

# Od zera do Endera

Ansible w niebieskiej pigułce

Autor: Marcin Steć  
Wersja: 1.0  
Projekt: Konferencja Common Wiosna 2024  
Klient: Common Polska

## Spis treści

1 Przygotowanie środowiska.....	3
1.1 Przygotowanie OS.....	3
2 Test środowiska.....	5
2.1 Konfigurowanie Git do wypychania commitów.....	5
2.2 Konfiguracja projektu Ansible.....	11
3 Moduł <i>ibm.storage_virtualize</i> .....	14
3.1 Instalacja i test połączenia.....	14
3.2 Test kolekcji <i>ibm.storage_virtualize</i> .....	14
3.3 Przygotowanie dysku.....	16
4 Tworzenie LPARu – <i>ibm.power_hmc</i> .....	20
4.1 Instalacja kolekcji <i>ibm.power_hmc</i> .....	20
4.2 Test kolekcji <i>ibm.power_hmc</i> .....	21
4.3 Tworzenie i aktywacja LPARu.....	23
5 Majsterkowanie w VIOS.....	30
5.1 Instalacja kolekcji <i>ibm.power_vios</i> .....	30
5.2 Przygotowanie serwerów VIOS do pracy z Ansible.....	31
5.3 Test kolekcji <i>ibm.power_vios</i> .....	35
6 Użyteczne informacje i linki.....	37

# 1 Przygotowanie środowiska

Zanotuj szczegóły Twojego środowiska szkoleniowego:

SID/Hasło sieci szkoleniowej	<b>M.I.L.F / C0mm0n@Gdyn1@</b>
IP stacji Ansible	<b>ender.iic / 10.10.13.24</b>
Macierz IBM FlashSystem	<b>fs5200-2.iic / 10.10.181.74</b>
HMC	
User/Hasło	<b>common__</b> (__) numer przydzielony przez prowadzącego) <b>C0mm0n@Gdyn1@</b>
User/Hasło HMC	
User/Hasło IBM	

1. Podłącz się do WIFI
2. Zaloguj się do stacji kontrolnej Ansible

## 1.1 Przygotowanie OS

System szkoleniowy został przygotowany, dlatego poniższe instrukcje są tu umieszczone z kronikarskiego obowiązku, na wypadek, gdyby ktoś chciał to jednak przećwiczyć od początku ;-)

1. Zweryfikuj i ewentualnie zainstaluj pakiety OS potrzebne do pracy:
  - ansible
  - ansible-core
  - python3-passlib

Na systemach RedHatopodobnych użyj komendy:

```
marcinek@ender:~$ rpm -qa | grep -e ansible -e python3-passlib -e "^git"
git-core-2.44.0-1.fc39.x86_64
git-core-doc-2.44.0-1.fc39.noarch
git-2.44.0-1.fc39.x86_64
ansible-srpm-macros-1-12.fc39.noarch
```

---

```
ansible-core-2.16.5-1.fc39.noarch
ansible-9.4.0-1.fc39.noarch
python3-passlib-1.7.4-13.fc39.noarch
```

---

## 2. Sprawdź zainstalowane kolekcje Ansible:

---

```
marcinek@ender:~$ ansible-galaxy collection list

# /usr/lib/python3.12/site-packages/ansible_collections
Collection                                         Version
-----
amazon.aws                                       7.4.0
ansible.netcommon                               5.3.0
ansible.posix                                   1.5.4
ansible.utils                                   2.12.0
ansible.windows                                 2.3.0
arista.eos                                       6.2.2

[ ... ]

ibm.qradar                                       2.1.0
ibm.spectrum_virtualize                         2.0.0
ibm.storage_virtualize                         2.3.1

[ ... ]
```

---

Zwróć uwagę, że niektóre kolekcje IBM są częścią standardowej instalacji pakietów Ansible. np. Fedora ma aktualny!! pakiet *ibm.storage\_virtualize*. Ale nie należy się do tego przywiązywać i dlatego warto doinstalować te kolekcje bezpośrednio z *ansible-galaxy*, co będzie przedmiotem kolejnych ćwiczeń.

