**LAB 7**

Roles

Zawartość dokumentu:

[**Wstęp**](#_hufwkpy27owu) **1**

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

[Instalacja gotowej roli](#_mbo3y5dxperc) 2

[**Omówienie użytych plików**](#_tl7di44m44dq) **3**

## Wstęp

Znamy już sposoby użycia *modułów* przy pomocy *Ansible* oraz *Playbooków* za pomocą *ansible-playbook*. Jak zapewne możemy zauważyć, istnieją przypadki, gdy podobne zadanie (kawałek skryptu), niekiedy różniące się drobną zmienną, chcemy wykonać na różnych grupach serwerów. Aby nie duplikować kodu, można te sekcje zamknąć w “paczki”. Dodatkowo mogą one zawierać logikę wykonania, zadania warunkowe (np *handlers*), domyślne zmienne (defaults), szablony (templates) i wiele innych.

## 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\_akademiaansible
   * W pliku tym występuje grupa ‘apache’, ‘backend’, ‘prod’
7. Zapoznaj się z przykładowym Playbookiem
   * playbooks/lab\_roles.yml
   * (ps - podpowiedź w playbooks/lab\_roles\_odpowiedz.yml)
8. Uruchom Playbook
   * (venv)$ ansible-playbook playbooks/lab\_roles.yml -vvv
9. Zauważ, w jakiej kolejności wykonują się poszczególne komendy (“przykladowa\_rola” “Timestamp” oraz “show time”) !
10. Zmodyfikuj Playbook, tak aby wykonana została dodatkowo rola
    * sys\_init\_common  
      rola ta istnieje - znajduje się w katalogu akademiaansible/roles/sys\_init\_common
11. Jak w poprzednich laboratoriach - poznamy role tworząc przy ich pomocy katalog - tym razem przy pomocy roli, którą będziemy zaraz stworzysz:
    * za pomocą tej roli utwórz katalog ~/szkolenie\_roles<nazwa\_uzytkownika> (podobnie jak w poprzednich laboratoriach)   
      przykładowe kroki:
    * Napisz *prostą rolę [app\_stworz\_katalog]* - czyli w katalogu szkolenie3/playbook/roles/ utwórz katalog wraz z podkatalogami  
       np:   
       mkdir roles/app\_stworz\_katalog  
       mkdir roles/app\_stworz\_katalog/tasks
    * w tym katalogu stwórz plik main.yml  
       touch roles/app\_stworz\_katalog/tasks/main.yml
    * w tym pliku umieść pojedyncze zadanie tworzenia katalogu (możesz je skopiować z playbooka install\_apache.yml z Lab Playbooks)
    * Chcemy aby katalog miał uprawnienia 0754 (koniecznie w formie oktalnej)
    * <http://docs.ansible.com/ansible/playbooks_roles.html#roles>
    * *zamiast tworzyć katalog, podkatalogi i pliki można skopiować inną istniejącą rolę, lub użyć   
      ansible-galaxy init nazwa\_twojej\_roli  
      osobiście nie jestem fanem tego rozwiązania bo tworzy dodatkowe puste katalogi*[*https://galaxy.ansible.com/docs/contributing/creating\_role.html*](https://galaxy.ansible.com/docs/contributing/creating_role.html)

### Instalacja gotowej roli

Zainstalujmy teraz apache przy pomocy roli pobranej z ansible-galaxy - użyjemy gotowego rozwiązania!

1. Ansible-Galaxy:
   * Instalacja gotowej roli <https://galaxy.ansible.com/geerlingguy/apache/>
   * Zapoznaj się tez z <https://github.com/geerlingguy/ansible-role-apache> (role publikowane są często na github.com)
   * (venv)$: ansible-galaxy install geerlingguy.apache --roles-path=~/akademiaansible/roles/
   * Dopisz do listy roli w playbook/lab\_roles.yml w miejsce zakomentowanej roli *apache2* - rolę: geerlingguy.apache (odkomentuj)
   * Uruchom:
     1. (venv)$ ansible-playbook playbooks/lab\_roles.yml -vvv
2. Limit wykonania
   * Przy użyciu parametru *when,* ogranicz wykonanie roli - wykonają tylko na pierwszym hoscie z grupy apache np:
     1. when: groups[‘apache’][0]
     2. przykładowe rozwiązanie na samym końcu instrukcji
   * Wykonaj playbooka używając
     1. (venv)$ <komenda> --limit <ip\_lub\_nazwa\_hosta>
     2. aktualnie zdefiniowane hosty możesz poznać uruchamiając playbook z flagą --list-hosts

## Omówienie użytych plików

$ cat playbooks/lab\_roles.yml   
---

- hosts: apache  
 become: True  
  
 tasks:  
 - name: "show time"  
 shell: date  
  
 roles:  
 - przykladowa\_rola  
# - { role: apache2 }  
# - geerlingguy.apache

Jak widać powyżej, uruchamiamy rolę w kontekście grupy hostów *apache*. W sekcji *tasks* kolejno zdefiniowane są wykonywane moduły (tutaj tylko jedne: shell). Dodatkowo istnieje sekcja *vars* zawierająca zmienne lokalne dla Playbooka, oraz sekcja *pre\_tasks,* która wykona się przed innymi sekcjami.

**Rozwiązanie:**

$ cat playbooks/lab\_roles.yml   
---

- hosts: apache  
 become: True  
  
 tasks:  
 - name: "show time"  
 shell: date  
  
 roles:  
 - przykladowa\_rola

- { role: sys\_init\_common, become: True, when: "inventory\_hostname == groups['apache'][0]" }  
 - role: geerlingguy.apache

when: inventory\_hostname == groups['apache'][0]