**LAB 4**

Playbooks

Zawartość dokumentu:

[**Plan ćwiczenia**](#_6m59hr95si7n) **1**

[**Omówienie użytych plików**](#_tl7di44m44dq) **2**

Wstęp

Za pomocą Ansible możemy uruchamiać pojedyncze komendy lub całe plany wykonania (skrypty) w formie Playbooków. Playbooki, podobnie jak skrypty, nadzorują jednakowe wykonanie zadań przy każdym przebiegu. Dodatkowo pozwalają zdefiniować szereg zmiennych i parametrów przydatnych do efektywnego użycia Ansible.

Wykonamy komendy z poprzedniego laboratorium za pomocą playbooka.

## Plan ćwiczenia

1. Włączamy nasz ulubiony terminal (Windows: Putty; Ubuntu: Terminal; macOS: Mac Terminal)
2. Logujemy się do serwera zarządzającego mgmt2.muszynski.pro
   * ssh mgmt3.muszynski.pro -l <id\_uzytkownika>
   * Hasło podane przez prowadzącego
3. Znajdujemy się w katalogu domowym  
   nasz\_uzytkownik@mgmt-server:/home/nasz\_uzytkownik$
4. Przechodzimy do katalogu szkolenie
   * cd akademiaansible
5. Ładujemy środowisko (można dodać do np do .bashrc)
   * source source\_me.sh  
     Oczekiwany rezultat:
   * **(venv)** nasz\_uzytkownik@akademiaansible-server:~/akademiaansible$
6. Sprawdź zawartość pliku inventory/inventory\_default
   * W pliku tym występuje grupa ‘apache’, ‘backend’, ‘prod’
7. Zapoznaj się z przykładowym playbookiem
   * playbooks/install\_apache.yml
8. Uruchom playbook
   * (venv)$ ansible-playbook playbooks/install\_apache.yml -vvv
   * Aplikacje na niektórych serwerach mogą być już zainstalowane przez Twojego kolegę  
     (mówi o tym status changed: false)
9. Zmodyfikuj playbook,
   * zmodyfikuj listę argumentów zadania ‘install packages’ o kolejną aplikację ‘dstat’
   * uruchom ponownie playbook - pakiet dstat zostanie zainstalowany
10. Jak w poprzednim laboratorium:
    * Dopisz zadanie (task), które stworzy (lub ustawi) **plik**
      1. ~/szkolenie\_playbooks<nazwa\_uzytkownika>
      2. Uprawnienia 0755
    * wykorzystaj ponownie moduł ***file touch***
    * przykładowe rozwiązanie na następnej stronie
11. Zadanie dodatkowe - napisz samodzielnie nowy playbook moj\_pierwszy\_playbook.yml za pomocą którego utworzysz (przykładową) plik /tmp/host\_is\_ready<nazwa\_uzytkownika> o zawartości (treści) OK   
    Zadanie to można zrealizować na kilka sposobów
    * za pomocą modułu ‘shell’ i komend bash (np echo)
    * za pomocą modułu template
    * za pomocą modułu **copy + content**
    * za pomocą modułu **copy** (wymaga uprzedniego przygotowania pliku)

polecam użycie modułu template

1. Zadanie dodatkowe: napisz playbook gdzie zainstalujesz apache2 za pomocą modułu *shell*.  
   Wykonaj playbooki z parametrem --check.  
   Porównaj ręczną pracę, pracę modułem shell do playbooka wykorzystującego moduły ansible.
2. Zadanie dodatkowe: użyj ‘creates’ w połącz z modułem shell  
   Propozycja komendy - wygeneruj klucz ssh.  
   ssh-keygen -f /tmp/klucz  
     
   Użyj:  
   args:  
    creates: /tmp/klucz  
   Jeśli ten plik będzie istniał, zadanie zostanie pominięte (jest to forma semafora dla zadania realizowanego za pomocą pliku)

## Omówienie użytych plików

$ cat playbooks/install\_apache.yml   
---  
#Uruchomienie:  
#~/akademiaansible$ ansible-playbook playbooks/install\_apache.yml  
  
- hosts: apache  
  
 tasks:  
 - name: update cache   
 apt:

update\_cache: yes

cache\_valid\_time: 7200  
 changed\_when: False  
 become: True  
  
 - name: install packages  
 apt:

pkg:  
 - apache2  
 - htop

become: True  
  
 - name: Wait for instances to listen on port 80  
 wait\_for: state=started host=localhost port=80 timeout=15 delay=5

Jak widać powyżej, uruchamiamy rolę w kontekście grupy hostów *apache*. W sekcji *tasks* zdefiniowano kolejno wykonywane moduły. Niektóre zadania mają dodatkową flagę *become: True*mówiącą o tym, że maja być wykonane nako superuser (sudo)

Odpowiedź:

$ cat playbooks/install\_apache\_**odpowiedz**.yml   
---  
#Uruchomienie:  
#~/szkolenie$ ansible-playbook playbooks/install\_apache.yml  
  
- hosts: apache  
 become: True  
  
 tasks:  
 - name: update cache   
 apt: update\_cache=yes cache\_valid\_time=7200  
 changed\_when: False  
  
 - name: install packages  
 apt: name={{ item }}  
 with\_items:  
 - apache2  
 - htop

- dstat  
  
 - name: Wait for instances to listen on port 80  
 wait\_for: state=started host=localhost port=80 timeout=15 delay=5

# odpowiedz na 3 sposoby

- name: create file   
 file:

state: touch

path: ~/szkolenie\_playbooks\_szkolenie01

mode: 0775

- name: create file

file:

path: "~/szkolenie\_playbooks\_{{ nazwa\_uzytkownika }}"

state: touch

mode: 0775

vars:

nazwa\_uzytkownika: szkolenie01

- name: create file with lookup

file:

path: "~/szkolenie\_playbooks\_{{ nazwa\_uzytkownika }}"

state: touch

mode: 0775

vars:

nazwa\_uzytkownika: "{{ lookup('pipe', 'whoami') }}"