## 2 Test środowiska

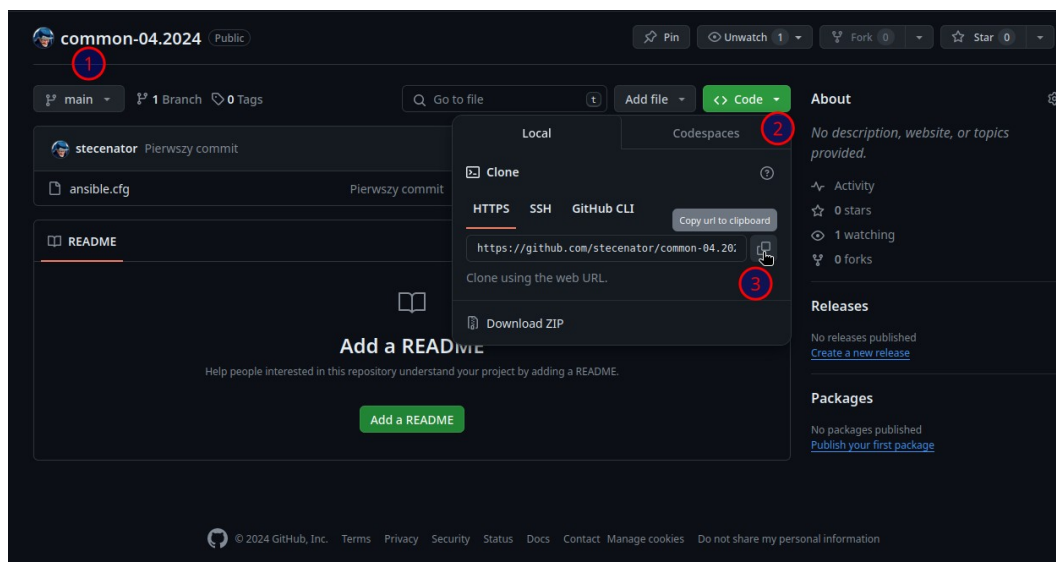
W tej części warsztatu przygotujesz sobie podstawowe środowisko wykonywania playbooków. Celem ćwiczenia jest utworzenie pustego repozytorium Git, oraz założenie szkieletu projektu Ansible.

Użycie Gita nie jest obowiązkowe, ale zalecane, gdyż pozwoli Ci uniknąć uciążliwej (dla niektórych) edycji plików pod Linuxem. Repozytorium Git może być wypchnięte do GitHuba, a ten spięty z Twoim lokalnym repozytorium i ulubionym edytorem.

### 2.1 Konfigurowanie Git do wypychania commitów.

Jeśli planujesz jedynie ściągać playbooksi napisane i zacommitowane do GitHuba z innej stacji.

1. Utwórz repozytorium w serwisie <https://GitHub.com>.
2. Skopiuj URL Twojego repo.



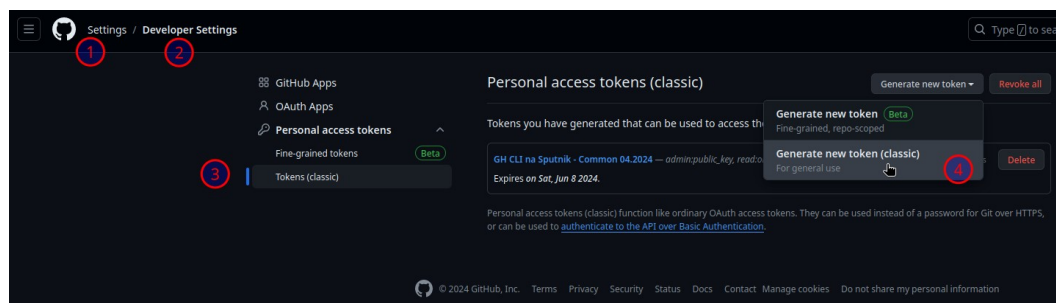
3. Sklonuj repozytorium na maszynie ćwiczeniową.

**Uwaga:** Jeśli nie chcesz klonować przykładowego repozytorium, upewnij się, że w komendzie `git clone` używasz **swojego** URL ;-)

```
marcinek@ender:~/prog/ansible$ git clone https://github.com/stecinator/common-04.2024.git
Cloning into 'common-04.2024'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
marcinek@ender:~/prog/ansible$ ls
common-04.2024
```

Jeśli planujesz wysyłać commity z maszyny ćwiczeniowej do Twojego repozytorium, musisz wykonać następujące, dodatkowe kroki:

1. Wygeneruj w serwisie GitHub token uwierzytelniający (classic). W tym celu przejdź do ustawień swojego profilu (1) *Settings* → (2) *Developer Settings* → (3) *Personal access tokens* → *Tokens (classic)*. Następnie wybierz (4) *Generate new token (classic)*.



