**LAB 1**

Konfiguracja środowiska

Zawartość dokumentu:

[**Hasła**](#_4guc9ggfd7rr) **1**

[**Wstęp**](#_6k328zuohn72) **2**

[**Virtualenv**](#_tz1xjmcyugd7) **2**

[**Plan ćwiczenia**](#_dy1hk6ux4kyg) **2**

[**Wyjaśnienia do instalacji**](#_xub69rfqz068) **4**

[Przykładowy układ katalogów](#_b7pnmvcddslg) 4

[environment.sh](#_c4vkwwdzespq) 4

[ansible.cfg](#_o1ungricrlua) 6

[requirements.txt](#_uau24ltr1yj0) 8

## Hasła

Hasło do Ansible Vault (potrzebne w dalszej części zadania - punkt 8):

szkoleniex.TAJNe!

github

<https://github.com/sirkubax/akademiaansible>

**UWAGA**

Użytkownicy na serwer mgmt:

**ssh -l <user> mgmt3.muszynski.pro**

User: szkolenieXX - gdzie XX to numer przyznany przez prowadzącego

**Pass**: Tajne12345# - hasło użytkownika

## Wstęp

Istnieje wiele sposobów instalacji Ansible [docs.ansible.com/intro\_installation.html]. Do najbardziej popularnych należy wykorzystanie systemowego managera pakietów lub ekosystemu pythona (pip install ansible). Opcje te wymagają zazwyczaj podniesionych uprawnień użytkownika.

Sugeruję użycie wirtualnego środowiska uruchomieniowego **virtualenv.**

Kolejnym krokiem ćwiczenia będzie konfiguracja klucza szyfrującego do sekretów znajdujących się w repozytorium.

## Virtualenv

* Instalacja w kontekście użytkownika (nie wymaga podwyższonych uprawnień)
* Łatwo zainstalować nową wersję Ansible (z zachowaniem poprzedniej)
* W różnych katalogach virtualenv mogą jednocześnie występować (i być używane) różne wersje Ansible (np. było to przydatne przy migracji z v1.9.x do v2.x)
* **Wada**: konieczność załadowania środowiska (source venv/bin/activate)
  + [można to obejść np ładując w .bashrc]

W tej części praktycznej zainstalujemy i skonfigurujemy środowisko ansible, przy wykorzystaniu środowiska virtualenv.

## Plan ćwiczenia

1. Włączamy nasz ulubiony terminal (Windows: Putty; Ubuntu: Terminal; macOS: Mac Terminal)
2. Logujemy się do serwera zarządzającego mgmt3.muszynski.pro
   * ssh mgmt3.muszynski.pro -l <user>
   * Hasło podane przez prowadzącego
3. Znajdujemy się w katalogu domowym  
   nasz\_uzytkownik@mgmt-server:/home/nasz\_uzytkownik$
4. Zklonuj repozytorium git  
   git clone <https://github.com/sirkubax/akademiaansible>
5. Przechodzimy do katalogu szkolenie
   * cd akademiaansible
6. Ładujemy środowisko (można dodać do np do .bashrc)
   * source source\_me.sh  
     Oczekiwany rezultat:
   * **(venv)** nasz\_uzytkownik@akademiaansible-server:~/akademiaansible$
7. Sprawdzanie zawartości pliku akademiaansible/playbooks/vars/szkolenie\_vault.yml
   * Czy jesteś wstanie odczytać zawartość tego pliku? Czy dane w nim zawarte są bezpieczne?
8. Tworzenie pliku ~/.ssh/vault\_pass\_11.txt
   * echo “szkoleniex.TAJNe!” > ~/.ssh/vault\_pass\_11.txt
9. Sprawdzamy działanie wykonując najprostszą komendę:
   * ansible -m ping localhost  
     localhost | SUCCESS => {  
      "changed": false,   
      "ping": "pong"  
     }

## Wyjaśnienia do instalacji

### Przykładowy układ katalogów

.

├── ansible.cfg

├── environment.sh

├── inventory

│ └── inventory\_default

├── opt

│ └── my\_env.tgz

├── playbooks

│ ├── create\_aws\_instance.yml

│ ├── init\_mgmt.yml

│ ├── lab\_a\_test\_hash\_behaviour.yml

│ └── vars

├── roles

├── README.md

├── requirements.txt

└── venv

### source\_me.sh

Jest to zbiór dobrych praktyk, wymaganych pakietów i konfiguracji środowiska, które wypracowaliśmy na przestrzeni lat.

#! /bin/bash  
export ANS\_ROOT\_DIR="$(readlink -m $(readlink -m "$(dirname "${BASH\_SOURCE[0]}")"))"  
export EC2\_INI\_PATH=$ANS\_ROOT\_DIR/etc/inv/ec2.ini  
export ANSIBLE\_VAULT\_PASSWORD\_FILE=~/.ssh/vault\_pass\_11.txt  
export ANSIBLE\_CONFIG="$(pwd)/ansible.cfg"  
#export ANSIBLE\_KEEP\_REMOTE\_FILES=1  
  
while getopts ":i" \_\_optarg; do  
 case "$\_\_optarg" in  
 i)  
 \_\_install\_requirements=true  
 ;;  
 esac  
done  
  
if [ -z "$VIRTUAL\_ENV" ]; then  
 if ! [ -d $ANS\_ROOT\_DIR/venv ]; then  
 echo '-- No virtualenv detected; creating'  
 virtualenv $ANS\_ROOT\_DIR/venv || (echo '!!! virtualenv creation failed' && return)  
 mkdir ~/.ansible

touch ~/.ssh/vault\_pass\_11.txt  
 \_\_venv\_created=true  
 fi  
 source $ANS\_ROOT\_DIR/venv2/bin/activate  
fi  
  
  
if [ "$\_\_venv\_created" = true ] || [ "$\_\_install\_requirements" = true ]; then  
 echo '-- Installing requirements'  
 pip install -r $ANS\_ROOT\_DIR/requirements.txt || (echo '!!! pip install failed' && return)  
fi  
  
  
if ! declare -f \_\_old\_deactivate 2>&1 > /dev/null; then  
 \_\_old\_deactivate\_source=$(declare -f deactivate)  
 eval "\_\_old\_deactivate${\_\_old\_deactivate\_source#deactivate}"  
 function deactivate {  
 \_\_old\_deactivate  
 unset ANS\_ROOT\_DIR

