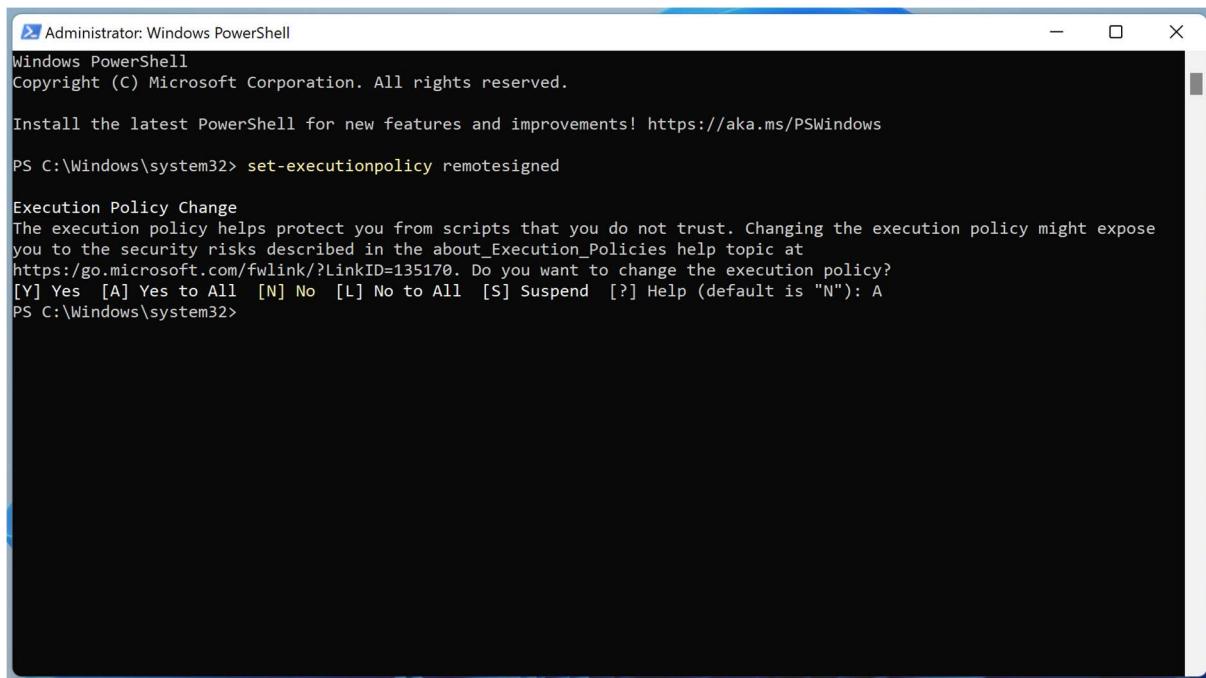


Rys. 9.24 Uruchomienie konsoli PowerShell jako administrator

Za pomocą wyszukiwania znajdź PowerShell, następnie kliknij „Run as Administrator”.

- Zaakceptuj wykonywanie skryptów „unsigned”

W tym calezu należy wykonać komendę „set-executionpolicy remotesigned”, a następnie wyrazić zgodę na wszystkie rodzaje skryptów.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

