

# Laboratoria z Ansible

## Zarządzanie Systemami Rozproszonymi

October 6, 2024

### Wprowadzenie

**Cel laboratoriów:** Celem tych laboratoriów jest zapoznanie studentów z narzędziem Ansible oraz automatyzacją zarządzania konfiguracją. Studenci będą używać Ansible do instalacji oprogramowania, zarządzania plikami oraz konfiguracji systemów w sieci za pomocą playbooków YAML. Laboratoria kończą się zadaniem zbudowania klastra Redis z jednym węzłem Master i dwoma węzłami Slave.

**Wymagana wiedza:** Znajomość podstaw systemu Linux (komendy terminala) oraz podstawowa znajomość sieci (np. SSH, IP).

**Narzędzia:** - Maszyna kontrolna (np. openSUSE) z zainstalowanym Ansible. - Trzy maszyny wirtualne w tej samej sieci, na których będą wykonywane zadania. - Połączenie SSH między maszynami.

**Rezultat:** Studenci nauczą się używać Ansible do zarządzania wieloma maszynami jednocześnie za pomocą playbooków YAML. Efektem końcowym będzie działający klastr Redis.

## 1 Przygotowanie środowiska

**Instalacja Ansible na maszynie kontrolnej:** Na maszynie kontrolnej (np. openSUSE) należy zainstalować Ansible:

```
sudo zypper install ansible
```

Po instalacji upewnij się, że Ansible działa:

```
ansible --version
```

Link do dokumentacji Ansible: [https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html)

**Utworzenie pliku hosts:** W katalogu `/etc/ansible/` utwórz plik `hosts`, w którym wymienione będą maszyny docelowe:

```
[myservers]
vm1 ansible_host=192.168.1.101
vm2 ansible_host=192.168.1.102
vm3 ansible_host=192.168.1.103
```

## 2 Sprawdzenie połączenia

**Polecenie ping:** Aby upewnić się, że Ansible może połączyć się z maszynami:

```
ansible myservers -m ping
```

Połączone maszyny odpowiedzą "pong". Link do dokumentacji modułu ping:

[https://docs.ansible.com/ansible/latest/collections/ansible/builtin/ping\\_module.html](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/ping_module.html)

## 3 Zbieranie faktów

**Zbieranie faktów z maszyn:**

```
ansible myservers -m setup
```

Moduł `setup` dostarcza szczegółowych informacji o systemach (adresy IP, system operacyjny itd.).

Link do dokumentacji: [https://docs.ansible.com/ansible/latest/collections/ansible/builtin/setup\\_module.html](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/setup_module.html)

## 4 Instalacja Nginx za pomocą playbooka

**Playbook YAML do instalacji Nginx:** Od tego momentu studenci będą korzystać wyłącznie z playbooków w formacie YAML. Stwórz playbook o nazwie `nginx-install.yml`:

```
---
- hosts: myservers
  become: yes
  tasks:
    - name: Install Nginx
      apt:
        name: nginx
        state: present
```

Wykonaj playbook:

```
ansible-playbook nginx-install.yml
```

Link do dokumentacji modułu apt: [https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt\\_module.html](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html)

## 5 Modyfikacja strony głównej za pomocą playbooka

**Tworzenie i kopiowanie pliku `index.html`:** Napisz playbook o nazwie `update-index.yml`, który skopiuje plik `index.html` do katalogu Nginx:

```
—
- hosts: myservers
  become: yes
  tasks:
    - name: Copy index.html to all servers
      copy:
        src: /path/to/index.html
        dest: /var/www/html/index.html
```

Wykonaj playbook:

```
ansible-playbook update-index.yml
```

Link do dokumentacji modułu `copy`: [https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy\\_module.html](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html)

## 6 Użycie szablonu do dynamicznej strony

**Tworzenie dynamicznego szablonu strony głównej:** Napisz szablon `index.html.j2`, który będzie zawierał adres IP każdej maszyny:

```
<html>
  <head><title>Ansible Lab</title></head>
  <body>
    <h1>This is {{ ansible_default_ipv4.address }}</h1>
  </body>
</html>
```

Następnie napisz playbook o nazwie `template-index.yml`, aby wdrożyć szablon:

```
—
- hosts: myservers
  become: yes
  tasks:
    - name: Deploy template index.html
      template:
        src: /path/to/index.html.j2
        dest: /var/www/html/index.html
```

Wykonaj playbook:

```
ansible-playbook template-index.yml
```

Link do dokumentacji modułu `template`: [https://docs.ansible.com/ansible/latest/collections/ansible/builtin/template\\_module.html](https://docs.ansible.com/ansible/latest/collections/ansible/builtin/template_module.html)

## 7 Zadanie samodzielne: instalacja Redis (Master-Slave)

**Zadanie główne:** Zadanie to polega na skonfigurowaniu Redis w trybie Master-Slave z jedną maszyną Master i dwiema maszynami Slave.

**Polecenie:** Stwórz playbook, który zainstaluje Redis na wszystkich maszynach, a następnie skonfiguruj go, aby działał w trybie Master-Slave. Na jednej maszynie skonfiguruj Redis jako Master, a na dwóch pozostałych jako Slave.

**Weryfikacja:** Zaloguj się na maszynę Master i dodaj dane do Redis, a następnie sprawdź na maszynach Slave, czy dane zostały zreplikowane:

```
redis-cli set mykey "Ansible Redis"
redis-cli get mykey
```

Dane muszą być replikowane na maszynach Slave.

## Podsumowanie

W trakcie tych laboratoriów studenci nauczyli się korzystać z Ansible, aby zarządzać konfiguracją na wielu maszynach przy pomocy playbooków YAML. Zakończyliśmy laboratoria, budując klaster Redis z konfiguracją Master-Slave i zweryfikowaliśmy replikację danych.