Objaśnienia najważniejszych fragmentów skryptu:

* ANS\_ROOT\_DIR   
  Skomplikowanie wyglądająca linia, której celem jest określenie katalogu, w którym znajduje się plik.
* EC2\_INI\_PATH   
  W przypadku użycia AWS - plik dynamicznego inventory (uwaga - do działania wymaga poprawnej konfiguracji poświadczeń IAM/AWS).
* ANSIBLE\_VAULT\_PASSWORD\_FILE   
  Klucz do szyfrowania/czytania naszych sekretów.  
  (Opcja ta może być skonfigurowana również w ansible.cfg, ale umieszczenie jej tutaj, pozwala na łatwe stosowanie wielu sekretów).
* ANSIBLE\_SSH\_CONTROL\_PATH   
  [Opcja] Przyspieszenie logowania po ssh.
* ANSIBLE\_KEEP\_REMOTE\_FILES=1   
  Bardzo przydatna opcja przy debugowaniu, pozostawia prekompilowane pliki wykonywalne na maszynie zdalnej (można je uruchomić lokalnie).
* virtualenv nazwa\_katalogu  
  virtualenv $ANS\_ROOT\_DIR/venv   
  Przygotowanie pustego środowiska virtualenv.
* pip install -r $ANS\_ROOT\_DIR/requirements.txt   
  W tym kroku dokonuje się właściwa instalacja.
* source $ANS\_ROOT\_DIR/venv/bin/activate   
  Załadowanie (w kontekście naszej sesji) środowiska Ansible.

### 

### ansible.cfg

Plik konfiguracyjny. Domyślnie gotowy do uruchomienia. Warto rozważyć kilka opcji konfiguracyjnych:

* *inventory*  
  Domyślne inventory, używane przez program
* *gathering*  
  # smart - gather by default, but don't regather if already gathered  
  # implicit - gather by default, turn off with gather\_facts: False  
  # explicit - do not gather by default, must say gather\_facts: True  
  gathering = smart
  + Fakty i informacje potrzebne do pracy, pobierane będą jedynie gdy wygaśnie ważność cache
* *log\_path*   
  Rzadko tu zaglądamy, ale warto wiedzieć, że są.
* *fact\_caching = jsonfile  
  fact\_caching\_connection = /home/<user>/facts\_cache  
  fact\_caching\_timeout = 86400*To ustawienia, pozwalające na przechowywanie zebranych faktów. Dzięki temu możliwe jest np. odwołanie się do informacji o wielkości pamięci hosta X bez pobierania informacji o jego parametrach.
* *ssh\_args=-o PasswordAuthentication=no -o ForwardAgent=yes -o ControlMaster=auto -o ControlPersist=60s*Mamy możliwość wpływania na parametry połączenia ssh.
* pipelining = True  
  Przyspiesza wykonywanie komend przez zmniejszenie ilości wykonywanych operacji ssh.
* *host\_key\_checking = False*Czy sprawdzać host względem pliku .ssh/known\_hosts
* *callback\_whitelist = profile\_tasks*Włącza profilowanie - po wykonaniu zadania. Otrzymujemy listę zadań i czasy wykonania.

Przykładowo plik ansible.cfg jest dostarczany z pakietem paczek systemowych

$ dpkg -L ansible |less

/.

/etc

/etc/ansible

/etc/ansible/hosts

/etc/ansible/**ansible.cfg**

/usr

/usr/bin

/usr/bin/ansible-playbook

/usr/bin/ansible-pull

/usr/bin/ansible-galaxy

/usr/bin/ansible-vault

/usr/bin/ansible-doc

/usr/bin/ansible

/usr/lib

/usr/lib/python2.7/(...)/ansible/

(...)

Najbardziej aktualny plik z domyślnymi (zakomentowanymi) wartościami dostępny zawsze na stronie projektu:

<https://github.com/ansible/ansible/blob/devel/examples/ansible.cfg>

Lub ze stabilnego brancha:

<https://github.com/ansible/ansible/blob/stable-2.3/examples/ansible.cfg>

### requirements.txt

https://releases.ansible.com/ansible/ansible-2.6.5.tar.gz

boto==2.49.0

Instalacja (wewnątrz skryptu environments.sh) za pomocą komendy:

pip install -r $ANS\_ROOT\_DIR/requirements.txt

Pip to jeden ze sposobów instalacji pakietów python. Opcja -r pozwala na podanie pliku wejściowego z nazwami pakietów do instalacji.

Sama instalacja Ansible przy pomocy pip również może być wykonana na co najmniej dwa sposoby.

Instalacja bezpośrednio z github.com będzie bardziej aktutalna, ale na starszych systemach zauważyliśmy bug w inicjalizacji ‘git submodules’, co skłoniło nas do instalacji konkretnej opublikowanej wersji: ‘Releases’.

[Komentarz - *Ansible Inc*. odchodzi od użycia submodułów w swoim projekcie, więc powyższe ograniczenie staje się nieaktualne].