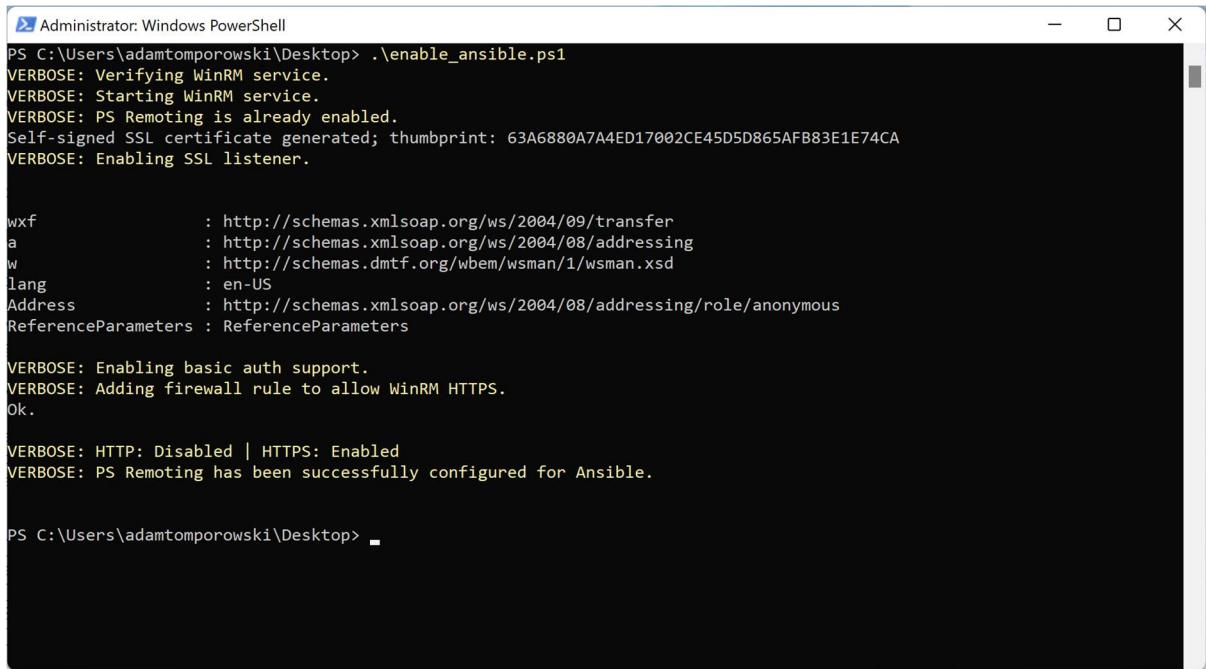
PS C:\Windows\system32> set-executionpolicy remotesigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
PS C:\Windows\system32>
```

Rys. 9.25 Zmiana polityki wykonywania skryptów PowerShell

- Wywołanie skryptu *enable_ansible.ps1*

Teraz wystarczy wskazać konsoli lokalizację pliku .ps1, a skrypt wykona całą pracę.



```
Administrator: Windows PowerShell
PS C:\Users\adamtomporowski\Desktop> .\enable_ansible.ps1
VERBOSE: Verifying WinRM service.
VERBOSE: Starting WinRM service.
VERBOSE: PS Remoting is already enabled.
Self-signed SSL certificate generated; thumbprint: 63A6880A7A4ED17002CE45D5D865AFB83E1E74CA
VERBOSE: Enabling SSL listener.

wxfs
a
w
lang
Address
ReferenceParameters : ReferenceParameters

VERBOSE: Enabling basic auth support.
VERBOSE: Adding firewall rule to allow WinRM HTTPS.
Ok.

VERBOSE: HTTP: Disabled | HTTPS: Enabled
VERBOSE: PS Remoting has been successfully configured for Ansible.

