Instrukcja instalacji i konfiguracji

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

Ten dokument przeprowadzi Cię krok po kroku przez proces instalacji i konfiguracji środowiska do pracy z Ansible oraz projektem network-probe-workshop. Każda sekcja zawiera szczegółowe instrukcje wraz z komendami do wykonania.

# 1. Instalacja WSL 2 z Ubuntu

WSL 2 (Windows Subsystem for Linux) pozwala na uruchamianie pełnoprawnego środowiska Linux bezpośrednio na Windows.

## Krok 1.1: Sprawdzenie wersji Windows

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

wsl --install

Ta komenda automatycznie:

* Włączy funkcje WSL i Virtual Machine Platform
* Pobierze i zainstaluje najnowsze jądro Linux
* Zainstaluje Ubuntu jako domyślną dystrybucję

## Krok 1.2: Restart systemu

Po zakończeniu instalacji, uruchom ponownie komputer.

## Krok 1.3: Konfiguracja Ubuntu

Po restarcie, Ubuntu uruchomi się automatycznie. Zostaniesz poproszony o utworzenie użytkownika i hasła. 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 połączenie z serwerem bez konieczności wpisywania hasła.

## Krok 2.1: Generowanie klucza SSH

Wygeneruj nowy klucz SSH (zalecany typ: ed25519):

ssh-keygen -t ed25519 -C "twoj\_email@example.com"

Naciśnij Enter, aby zaakceptować domyślną lokalizację (~/.ssh/id\_ed25519). Możesz również ustawić hasło dla dodatkowego bezpieczeństwa.

## Krok 2.2: Instalacja ssh-copy-id (jeśli potrzebna)

Sprawdź, czy masz zainstalowane narzędzie ssh-copy-id:

which ssh-copy-id

Jeśli nie jest zainstalowane:

sudo apt install openssh-client -y

## Krok 2.3: Kopiowanie klucza na serwer

Skopiuj swój klucz publiczny na zdalny serwer (zastąp USER i SERVER\_IP odpowiednimi wartościami):

ssh-copy-id USER@SERVER\_IP

Przykład:

ssh-copy-id admin@192.168.1.100

Zostaniesz poproszony o hasło do serwera. Po pomyślnym skopiowaniu klucza, będziesz mógł łączyć się bez hasła.

## Krok 2.4: Test połączenia SSH

Sprawdź, czy możesz połączyć się z serwerem bez hasła:

ssh USER@SERVER\_IP

# 3. Sprawdzenie i instalacja Git

## Krok 3.1: Sprawdzenie czy Git jest zainstalowany

Sprawdź, czy Git jest już zainstalowany:

git --version

Jeśli zobaczysz numer wersji (np. git version 2.34.1), Git jest już zainstalowany. Przejdź do następnego kroku.

## Krok 3.2: Instalacja Git (jeśli potrzebna)

Jeśli Git nie jest zainstalowany, zainstaluj go:

sudo apt install git -y

Sprawdź ponownie wersję, aby potwierdzić instalację:

git --version

## Krok 3.3: Konfiguracja Git (opcjonalnie)

Skonfiguruj swoje dane użytkownika Git:

git config --global user.name "Twoje Imię"

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: Przejście do katalogu projektu

Wejdź do sklonowanego katalogu:

cd network-probe-workshop

Sprawdź zawartość katalogu:

ls -la

# 5. Instalacja Ansible

## Krok 5.1: Instalacja wymaganych pakietów

Zainstaluj Python pip (menedżer pakietów Python):

sudo apt install python3-pip -y

## Krok 5.2: Instalacja Ansible

Zainstaluj Ansible za pomocą pip:

pip3 install ansible

Możesz również zainstalować Ansible z repozytorium Ubuntu:

sudo apt install ansible -y

## Krok 5.3: Weryfikacja instalacji

Sprawdź wersję Ansible:

ansible --version

# 6. Instalacja zależności Ansible

## Krok 6.1: Sprawdzenie pliku requirements.txt

Sprawdź, czy w repozytorium znajduje się plik requirements.txt (dla Ansible nazywa się zwykle requirements.yml):