2. Nazwij token, wybierz jego czas życia oraz uprawnienia (minimalne uprawnienia to: *Repo*, *Workflow* i *Read:org*):

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

### New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

GH CLI - Ender

What's this token for?

**Expiration \***

30 days The token will expire on Sun, May 12 2024

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows

Po wybraniu uprawnień, naciśnij *Generate token*.

3. **Uwaga:** Po wygenerowaniu tokena, będzie on widoczny na ekranie. To jest **JEDYNA** szansa na skopiowanie go i wklejenie do aplikacji. Dlatego zachowaj go gdzieś w notatniku na boku.

ed in other scopes. Only the minimum set of necessary scopes has been saved.

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

### Personal access tokens (classic)

Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_...	Delete
GH CLI na Sputnik - Common 04.2024 — admin:public_key, read:org, repo	Last used within the last 2 months Delete
Expires on Sat, Jun 8 2024.	

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

4. Zaloguj się do stacji ćwiczeniowej i skonfiguruj uwierzytelnianie klienta CLI GitHub:

---

```
marcinek@ender:~/prog/ansible$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: *****
- gh config set -h github.com git_protocol https
✓ Configured git protocol
! Authentication credentials saved in plain text
✓ Logged in as stecenator
```

---

5. Przedstaw się lokalnemu Gitowi. Będzie używał tych informacji przy wypychaniu commitów do GitHuba:

---

```
marcinek@ender:~/prog/ansible$ git config --global user.email "stecenator@pornhub.com"
marcinek@ender:~/prog/ansible$ git config --global user.name "Marcin Steć"
```

---

6. Upewnij się, że repozytorium jest aktualne. Poniższy przykład pokazuje ściąganie zmian zacommitowanych z innej stacji:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ git pull
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 13 (delta 0), reused 13 (delta 0), pack-reused 0
Unpacking objects: 100% (13/13), 3.00 KiB | 439.00 KiB/s, done.
From https://github.com/stecenator/common-04.2024
   bf802ff..a9e6cc8  main       -> origin/main
Updating bf802ff..a9e6cc8
Fast-forward
 ansible-hosts                | 1 +
 ansible_host-prepare.yaml    | 27 ++++++
 fs_info.yaml                 | 24 ++++++
 group_vars/all.yml           | 9 ++++++
 group_vars/secrets.enc       | 11 ++++++
 hmc_info.yaml                | 18 ++++++
 host_vars/ender.iic.yaml     | 62 ++++++
 ++
 hosts                        | 14 ++++++
 vios_info.yaml               | 14 ++++++
 9 files changed, 180 insertions(+)
 create mode 100644 ansible-hosts
 create mode 100644 ansible_host-prepare.yaml
 create mode 100644 fs_info.yaml
 create mode 100644 group_vars/all.yml
 create mode 100644 group_vars/secrets.enc
 create mode 100644 hmc_info.yaml
 create mode 100644 host_vars/ender.iic.yaml
 create mode 100644 hosts
 create mode 100644 vios_info.yaml
```

---

7. Spróbuj coś dodać, np. plik `.gitignore` o poniższej treści:



---

```

marcinek@ender:~/prog/ansible/common-04.2024$ cat .gitignore
# Compiled source #
#####
*.com
*.class
*.dll
*.exe
*.o
*.so

# Packages #
#####
# it's better to unpack these files and commit the raw source
# git has its own built in compression methods
*.7z
*.dmg
*.gz
*.iso
*.jar
*.rar
*.tar
*.zip

# Logs and databases #
#####
*.log
*.sql
*.sqlite

# OS generated files #
#####
.DS_Store
.DS_Store?
.*
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db

```

---

## 8. Git wykryje, że nowy plik nie jest śledzony. Dodaj go do repozytorium:

---

```

marcinek@ender:~/prog/ansible/common-04.2024$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore

nothing added to commit but untracked files present (use "git add" to track)

marcinek@ender:~/prog/ansible/common-04.2024$ git add -A
marcinek@ender:~/prog/ansible/common-04.2024$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)

```

---

---

new file: .gitignore

---

9. Zacommituj zmianę z odpowiednim komentarzem:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ git commit -m "Dodany generyczny .gitignore"
[main fcb9832] Dodany generyczny .gitignore
1 file changed, 37 insertions(+)
create mode 100644 .gitignore
```