PS C:\Users\adamtomporowski\Desktop>
```

Rys. 9.26 Wywołanie skryptu *enable_ansible.ps1*

9.2. Home page

Pierwszym ekranem strony internetowej jest podstrona „Home”, na której znajduje się ogólne przedstawienie możliwości systemu SMMC oraz przycisk do pobrania instrukcji uruchomienia i obsługi aplikacji.



HOME LOGIN SIGN UP

SERVER MONITORING AND MANAGEMENT SYSTEM

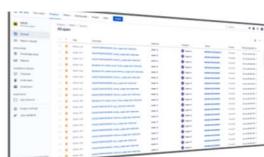


CHECK CURRENT
VALUES

AUTOMATICALLY
SCAN YOUR
NETWORK



Zdj. 9.27 Strona Home, cz. 1



AUTOMATIC TICKETS
TO JIRA

YOU WANNA KNOW
SMMC BETTER?

DOWNLOAD MANUAL

CONTACT

Zdj. 9.28 Strona Home, cz. 2

9.3. Register and login

Aby uzyskać dostęp do kluczowych elementów aplikacji SMMC, konieczna jest rejestracja, w tym celu należy przejść do odpowiedniej podstrony, klikając „SIGN UP” w prawym górnym rogu ekranu.

Formularz rejestracyjny zbiera i waliduje podstawowe informacje o użytkowniku aplikacji.

SIGN UP

Adam Tomporowski
s16740@pjwstk.edu.pl
PJATK

123 456 789

Submit

Zdj. 9.29 Formularz rejestracyjny SMMC

Poprawnie wypełniony formularz należy potwierdzić klikając „Submit”, a jeżeli nie zostały wykryte żadne błędy, użytkownik zostanie przeniesiony na stronę główną, gdzie zostanie wyświetlony stosowny komunikat.



Rys. 9.30 Komunikat potwierdzający poprawną rejestrację

Podstrona „LOGIN” pozwala oczywiście na zalogowanie się podając te same dane co przy rejestracji.

The screenshot shows a login form titled "LOGIN" in large white letters at the top. Below the title are two input fields: the first contains the email "s16740@pjwstk.edu.pl" and the second contains a series of asterisks representing a password. At the bottom of the form is a large dark grey button labeled "Login".

Rys. 9.31 Formularz logowania

9.4. Configuration

The screenshot shows the "CONFIGURATION" section of the SMMC application. At the top, there are navigation links: HOME, ASSETS, REPORTS, CONFIGURATION, and LOGOUT (Adam Tomporowski). The main area is divided into two sections: "Server Configuration" on the left and "Alert sender Configuration" on the right. Both sections contain several input fields and buttons. The "Server Configuration" section includes fields for Network IP, Login for the service account, and Password for the service account, along with "Submit" and "Reset server config" buttons. The "Alert sender Configuration" section includes fields for E-mail which sends notifications, Password for e-mail account, SMTP address, Port number (with a dropdown menu), Monitoring server IP, and "Submit" and "Reset e-mail config" buttons.

Rys. 9.32 Formularze konfiguracyjne aplikacji SMMC

Podstrona „CONFIGURATION” zbiera kluczowe informacje wymagane do poprawnego działania aplikacji SMMC, wszystkie pola są wymagane, jednak możliwa jest późniejsza zmiana wprowadzonych wartości.

Your current server config is:	
Network address	192.168.0.1
Admin login	admin
Admin password	admin

Your current e-mail sender config is:	
E-mail Account	servermmcenter@gmail.com
E-mail Account Password	Password hidden
SMTP for E-mail Account	smtp.gmail.com
E-mail Account Port	587
Monitoring Server IP	192.168.0.1

Rys. 9.33 Uzupełniona konfiguracja dla przykładowej sieci

Formularze posiadają walidację wprowadzanych danych, co pomaga użytkownikowi uniknąć popełnienia błędów.

9.5. Assets

Podstrona „Assets” służy do zarządzania listą urządzeń objętych monitoringiem. Wykorzystuje konfigurację podaną w poprzednim kroku i wyszukuje wszystkie urządzenia w sieci lokalnej.

ASSETS

SCAN

We have discovered the below devices in your network:

ID	IP Addresses	Option
1	192.168.0.1	Delete
2	192.168.0.59	Delete
3	192.168.0.132	Delete
4	192.168.0.164	Delete
5	192.168.0.150	Delete

SAVE SERVERS FOR ANSIBLE

OPEN CONFIG FILE FOR ANSIBLE

DELETE CONFIG FILE

Monitoring server IP (e.g. 192.168.1.1)

ADD OTHER SERVER

Rys. 9.34 Wyniki skanowania sieci lokalnej

Istnieje możliwość ręcznego dodawania i usuwania adresów IP. Listę urządzeń należy potwierdzić klikając „SAVE SERVERS FOR ANSIBLE”.

9.6. Reports

Aby rozpocząć komunikację między serwerem monitorującym, a klientami należy uruchomić skrypt odpowiedzialny za odbieranie połączeń (*SMMC/mainwebsite/monitoring/server.py*).

