

Zarządzanie Systemami Rozproszonymi

Laboratoria z Ansible

mgr inż. Jakub Woźniak

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).
- 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.

1 Przygotowanie środowiska

Teoria: W środowiskach rozproszonych ważne jest zapewnienie bezproblemowej komunikacji między maszynami. Dzięki Ansible możemy automatycznie zarządzać wieloma systemami jednocześnie. W tym laboratorium studenci skonfigurują trzy maszyny wirtualne w tej samej sieci, które będą mogły komunikować się ze sobą oraz z maszyną kontrolną przez SSH.

Tutorial: Konfiguracja maszyn wirtualnych w VirtualBox z użyciem gotowych obrazów

1. Pobranie gotowych obrazów systemu:

- **Ubuntu Minimal:** Pobierz obraz minimalny Ubuntu z Ubuntu Cloud.

- **openSUSE Minimal:** Pobierz minimalny obraz openSUSE z openSUSE Tumbleweed.

2. Konfiguracja sieci:

- Utwórz sieć typu "Host-Only" w VirtualBox, która pozwala na komunikację między maszynami.
- Dla każdej maszyny wirtualnej skonfiguruj Adapter 1 jako NAT (dostęp do internetu) i Adapter 2 jako Host-Only.

3. Importowanie maszyn wirtualnych:

- Utwórz nowe maszyny w VirtualBox, korzystając z pobranych obrazów jako dysków.

4. Konfiguracja bezhasłowego SSH:

- Na maszynie hosta wygeneruj parę kluczy SSH za pomocą `ssh-keygen -t rsa -b 4096`.
- Skopiuj klucz publiczny na każdą z maszyn wirtualnych, korzystając z `ssh-copy-id`.

5. Sprawdzenie komunikacji między maszynami:

- Użyj polecenia `ping`, aby upewnić się, że maszyny wirtualne mogą się ze sobą komunikować.

2 Sprawdzenie połączenia

Teoria: Pierwszym krokiem przy pracy z Ansible jest upewnienie się, że maszyna kontrolna może komunikować się z innymi maszynami. Ansible umożliwia to za pomocą modułu `ping`, który testuje dostępność maszyn w grupie.

Polecenie `ping`:

```
ansible myservers --m ping
```

To polecenie wysyła żądanie `ping` do każdej maszyny w grupie `myservers`, sprawdzając, czy jest dostępna.

3 Zbieranie faktów

Teoria: Moduł `gather_facts` pozwala zebrać szczegółowe informacje o maszynach, takie jak adresy IP, system operacyjny czy dostępne zasoby. Jest to przydatne przy dalszej automatyzacji.

Zbieranie faktów z maszyn:

```
ansible myservers --m gather_facts
```

Wyświetli to szczegółowe dane na temat każdej z maszyn w formacie JSON.

4 Instalacja Nginx za pomocą playbooka

Teoria: Playbooki Ansible to pliki YAML, które definiują zestawy instrukcji do wykonania na zdalnych maszynach. W tym kroku zainstalujemy serwer Nginx na wszystkich maszynach.

Playbook YAML do instalacji Nginx:

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

Uruchomienie playbooka:

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

5 Modyfikacja strony głównej za pomocą playbooka

Teoria: Ansible umożliwia automatyczne zarządzanie plikami na zdalnych maszynach. W tym kroku skopiujemy plik `index.html` na wszystkie maszyny, zmieniając stronę główną Nginx.

Playbook YAML do skopiowania pliku `index.html`:

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

Uruchomienie playbooka:

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

6 Wykorzystanie szablonu do dynamicznej strony

Teoria: Ansible pozwala na korzystanie z szablonów, które dynamicznie generują pliki na podstawie zmiennych. W tym kroku stworzymy szablon `index.html`, który będzie wyświetlał adres IP każdej maszyny.

Szablon `index.html.j2` z dynamicznymi zmiennymi:

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

Playbook YAML do wdrożenia szablonu:

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

Uruchomienie playbooka:

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

7 Instalacja Redis (Master-Slave)

Teoria: Redis jest bazą danych typu key-value, która obsługuje replikację danych w trybie Master-Slave. W tym kroku skonfigurujemy klaster Redis z jednym węzłem Master i dwoma węzłami Slave.

Zadanie: Stwórz playbook, który zainstaluje Redis na wszystkich maszynach i skonfiguruje replikację Master-Slave. Po zakończeniu zadania sprawdź, czy dane są replikowane między węzłami.

Kryteria zaliczenia:

- Poprawnie skonfigurowany klaster Redis z jednym Master i dwoma Slave.
- Weryfikacja, że dane dodane na węzle Master są replikowane na węzłach Slave.