---

10. Wyślij zmiany do zdalnej gałęzi:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 541 bytes | 541.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/stecenator/common-04.2024.git
a9e6cc8..fcb9832 main -> main
```

---

11. Jeśli *git push* się powiódł, to możesz pracować nad kodem także z tej maszyny. Jeśli nie, to prawdopodobnie coś poszło nie tak z generowaniem tokena

**Uwaga:** Od teraz możesz pracować na dwóch stanowiskach: Twoim ulubionym środowisku, gdzie repozytorium zostało utworzone i na maszynie ćwiczeniowej. To może być źródłem problemów, gdy dokonasz zmian w obu lokalnych kopiach repozytoriów. Komenda *git push* poinformuje Cię o tym w przykry sposób, dlatego warto pracować z jedną kopią repozytorium, a po *git commit/git push* robić tylko *git pull* na stacji ćwiczeniowej.

## 2.2 Konfiguracja projektu Ansible

Struktura katalogów projektu Ansible jest dobrze przemyślana i określona. W tej części ćwiczenia przygotujesz podstawowy układ katalogów. Więcej informacji o konfigurowaniu „pustego” projektu znajdziesz w dokumentacji Ansible:

[https://docs.ansible.com/ansible/latest/tips\\_tricks/sample\\_setup.html](https://docs.ansible.com/ansible/latest/tips_tricks/sample_setup.html)

1. Przygotuj podstawowe katalogi. Potem, w razie potrzeby będziesz dodawać inne.

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ mkdir group_vars host_vars
marcinek@ender:~/prog/ansible/common-04.2024$ tree -d
.
├── group_vars
└── host_vars
```

---

2. W katalogu głównym Twojego projektu utwórz plik *ansible.cfg* o następującej treści:

---

```
[defaults]
inventory = hosts
host_key_checking = False
```

---

Pełna dokumentacja dotycząca możliwej zawartości pliku *ansible.cfg* oraz kolejności parsowania poszczególnych jego kopii znajduje się tutaj:

[https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_configuration.html](https://docs.ansible.com/ansible/latest/installation_guide/intro_configuration.html)

3. Utwórz podstawowy plik *hosts* w głównym katalogu projektu. Zmienna *inventory* pliku *ansible.cfg* wskazuje właśnie na ten plik. W trakcie ćwiczeń, będziesz korzystać z kilku plików *inventory*.

---

```
localhost
s824-6013-vios2.iic
s824-6013-vios1.iic

[ctrl]
localhost
s824-6013-vios2.iic
s824-6013-vios1.iic

[viosy]
```

---

#### 4. Przetestuj połączenie modulem *ping*:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible ctrl -m ping
localhost | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Warning: Permanently added 'localhost'
(ED25519) to the list of known hosts.\r\nmarcinek@localhost: Permission denied
(publickey,gssapi-keyex,gssapi-with-mic,password).",
  "unreachable": true
}
```

---

Jak widać zwarzył się. Ansible nie wiedział na kogo się łączyć i jak. Próbował zatem na bieżącym uzyszkodniku. Dodaj jego klucz ssh i spróbuj ponownie:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ssh-copy-id localhost
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/marcinek/.ssh/id_ecdsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that
are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is
to install the new keys
marcinek@localhost's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'localhost'"
and check to make sure that only the key(s) you wanted were added.

marcinek@ender:~/prog/ansible/common-04.2024$ ansible ctrl -m ping
localhost | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

---

5. Utwórz plik *secrets.enc*, który będzie przechowywać kluczowe ze względu bezpieczeństwa zmienne w zaszyfrowanej formie. Program *ansible-vault* najpierw spyta dwukrotnie o hasło do szyfrowania (daj coś prostego, np. *ibm12345*, bo będziesz to hasło wpisywać przy każdym playbooku korzystającym z tych zmiennych).

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-vault create secrets.enc
New Vault password:
Confirm New Vault password:
```

---

6. Uruchomi się edytor wskazany zmienną *\$EDITOR*. Domyślny dla współczesnych Fedor to *nano*. Wpisz tam następującą treść:

---

```
---
hmc_user: "common"
hmc_pass: "C0mm0n@Gdyn1@"
common_user_pass: "C0mm0n@Gdyn1@"
```

---

W dalszych częściach ćwiczeń, będziesz dopisywać tam kolejne elementy przy pomocy komendy *ansible-vault edit secrets.enc*.