```
[STARTING] server is starting...
[LISTENING] Server is listening on 192.168.0.59
```

Rys. 9.35 Server w trybie nasłuchiwanego

Następnie wywołać komendy Ansible:

```
[adamtomprowski@MacBook-Pro-Adam monitoring % ansible-playbook -u admin smmc-linux-deploy.yml --ask-become-pass
[BECOME password:

PLAY [Setup environment for SMMC monitoring] *****

TASK [Gathering Facts] *****
fatal: [192.168.0.1]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.0.1 port 22: Connection refused", "unreachable": true}
fatal: [192.168.0.59]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.0.59 port 22: Connection refused", "unreachable": true}
[WARNING]: Platform linux on host 192.168.0.178 is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.12/reference_appendices/interpreter_discovery.html for more information.
ok: [192.168.0.178]
[WARNING]: Platform linux on host 192.168.0.164 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.12/reference_appendices/interpreter_discovery.html for more information.
ok: [192.168.0.164]
fatal: [192.168.0.150]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.0.150 port 22: Connection refused", "unreachable": true}

TASK [Install systemd] *****
ok: [192.168.0.164]
ok: [192.168.0.178]

TASK [Create working directory] *****
ok: [192.168.0.164]
ok: [192.168.0.178]

TASK [Copy monitoring script] *****
ok: [192.168.0.178]
ok: [192.168.0.164]

TASK [Copy service file] *****
ok: [192.168.0.164]
ok: [192.168.0.178]

TASK [Copy config file] *****
ok: [192.168.0.178]
ok: [192.168.0.164]

TASK [Install python] *****
ok: [192.168.0.164]
ok: [192.168.0.178]

TASK [Install psutil] *****
changed: [192.168.0.164]
changed: [192.168.0.178]

TASK [Enable monitoring service] *****
changed: [192.168.0.178]
ok: [192.168.0.164]

TASK [Start monitoring service] *****
changed: [192.168.0.178]
ok: [192.168.0.164]

PLAY RECAP *****
192.168.0.1      : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.150    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.164    : ok=10   changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.178    : ok=10   changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.59     : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

adamtomprowski@MacBook-Pro-Adam monitoring % ]
```

Rys. 9.36 Wywołanie Ansible dla klientów Linux

```

fatal: [192.168.0.150]: UNREACHABLE! => {"changed": false, "msg": "ssl: HTTPSConnectionPool(host='192.168.0.150', port=5986): Max retries exceeded with url: /wsman (Caused by NewConnectionError('<urllib3.connection.HTTPSConnection object at 0x7faf98035d90>: Failed to establish a new connection: [Errno 61] Connection refused'))", "unreachable": true}
fatal: [192.168.0.164]: FAILED! => {"msg": "the connection attempt timed out"}
fatal: [192.168.0.132]: UNREACHABLE! => {"changed": false, "msg": "ssl: HTTPSConnectionPool(host='192.168.0.132', port=5986): Max retries exceeded with url: /wsman (Caused by ConnectTimeoutError(<urllib3.connection.HTTPSConnection object at 0x7fafd860f1f0>, 'Connection to 192.168.0.132 timed out. (connect timeout=30)'))", "unreachable": true}
[WARNING]: Error when collecting bios facts: New-Object : Exception calling ".ctor" with "0" argument(s): "String was not recognized as a valid DateTime." At line:2 char:21 + ... $bios = New-Object -TypeName Ansible.Windows.Setup.SMBIOSInfo + CategoryInfo : InvalidOperation: () [New-Object], MethodInvocationException + FullyQualifiedErrorCode : ConstructorInvokedThrowException,Microsoft.PowerShell.Commands.NewObjectCommand at <ScriptBlock>, <No file>; line 2
[WARNING]: Error when collecting platform facts: New-Object : Exception calling ".ctor" with "0" argument(s): "String was not recognized as a valid DateTime." At line:2 char:21 + ... $bios = New-Object -TypeName Ansible.Windows.Setup.SMBIOSInfo + CategoryInfo : InvalidOperation: () [New-Object], MethodInvocationException + FullyQualifiedErrorCode : ConstructorInvokedThrowException,Microsoft.PowerShell.Commands.NewObjectCommand at <ScriptBlock>, <No file>; line 2
[WARNING]: Error when collecting processor facts: New-Object : Exception calling ".ctor" with "0" argument(s): "String was not recognized as a valid DateTime." At line:2 char:21 + ... $bios = New-Object -TypeName Ansible.Windows.Setup.SMBIOSInfo + CategoryInfo : InvalidOperation: () [New-Object], MethodInvocationException + FullyQualifiedErrorCode : ConstructorInvokedThrowException,Microsoft.PowerShell.Commands.NewObjectCommand at <ScriptBlock>, <No file>; line 2
[WARNING]: Error when collecting processor facts: Cannot index into a null array. At line:22 char:13 + ... $ansibleFacts.ansible_processor_cores = $bios.ProcessorIn ... + CategoryInfo : InvalidOperation: () [], RuntimeException + FullyQualifiedErrorCode : NullArray at <ScriptBlock>, <No file>; line 2
[WARNING]: Error when collecting processor facts: Cannot index into a null array. At line:24 char:13 + ... $ansibleFacts.ansible_processor_threads_per_core = $bios. ... + CategoryInfo : InvalidOperation: () [], RuntimeException + FullyQualifiedErrorCode : NullArray at <ScriptBlock>, <No file>; line 2
[WARNING]: Error when collecting virtual facts: New-Object : Exception calling ".ctor" with "0" argument(s): "String was not recognized as a valid DateTime." At line:2 char:21 + ... $bios = New-Object -TypeName Ansible.Windows.Setup.SMBIOSInfo + CategoryInfo : InvalidOperation: () [New-Object], MethodInvocationException + FullyQualifiedErrorCode : ConstructorInvokedThrowException,Microsoft.PowerShell.Commands.NewObjectCommand at <ScriptBlock>, <No file>; line 2
ok: [192.168.0.229]

