

**Lab 1: Przekształć skrypt z serwerem www w playbooka.  
Wykorzystaj moduł uri do sprawdzenia, czy strona działa.**

```
[ansible@base ~]# cat serwer_www.yml
```

```
- name: Konfiguracja serwera www – PLAY 1
  hosts: servers
  become: True
  tasks:
    - name: Instalacja oprogramowania
      dnf:
        name: httpd
        state: latest

    - name: Uruchomienie service
      service:
        name: httpd
        state: started
        enabled: yes
        ignore_errors: yes

    - name: Plik index.html
      copy:
        dest: /var/www/html/index.html
        content: "Witaj w swiecie Ansible"

    - name: Konfiguracja fiwalld
      firewalld:
        service: http
        immediate: yes
        permanent: yes
        state: enabled
```

```
- name: Sprawdzanie stanu serwera www – PLAY 2
hosts: localhost
become: False
tasks:
  - name: Zwrocenie contentu strony
    uri:
      url: http://poznan
      return_content: yes
      status_code: 200
    register: www_content

  - name: Wyświetlanie contentu
    debug:
      var: www_content.content
```

## Lab 2: Wykorzystaj zmienne w playbooku. Opcja `vars`

```
[ansible@base ~]# cat users.yml
---
- name: Zarządzanie users - groups
  become: True
  hosts: servers
  vars:
    users:
      - john
      - bob
      - tom
  tasks:
    - name: Tworzenie grupy
```

```
group:
  name: administrator
  state: present

- name: Tworzenie userow
  user:
    name: '{{ item }}'
    state: present
    group: administrator
  loop: '{{ users }}'
```

**LAB 3: Dodaj grupy i użytkowników do zarządzanych serwerów. Wykorzystaj zmienne w playbooku.**

```
[ansible@base ~]$ cat users2.yml
```

---

- name: Zarządzanie users - groups

become: True

hosts: servers

vars:

users:

- imie: "john"

grupa: "administrator"

- imie: "bob"

grupa: "admin"

tasks:

- name: Tworzenie grupy

group:

name: '{{ item }}'

state: present

loop:

- administrator

- admin

- name: Tworzenie userow

user:

name: '{{ item.imie }}'

state: present

group: '{{ item.grupa }}'

loop: '{{ users }}'

db

www

**LAB 4: Dodaj grupy i użytkowników do zarządzanych serwerów. Wykorzystaj zmienne do playbooka umieszczone w pliku zewnętrznym.**

```
[ansible@base ~]$ cat users3.yml
```

```
---
```

```
- name: Zarzadzanie users - groups
  become: True
  hosts: servers
  vars_files: lista_pracownikow.yml
  tasks:
    - name: Tworzenie grupy
      group:
        name: '{{ item }}'
        state: present
      loop:
        - administrator
        - admin

    - name: Tworzenie userow
      user:
        name: '{{ item.imie }}'
        state: present
        group: '{{ item.grupa }}'
      loop: '{{ users }}'
```

```
[ansible@base ~]$ cat lista_pracownikow.yml
```

```
users:
  - { imie: "john", grupa: "administrator" }
  - { imie: "bob", grupa: "admin" }
```

**LAB 5: Sprawdź uprawnienia pliku /etc/hosts na poznaniu i gdansku. Jeśli uprawnienia są równe 0644, to zmień je na 0600. Przed zmianą wyświetl uprawnienia pliku.**

```
[ansible@base ~]$ cat uprawnienia.yml
```

```
---
```

```
- name: Sprawdzam uprawnienia plikow
  hosts: poznan
  become: True
  vars:
    plik: /etc/hosts
  tasks:
    - name: Pobranie info o pliku
      stat:
        path: '{{ plik }}'
        register: spr_plik

    - name: Wyświetlenie info o pliku
      debug:
        var: spr_plik.stat.mode

    - name: Zmiana uprawnień do pliku
      file:
        path: '{{ plik }}'
        mode: 0600
      when: spr_plik.stat.mode == "0644"
```

**LAB 6: Utwórz użytkownika admin1, który należy do podstawowej grupy(sprawdź czy istnieje) admins. Użyj zmiennych uid=2001, comment="Pracownik IT", password="altkom"**

```
[ansible@base ~]$ cat playbook.yml
```

```
---
```

```
- name: Tworzenie userow
```

```
  hosts: servers
```

```
  gather_facts: False
```

```
  become: True
```

```
  tasks:
```

```
    - name: Tworzenie grupy admins
```

```
      group:
```

```
        name: admins
```

```
    - name: Tworzenie uzytkownika admin1
```

```
      user:
```

```
        name: admin1
```

```
        group: admins
```

```
        uid: 2001
```

```
        comment: "pracownik IT"
```

```
        password:
```

```
$6$oWTeP6NS9YYU4P.m$2r3EV6Zq6/yqvMmHIGhA.PTCAYqmlfbWgBb9fhVgjGdygA  
/EuuNGyoROB3HAtTwjps6Vgp7UQwAaHolJ4XVEY.
```