7. W katalogu *group\_vars* utwórz plik *all.yml*:

```
---
fs_cluster: "v5ke.iic"
fs_log: "/tmp/fs.log"
hmc_user: marciniek
hmc: hmc10.iic
```

8. Zacomituj i wypchnij zmiany w repozytorium. Nazwij swój commit np. „Wstępny konfig”:

```
marciniek@ender:~/prog/ansible/common-04.2024$ git add -A
marciniek@ender:~/prog/ansible/common-04.2024$ git commit -m "Wstępny konfig"
[main 552c768] HMC user w secrets
2 files changed, 10 insertions(+), 11 deletions(-)
delete mode 100644 group_vars/secrets.enc
create mode 100644 secrets.enc
marciniek@ender:~/prog/ansible/common-04.2024$ git push
Wymienianie obiektów: 10, gotowe.
Zliczanie obiektów: 100% (10/10), gotowe.
Kompresja delt z użyciem do 8 wątków
Kompresowanie obiektów: 100% (6/6), gotowe.
Zapisywanie obiektów: 100% (7/7), 996 bajtów | 498.00 KiB/s, gotowe.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:stecenator/common-04.2024.git
cddbef4..552c768 main -> main
```

Pozostałe komponenty, w tym konfiguracja i testowanie kolekcji IBM będzie przeprowadzona w dalszych częściach warsztatu

## 3 Moduł `ibm.storage_virtualize`

### 3.1 Instalacja i test połączenia

Celem tego ćwiczenia jest instalacja kolekcji `ibm.storage_virtualize` oraz uruchomienie prostego playbooksa, celem przetestowania połączenia z macierzą.

1. Linux (Fedora) ma zainstalowaną aktualną wersję kolekcji `ibm.storage_virtualize`. Na innych dystrybucjach może być konieczność instalacji. Użyj instrukcji z dokumentacji: [https://galaxy.ansible.com/ui/repo/published/ibm/storage\\_virtualize/](https://galaxy.ansible.com/ui/repo/published/ibm/storage_virtualize/)

### 3.2 Test kolekcji `ibm.storage_virtualize`

W tym ćwiczeniu utworzysz prosty playbook, który połączy się z macierzą IBM Flash System i wyświetli kilka informacji o niej.

1. Dopisz zmienne `fs_user/fs_pass` do pliku `secrets.enc`. Będą one potem używane do uwierzytelnienia przy każdym połączeniu z macierzą.

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-vault edit secrets.enc
Vault password:
```

---

Zostanie uruchomiony \$EDITOR. Dodaj zmienne wspomniane wyżej zmienne:

---

```
---
hmc_user: "common"
hmc_pass: "C0mm0n@Gdyn1@"
common_user_pass: "C0mm0n@Gdyn1@"
fs_user: "common"
fs_pass: "C0mm0n@Gdyn1@"
```

---

2. Upewnij się, że w pliku `group_vars/all.yml` jest ustawiona zmienna `fs_cluster`, której będziesz używać w dalszej części ćwiczenia do wskazywania macierzy.

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ cat group_vars/all.yml
---
```

---

---

```
fs_cluster: "fs5200-2.iic"
fs_log: "/tmp/fs.log"
```

---

### 3. Przygotuj następujący playbook (*fs\_info.yaml*):

```
---
# Przykładowy playbook zbierający informacje o macierzy IBM FS

- hosts: localhost      # Za wykonanie modułów ibm.storage_virtualize odpowiada stacja
  kontrolna
  gather_facts: false # bo nie obchodzi mnie co wiem o localhost

  tasks:
    - name: "Uwierzytelnienie na jako {{ fs_user }} na {{ fs_cluster }}"
      register: auth_info # auth_info.token pozwala potem nie użwać user i hasła
      ibm.storage_virtualize.ibm_svc_auth:
        clustername: "{{ fs_cluster }}"
        log_path: "{{ fs_log }}"
        username: "{{ fs_user }}"
        password: "{{ fs_pass }}"
    - name: Zbieranie informacji o macierzy
      register: svc_inf
      ibm.storage_virtualize.ibm_svc_info:
        clustername: "{{ fs_cluster }}"
        log_path: "{{ fs_log }}"
        token: "{{ auth_info.token }}"
        gather_subset: system # Bo domyślne `all` to trochę za dużo
    - name: Wyświetlenie informacji o macierzy
      debug:
        var: svc_inf
```

---

**Uwaga:** Warto zwrócić uwagę na konstrukcję tego playbooka. Większość modułów tej kolekcji używa REST API, dlatego możliwe jest jednokrotne uwierzytelnienie i zarejestrowanie zmiennej *auth\_info*.