TASK [Create working directory] *****
[WARNING]: ERROR DURING WINRM SEND INPUT - attempting to recover: ReadTimeout HTTPSConnectionPool(host='192.168.0.229', port=5986): Read timed out. (read timeout=30)
changed: [192.168.0.229]

TASK [Copy monitoring script] *****
changed: [192.168.0.229]

TASK [Copy config file] *****
changed: [192.168.0.229]

TASK [Install NSSM (service manager)] *****
[WARNING]: Chocolatey was missing from this system, so it was installed during this task run.
changed: [192.168.0.229]

TASK [Install python] *****
changed: [192.168.0.229]

TASK [Install modules] *****
changed: [192.168.0.229]

TASK [Install service] *****
changed: [192.168.0.229]

TASK [Start monitoring service] *****
changed: [192.168.0.229]

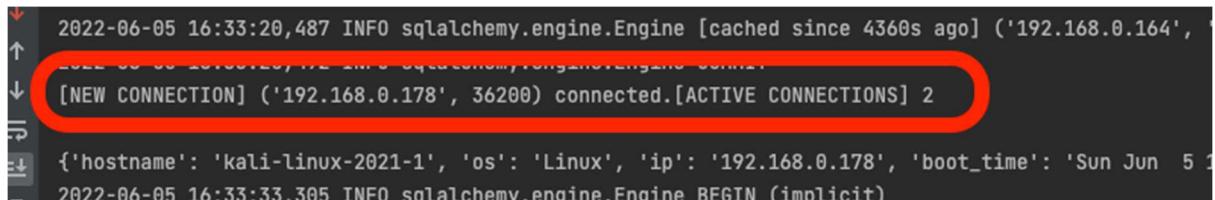
PLAY RECAP *****
192.168.0.1      : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.132    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.150    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.164    : ok=0    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.0.229    : ok=9    changed=8    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.0.59     : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

adamtomporowski@MacBook-Pro-Adam monitoring % 

```

Rys. 9.37 Wywołanie Ansible dla klientów Windows

Ansible samodzielnie przefiltruje urządzenia niechciane, tj. routery, telefony komórkowe i inne, które nie mają być objęte monitoringiem. W konsoli logów *server.py* pojawiły się odpowiednie komunikaty informujące o podłączaniu kolejnych urządzeń.



