Instrukcja instalacji i konfiguracji

WSL 2, Ubuntu, Ansible i network-probe-workshop

Ten dokument przeprowadzi Ci■ krok po kroku przez proces instalacji i konfiguracji ■rodowiska do pracy z Ansible oraz projektem network-probe-workshop. Ka■da sekcja zawiera szczegó■owe instrukcje wraz z komendami do wykonania.

1. Instalacja WSL 2 z Ubuntu

WSL 2 (Windows Subsystem for Linux) pozwala na uruchamianie pe■noprawnego ■rodowiska Linux bezpo■rednio na Windows.

Krok 1.1: Sprawdzenie wersji Windows

Upewnij si■, ■e masz Windows 10 w wersji 2004 lub nowszej (Build 19041+) albo Windows 11. Otwórz PowerShell jako Administrator i wykonaj:

```
wsl --install
```

Ta komenda automatycznie:

W■■czy funkcje WSL i Virtual Machine Platform

Pobierze i zainstaluje najnowsze j■dro Linux

Zainstaluje Ubuntu jako domy■ln■ dystrybucj■

Krok 1.2: Restart systemu

Po zako■czeniu instalacji, uruchom ponownie komputer.

Krok 1.3: Konfiguracja Ubuntu

Po restarcie, Ubuntu uruchomi si■ automatycznie. Zostaniesz poproszony o utworzenie u■ytkownika i has■a. Zapisz je w bezpiecznym miejscu!

Krok 1.4: Aktualizacja systemu

Po pierwszym uruchomieniu Ubuntu, zaktualizuj system:

```
sudo apt update && sudo apt upgrade -y
```

2. Utworzenie klucza SSH i konfiguracja

Klucz SSH pozwoli Ci na bezpieczne po■■czenie z serwerem bez konieczno■ci wpisywania has■a.

Krok 2.1: Generowanie klucza SSH

Wygeneruj nowy klucz SSH (zalecany typ: ed25519):

```
ssh-keygen -t ed25519 -C "twoj_email@example.com"
```

Naci■nij Enter, aby zaakceptowa■ domy■ln■ lokalizacj■ (~/.ssh/id_ed25519). Mo■esz równie■ ustawi■ has■o dla dodatkowego bezpiecze■stwa.

Krok 2.2: Instalacja ssh-copy-id (je■li potrzebna)

Sprawd■, czy masz zainstalowane narz■dzie ssh-copy-id:

```
which ssh-copy-id
```

Je■li nie jest zainstalowane:

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

Krok 2.3: Kopiowanie klucza na serwer

Skopiuj swój klucz publiczny na zdalny serwer (zast■p USER i SERVER_IP odpowiednimi warto■ciami):

```
ssh-copy-id USER@SERVER_IP
```

Przyk■ad:

```
ssh-copy-id admin@192.168.1.100
```

Zostaniesz poproszony o has■o do serwera. Po pomy■lnym skopiowaniu klucza, b■dziesz móg■ ■■czy■ si■ bez has■a.

Krok 2.4: Test po■■czenia SSH

Sprawd■, czy mo■esz po■■czy■ si■ z serwerem bez has■a:

```
ssh USER@SERVER_IP
```

3. Sprawdzenie i instalacja Git

Krok 3.1: Sprawdzenie czy Git jest zainstalowany

Sprawd■, czy Git jest ju■ zainstalowany:

```
git --version
```

Je■li zobaczysz numer wersji (np. git version 2.34.1), Git jest ju■ zainstalowany. Przejd■ do nast■pnego kroku.

Krok 3.2: Instalacja Git (je■li potrzebna)

Je■li Git nie jest zainstalowany, zainstaluj go:

```
sudo apt install git -y
```

Sprawd■ ponownie wersj■, aby potwierdzi■ instalacj■:

```
git --version
```

Krok 3.3: Konfiguracja Git (opcjonalnie)

Skonfiguruj swoje dane u■ytkownika Git:

```
git config --global user.name "Twoje Imi■"
```

```
git config --global user.email "twoj_email@example.com"
```

4. Pobranie repozytorium z GitHub

Krok 4.1: Klonowanie repozytorium

Sklonuj repozytorium network-probe-workshop z GitHub przez HTTPS:

```
git clone https://github.com/slavpiet/network-probe-workshop.git
```

Krok 4.2: Przej■cie do katalogu projektu

Wejd■ do sklonowanego katalogu:

```
cd network-probe-workshop
```

Sprawd■ zawarto■■ katalogu:

```
ls -la
```

5. Instalacja Ansible

Krok 5.1: Instalacja wymaganych pakietów

Zainstaluj Python pip (mened■er pakietów Python):

```
sudo apt install python3-pip -y
```

Krok 5.2: Instalacja Ansible

Zainstaluj Ansible za pomoc■ pip:

```
pip3 install ansible
```

Mo■esz równie■ zainstalowa■ Ansible z repozytorium Ubuntu:

```
sudo apt install ansible -y
```

Krok 5.3: Weryfikacja instalacji

Sprawd■ wersj■ Ansible:

```
ansible --version
```

6. Instalacja zale■no■ci Ansible

Krok 6.1: Sprawdzenie pliku requirements.txt

Sprawd■, czy w repozytorium znajduje si■ plik requirements.txt (dla Ansible nazywa si■ zwykle requirements.yml):

```
ls requirements.*
```

Krok 6.2: Instalacja zale■no■ci (collections i roles)

Je■li plik nazywa si■ requirements.yml:

```
ansible-galaxy install -r requirements.yml
```

Lub je■li plik nazywa si■ requirements.txt (dla modu■ów Python):

```
pip3 install -r requirements.txt
```

Uwaga: Je■li repo zawiera oba pliki, zainstaluj zale■no■ci z obu:

```
ansible-galaxy install -r requirements.yml
pip3 install -r requirements.txt
```

7. Modyfikacja inventory i group_vars

Teraz musisz dostosowa■ konfiguracj■ Ansible do swoich potrzeb.