ls requirements.\*

## Krok 6.2: Instalacja zależności (collections i roles)

Jeśli plik nazywa się requirements.yml:

ansible-galaxy install -r requirements.yml

Lub jeśli plik nazywa się requirements.txt (dla modułów Python):

pip3 install -r requirements.txt

Uwaga: Jeśli repo zawiera oba pliki, zainstaluj zależności z obu:

ansible-galaxy install -r requirements.yml

pip3 install -r requirements.txt

# 7. Modyfikacja inventory i group\_vars

Teraz musisz dostosować konfigurację Ansible do swoich potrzeb.

## Krok 7.1: Edycja pliku inventory

Plik inventory zawiera listę serwerów, którymi będziesz zarządzać. Znajdź plik inventory (może nazywać się inventory, hosts, lub inventory.ini):

ls inventory\* hosts\*

Edytuj plik za pomocą nano lub vim:

nano inventory

Przykładowa struktura pliku inventory:

[servers]

server1 ansible\_host=192.168.1.100 ansible\_user=admin

server2 ansible\_host=192.168.1.101 ansible\_user=admin

Zastąp wartości IP i użytkownika 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. Sprawdź jego zawartość:

ls group\_vars/

Edytuj odpowiedni plik (np. all.yml lub nazwę odpowiadającą Twojej grupie):

nano group\_vars/all.yml

Dostosuj zmienne do swoich potrzeb (np. ścieżki, porty, parametry konfiguracyjne).

# 8. Test połączenia za pomocą Ansible

## Krok 8.1: Ansible ping

Sprawdź, czy Ansible może połączyć się z Twoimi serwerami:

ansible all -m ping -i inventory

Jeśli wszystko działa poprawnie, zobaczysz komunikat SUCCESS dla każdego serwera:

server1 | SUCCESS => {

"ping": "pong"

}

## Krok 8.2: Rozwiązywanie problemów

Jeśli napotkasz błędy:

* Sprawdź, czy klucz SSH został poprawnie skopiowany
* Zweryfikuj IP serwera i nazwę użytkownika w inventory
* Upewnij się, że serwer jest włączony i dostępny w sieci
* Sprawdź, czy firewall nie blokuje połączenia SSH (port 22)

# 9. Wykonywanie playbooków krok po kroku

## Krok 9.1: Lista dostępnych playbooków

Sprawdź, jakie playbooki znajdują się w repozytorium:

ls \*.yml

## Krok 9.2: Wykonanie playbooka z flagą --step

Flaga --step pozwala na wykonywanie playbooka krok po kroku z potwierdzeniem przed każdą akcją. Jest to przydatne podczas testowania i nauki.

Podstawowa składnia:

ansible-playbook -i inventory nazwa\_playbooka.yml --step

Przykład:

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

## Krok 9.3: Interakcja podczas wykonywania --step

Po uruchomieniu z flagą --step, Ansible będzie pytał przed każdym zadaniem:

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

Możesz wybrać:

* y (yes) - wykonaj to zadanie i zapytaj o następne
* n (no) - pomiń to zadanie i przejdź do następnego
* c (continue) - wykonaj wszystkie pozostałe zadania bez pytania

## Krok 9.4: Dodatkowe przydatne flagi

Inne użyteczne flagi podczas wykonywania playbooków:

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

Tryb dry-run (--check) - pokazuje co zostałoby zmienione bez faktycznego wykonywania zmian.

ansible-playbook -i inventory site.yml -v

Zwiększony poziom szczegółowości (-v, -vv, -vvv) - pokazuje więcej informacji o wykonywanych działaniach.

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

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

# Podsumowanie

Gratulacje! Pomyślnie skonfigurowałeś kompletne środowisko do pracy z Ansible. Teraz możesz:

* Zarządzać serwerami za pomocą Ansible
* Wykonywać playbooki automatyzujące konfigurację
* Testować zmiany krok po kroku używając flagi --step
* Rozwijać 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!*