```
2022-06-05 16:33:20,487 INFO sqlalchemy.engine.Engine [cached since 4360s ago] ('192.168.0.164',  
2022-06-05 16:33:20,487 INFO sqlalchemy.engine.Engine [cached since 4360s ago] ('192.168.0.164',  
[NEW CONNECTION] ('192.168.0.178', 36200) connected. [ACTIVE CONNECTIONS] 2  
{'hostname': 'kali-linux-2021-1', 'os': 'Linux', 'ip': '192.168.0.178', 'boot_time': 'Sun Jun 5 16:33:33.305 2022-06-05 16:33:33.305 INFO sqlalchemy.engine.Engine BEGIN (implicit)
```

Rys. 9.38 Logi skryptu *server.py*

Na podstronie „REPORTS” są już wyświetlane skonfigurowane urządzenia.

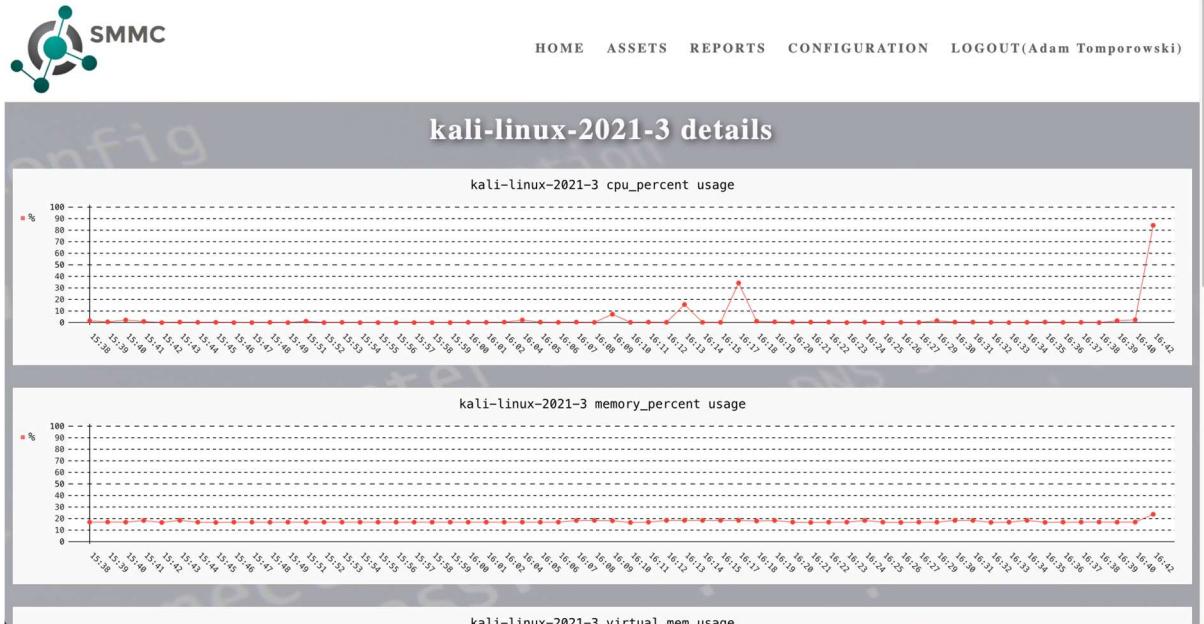


Hostname	Server IP	Boot time	CPU percent	Memory percent	Virtual memory	Avg Load	Timestamp
ADAMTOMPOROB51E	192.168.0.229	Thu Jun 2 13:55:46 2022	100.0	75.8	75.8	1.3	Sun Jun 5 16:56:22 2022
kali-linux-2021-3	192.168.0.164	Sun Jun 5 13:31:14 2022	84.2	23.7	23.6	0.0	Sun Jun 5 16:42:01 2022
kali-linux-2021-1	192.168.0.178	Sun Jun 5 16:21:45 2022	5.1	41.5	41.5	0.1	Sun Jun 5 16:38:58 2022

Rys. 9.39 Lista klientów objętych monitoringiem

W tabeli wyświetlane są najnowsze wpisy z bazy danych dla danego urządzenia, dodatkowo tabela objęta jest formatowaniem warunkowym co skutkuje zmianą kolorów w zależności od odczytanych wartości. Cała podstrona „REPORTS” automatycznie odświeża się w interwałach co 60 sekund, aby zawsze pokazywała najnowsze wpisy.

Kliknięcie w dowolną nazwę urządzenia otworzy podstronę z dokładnymi wykresami zużycia zasobów klienta – są tą odczyty zużycia CPU, pamięci RAM, pamięci wirtualnej oraz średniego obciążenia.



Rys. 9.40 Podstrona jednego z klientów objętych SMMC

9.6.1. Skrypty monitorujące

Skrypty monitorujące są uruchamiane na urządzeniach klienckich w postaci usług. Zapewnia to większą niezawodność i intuicyjność – usługa samodzielnie próbuje nawiązać nowe połączenie w przypadku wystąpienia awarii, restartu urządzenia i innych. Nie jest wymagane ponowne skanowanie sieci ani uruchamianie Ansible.