Krok 7.1: Edycja pliku inventory

Plik inventory zawiera list■ serwerów, którymi b■dziesz zarz■dza■. Znajd■ plik inventory (mo■e nazywa■ si■ inventory, hosts, lub inventory.ini):

```
ls inventory* hosts*
```

Edytuj plik za pomoc■ nano lub vim:

```
nano inventory
```

Przyk■adowa struktura pliku inventory:

```
[servers]
server1 ansible_host=192.168.1.100 ansible_user=admin
server2 ansible_host=192.168.1.101 ansible_user=admin
```

Zast■p warto■ci IP i u■ytkownika swoimi danymi. Zapisz plik (Ctrl+O w nano, Enter, potem Ctrl+X).

Krok 7.2: Edycja group_vars

Katalog group_vars zawiera zmienne specyficzne dla grup hostów. Sprawd■ jego zawarto■■:

```
ls group_vars/
```

Edytuj odpowiedni plik (np. all.yml lub nazw■ odpowiadaj■c■ Twojej grupie):

```
nano group_vars/all.yml
```

Dostosuj zmienne do swoich potrzeb (np. ■cie■ki, porty, parametry konfiguracyjne).

8. Test po■■czenia za pomoc■ Ansible

Krok 8.1: Ansible ping

Sprawd■, czy Ansible mo■e po■■czy■ si■ z Twoimi serwerami:

```
ansible all -m ping -i inventory
```

Je■li wszystko dzia■a poprawnie, zobaczysz komunikat SUCCESS dla ka■dego serwera:

```
server1 | SUCCESS =&gt; {
```

"ping": "pong"

}

Krok 8.2: Rozwi■zywanie problemów

Je■li napotkasz b■■dy:

Sprawd■, czy klucz SSH zosta■ poprawnie skopiowany

Zweryfikuj IP serwera i nazw■ u■ytkownika w inventory

Upewnij si■, ■e serwer jest w■■czony i dost■pny w sieci

Sprawd■, czy firewall nie blokuje po■■czenia SSH (port 22)

9. Wykonywanie playbooków krok po kroku

Krok 9.1: Lista dost■pnych playbooków

Sprawd■, jakie playbooki znajduj■ si■ w repozytorium:

```
ls *.yml
```

Krok 9.2: Wykonanie playbooka z flag■ --step

Flaga --step pozwala na wykonywanie playbooka krok po kroku z potwierdzeniem przed ka■d■ akcj■. Jest to przydatne podczas testowania i nauki.

Podstawowa sk■adnia:

```
ansible-playbook -i inventory nazwa_playbooka.yml --step
```

Przyk■ad:

```
ansible-playbook -i inventory site.yml --step
```

Krok 9.3: Interakcja podczas wykonywania --step

Po uruchomieniu z flag■ --step, Ansible b■dzie pyta■ przed ka■dym zadaniem:

Perform task: TASK: Nazwa zadania (N)o/(y)es/(c)ontinue:

Mo■esz wybra■:

y (yes) - wykonaj to zadanie i zapytaj o nast■pne

n (no) - pomi■ to zadanie i przejd■ do nast■pnego

c (continue) - wykonaj wszystkie pozosta■e zadania bez pytania

Krok 9.4: Dodatkowe przydatne flagi

Inne u■yteczne flagi podczas wykonywania playbooków:

```
ansible-playbook -i inventory site.yml --check
```

Tryb dry-run (--check) - pokazuje co zosta■oby zmienione bez faktycznego wykonywania zmian.

```
ansible-playbook -i inventory site.yml -v
```

Zwi■kszony poziom szczegó■owo■ci (-v, -vv, -vvv) - pokazuje wi■cej informacji o wykonywanych dzia■aniach.

```
ansible-playbook -i inventory site.yml --limit server1
```

Ograniczenie do konkretnych hostów (--limit) - wykonuje playbook tylko na wybranych serwerach.

Podsumowanie

Gratulacje! Pomy■lnie skonfigurowa■e■ kompletne ■rodowisko do pracy z Ansible. Teraz mo■esz:

Zarz■dza■ serwerami za pomoc■ Ansible

Wykonywa■ playbooki automatyzuj■ce konfiguracj■

Testowa■ zmiany krok po kroku u■ywaj■c flagi --step

Rozwija■ projekt network-probe-workshop

Przydatne linki:

Dokumentacja Ansible: https://docs.ansible.com/

Repozytorium projektu: https://github.com/slavpiet/network-probe-workshop

WSL Documentation: https://learn.microsoft.com/windows/wsl/

Powodzenia w automatyzacji!