## LAB 7: Wykorzystaj handlers w playbooku. Zablokuj bezpośrednie logowanie się usera root na wszystkich serwerach zarządzanych.

```
[ansible@base ~]$ cat handlers.yml
```

```
---
```

```
- name: Wykorzystanie handlers
  hosts: servers
  become: True
  tasks:
    - name: Zmiana pliku konfiguracyjnego sshd_config
      lineinfile:
        path: /etc/ssh/sshd_config
        regexp: "^PermitRootLogin"
        line: "PermitRootLogin no"
      notify: restart

  handlers:
    - name: restart
      service:
        name: sshd
        state: restarted
```

## LAB 8: Konfiguracja serwera NFS i klientów

```
[ansible@base ~]$ cat nfs.yml
```

```
---
```

```
- name: Instalacja oprogramowania na potrzeby serwera NFS
  become: True
  hosts: all
  tasks:
```



- name: Instalacja paczki nfs-utils  
dnf:
  - name: nfs-utils
  - state: latest
- name: Konfiguracja serwera NFS  
hosts: gdansk  
become: True  
tasks:
  - name: Przygotowanie partycji  
parted:
    - device: /dev/vdb
    - number: 1
    - state: present
    - part\_end: 2GiB
  - name: Przygotowanie filesystemu  
filesystem:
    - fstype: ext4
    - dev: /dev/vdb1
  - name: Przygotowanie pkt montowania  
file:
    - path: /srv/nfs
    - state: directory
  - name: Dodawanie /dev/vdb2 do /etc/fstab  
mount:
    - path: /srv/nfs
    - src: /dev/vdb1
    - fstype: ext4
    - state: mounted

- name: montowanie /dev/vdb1  
shell: "mount -a"
- name: Plik kontrolny  
copy:  
dest: /srv/nfs/plik1  
content: "NFS zamontowany pomyslnie"

- name: Konfiguracja firewalld  
firewalld:  
service: nfs  
immediate: yes  
permanent: yes  
state: enabled

- name: Startowanie serwisu NFS  
service:  
name: nfs-server  
state: started  
enabled: yes

- name: Edycja pliku /etc/exports  
copy:  
dest: /etc/exports  
content: "/srv/nfs \*(ro)"  
notify: exportfs

handlers:

- name: exportfs  
shell: 'exportfs -rv'
- name: Montowanie udziału NFS na katowice i poznan

```
hosts: nfs_client
become: True
tasks:
  - name: Tworzenie katalogu /mnt/nfs
    file:
      path: /mnt/nfs
      state: directory

  - name: Wpis udziału NFS do /etc/fstab
    mount:
      path: /mnt/nfs
      src: gdansk:/srv/nfs
      fstype: nfs
      state: mounted

  - name:
    shell: 'mount -a'

  - name: Sprawdzanie konfiguracji
    shell: cat /mnt/nfs/plik1
    register: plik_nfs

  - name: Wyszukiwanie pliku z NFS
    debug:
      var: plik_nfs.stdout
```

## LAB 9.

---

```
- name: Konfiguracja serwera ssh na porcie 2222
  become: True
  hosts: servers
```

tasks:

- name: Wlaczanie SELinux w trybie Enforcing po reboocie  
selinux:
  - policy: targeted
  - state: enforcing
  
- name: Wlaczanie SELinux  
shell: setenforce 1
  
- name: Zabezpieczenie przed logowaniem na konto root  
lineinfile:
  - regexp: "^PermitRootLogin"
  - line: "PermitRootLogin no"
  - path: /etc/ssh/sshd\_confignotify: restart\_sshd
  
- name: Wylaczenie uwierzytelniania przez haslo  
lineinfile:
  - regexp: "^PasswordAuthentication"
  - line: "PasswordAuthentication no"
  - path: /etc/ssh/sshd\_confignotify: restart\_sshd
  
- name: Zmiana port sshd na 2222  
lineinfile:
  - regexp: "^#Port 22"
  - line: "Port 2222"
  - path: /etc/ssh/sshd\_confignotify: restart\_sshd
  
- name: Konfiguracja firewalld  
firewalld:

port: 2222/tcp  
permanent: yes  
immediate: yes  
state: enabled

- name: Zmiana polityki portu 2222

seport:

setype: ssh\_port\_t  
ports: 2222  
state: present  
proto: tcp

handlers:

- name: restart\_sshd

service:

name: sshd  
state: restarted