Services (Local)					
	Name	Description	Status	Startup Type	Log On As
Stop the service	Secondary Logon	Enables star...	Manual	Local Syste...	
Restart the service	Secure Socket Tunneling Protocol Service	Provides su...	Running	Manual	Local Service
	Security Accounts Manager	The startup ...	Running	Automatic	Local Syste...
	Security Center	The WSCSV...	Running	Automatic (...)	Local Service
	Sensor Data Service	Delivers dat...	Manual (Trig...	Local Syste...	
	Sensor Monitoring Service	Monitors va...	Manual (Trig...	Local Service	
	Sensor Service	A service fo...	Manual (Trig...	Local Syste...	
	Server	Supports fil...	Running	Automatic (T...	Local Syste...
	Shared PC Account Manager	Manages pr...	Disabled	Local Syste...	
	Shell Hardware Detection	Provides no...	Running	Automatic	Local Syste...
	Smart Card	Manages ac...	Manual (Trig...	Local Service	
	Smart Card Device Enumeration Service	Creates soft...	Manual (Trig...	Local Syste...	
	Smart Card Removal Policy	Allows the s...	Manual	Local Syste...	
	SMMC_client	Receives tra...	Running	Automatic	Local Syste...
	SNMP Trap	Enables the ...	Manual	Local Service	
	Software Protection	This service ...	Running	Automatic (...)	Network S...
	Spatial Data Service	Verifies pote...	Manual	Local Service	
	Spot Verifier	Discovers n...	Manual	Manual (Trig...	Local Syste...
	SSDP Discovery	Provides re...	Running	Automatic	Local Syste...
	State Repository Service	Steam Clien...	Manual	Local Syste...	
	Steam Client Service	Launches a...	Manual	Local Syste...	
	Still Image Acquisition Events				

Rys. 9.41 Usługa SMMC w systemie Windows

W przypadku systemu Windows wykorzystany został menedżer serwisów „NSSM”. Instalacja NSSM odbywa się za pomocą wbudowanego w Ansible modułu „win_chocolatey”

```
(parallels@kali-linux-2021-3:[~]
$ systemctl status SMMC_client
● SMMC_client.service - SMMC monitoring script for client nodes
  Loaded: loaded (/etc/systemd/system/SMMC_client.service; enabled; vendor preset: disabled)
  Active: active (running) since Sun 2022-06-05 17:57:07 CEST; 1min 19s ago
    Main PID: 1822 (python3)
      Tasks: 1 (limit: 2211)
     Memory: 9.0M
        CPU: 50ms
       CGroup: /system.slice/SMMC_client.service
               └─1822 /usr/bin/python3 /SMMC/client.py

Jun 05 17:57:07 kali-linux-2021-3 systemd[1]: Started SMMC monitoring script for client nodes.
```

Rys. 9.42 Usługa SMMC w systemie Linux

W przypadku systemu Linux wykorzystany został menedżer serwisów „systemctl”. Większość dystrybucji Linuxa jest domyślnie w niego wyposażona, jednak dla zapewnienia maksymalnej uniwersalności, jednym z zaprogramowanych zadań Ansible jest pobranie tego menedżera.

Instalacja i pierwsze uruchomienie usług również jest obsługiwane przez playbooki Ansible.

9.6.2. Reporting issues

Poza dashboardem na podstronie „REPORTS”, oraz wykresami z podstrony „DETAILS”, aplikacja SMMC umożliwia również zgłaszanie awarii drogą mailową.

<input type="checkbox"/>	<input checked="" type="checkbox"/>	ja
		ADAMTOMPORO4AE9 cpu_usage alert detected. - Alert details: ADAMTOMPORO4AE9 cpu_usage 100.0

<input type="checkbox"/>	<input checked="" type="checkbox"/>	ja
		ADAMTOMPOROB51E cpu_usage alert detected. - Alert details: ADAMTOMPOROB51E cpu_usage 100.0

Rys. 9.43 Ostrzeżenia wykryte przez skrypty monitorujące

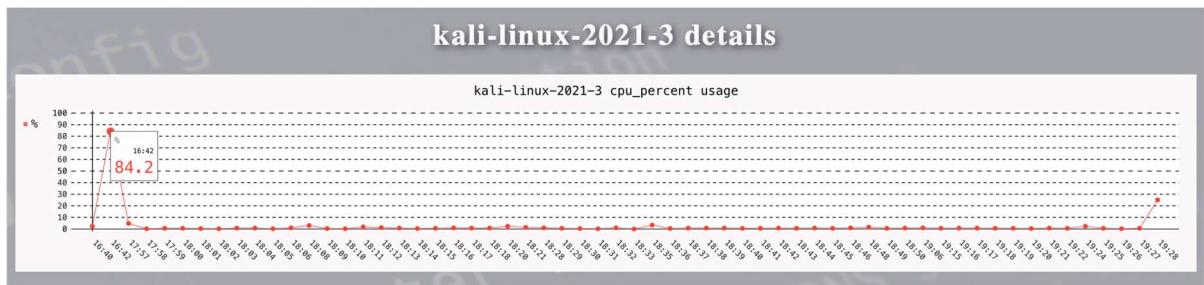
Tak jak w przypadku formularza kontaktowego, korzystając z API usług ticketowania, istnieje możliwość zautomatyzowania tworzenia zgłoszeń np. w systemie Jira.

Poziomy zużycia parametrów zostały podzielone na dwie kategorie; „WARNING” oraz „CRITICAL”. W przypadku zużycia procesora, pamięci RAM i pamięci wirtualnej wynoszą odpowiednio 90 i 95 procent. W przypadku parametru „avg_load” jest to 10 i 15 procent zużycia.

Odczyty zużycia parametrów następują w interwałach co 60 sekund, a alarmy wysyłane są co 10 odczytów spełniających warunki WARNING lub CRITICAL.

9.6.3. Details

Podstrona „DETAILS” jest automatycznie tworzona dla każdego urządzenia objętego monitoringiem. Znajdują się na niej wykresy zużycia parametrów aż do ostatnich 60 wpisów do bazy danych. W rzeczywistości wpisy tworzone są co minutę, więc każdy wykres przedstawia zużycie z ostatniej godziny. Testy nie wykazały utraty danych większej niż 2%.



Rys. 9.44 Wykres zużycia procesora na jednej z maszyn wirtualnych

Wykresy są interaktywne i pozwalają sprawdzić dokładne odczyty parametrów.

9.7. Support contact

W stopce strony SMMC znajduje się formularz kontaktowy, który ma służyć jako punkt kontaktu między użytkownikami, a zespołem SMMC.

CONTACT

Any questions? Let us know

I agree to the processing of my personal data in order to handle the question.

Send

Rys. 9.45 Uzupełniony formularz kontaktowy

Po uzupełnieniu formularza wystarczy potwierdzić żądanie klikając „Send”, powinien zostać wyświetlony stosowny komunikat:



Rys. 9.46 Potwierdzenie wysłania wiadomości

Adres odbierający wiadomości z tego formularza, został skonfigurowany w taki sposób, aby automatycznie tworzyć nowe tickety w serwisie Jira.

[Back](#) [SMMC-161](#)

Question From Website from: Adam Tomporowski

[Link issue](#) [...](#)

 Adam G raised this request via Jira

Description

Message:

Testowa wiadomość do wsparcia SMMC.

Adam Tomporowski

s16740@pjwstk.edu.pl

Similar requests

...

Activity

Show: All [Comments](#) History Work log

Newest first ↗



[Add internal note](#) / [Reply to customer](#)



Pro tip: press **M** to comment

Rys. 9.47 Ticket wygenerowany automatycznie na podstawie wiadomości email