Drugą ciekawostką jest wskazany zmienną *fs\_log* plik loga, gdzie można znaleźć informacje debugowe.

### 4. Uruchom playbook *fs\_info.yaml*:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-playbook fs_info.yaml -e @secrets.enc --ask-vault-
pass
Vault password:

PLAY [localhost]
*****

TASK [Uwierzytelnienie na jako common na fs5200-2.iic]
*****
ok: [localhost]

TASK [Zbieranie informacji o macierzy]
*****
ok: [localhost]
```

---

---

TASK [Wyświetlenie informacji o macierzy]

\*\*\*\*\*

```
ok: [localhost] => {
  "svc_inf": {
    "Array": [],
    "CallHome": [],
    "CloudAccount": [],
    "CloudAccountUsage": [],
    "CloudBackup": [],
    "CloudBackupGeneration": [],
    "CloudImportCandidate": [],
    "DnsServer": [],
    [ ... ]
    "Volumegroupreplication": [],
    "changed": false,
    "failed": false,
    "iSCSIPort": []
  }
}
```

PLAY RECAP

\*\*\*\*\*

```
localhost          : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

---

5. Git add, git commit, git push.

### 3.3 Przygotowanie dysku

W tym ćwiczeniu zapoznasz się z mechanizmami tworzenia LUNów przy pomocy modułu *ibm.storage\_virtualize*. Dowiesz się także jak odtworzyć migawkę i zamapować LUN do hosta.

#### Założenia:

- Dobrą praktyką jest unikanie hard-kodowania wartości w playbooku. Dlatego większość parametrów tasków jest wyprowadzona do zmiennych.
- Każdy użytkownik dostał unikalny login i należy go użyć jako nazwy tworzonego LPARu. Do tego celu służy zmienna *my\_lpar\_name*.
- Każdy uczestnik warsztatów tworzy jeden LUN o nazwie takiej jak login z sufiksem *\_os*. Czyli np. dla użytkownika *common01*, vdisk na macierzy będzie się nazywać *common01\_os*.
- Nazwa LUNu/vdisku jest określana zmienną *my\_vdisk* o wartości *{{ my\_lpar\_name }}\_os*
- Rozmiar jest określony zmienną *my\_vdisk\_size*. Ten parametr prawdopodobnie się nie przyda, bo tworzony LUN będzie klonem lunu *common\_master\_os*, na który wskazuje zmienna *src\_vdisk*.
- Utworzony LUN jest mapowany na *hostcluster* określony zmienną *vios\_cluster*.



1. Zaktualizuj plik `group_vars/all.yaml` tak żeby były w nim podkreślone na poniższym wydruku zmienne.

**Uwaga:** upewnij się, że zmiennej `my_lpar_name` nadano nazwę taką jak Twój login! Od jej wartości sporo zależy :-). W poniższych przykładach użyto loginu `common30`.

```
---
# Sprawy ogólne
my_lpar_name: "common30" # Hostname i nazwa LPARu. Zamień 00 na numer Twojego
użytkownika

# Sprawy storydżowe
fs_cluster: "fs5200-2.iic"
fs_log: "/tmp/fs-{{ my_lpar_name }}.log"
my_vdisk: "{{ my_lpar_name }}_os" # common30_os
my_vdisk_size: 64 # 64 czego określa marametr unit. Zakładamy że GiB
src_vdisk: "common_master_os" # Wcześniej przygotowany obraz Fedory
vios_cluster: "S824-6013" # hoscluster VIOSów zdefiniowany na macierzy
fs_pool: "Pool_SiteA" # mdiskgrupa

# Flaśer-paler
hmc: "hmc10.iic"
```

2. Utwórz playbooka `fs_create_os_lun.yaml`:

```
---
# Przykładowy playbook zbierający informacje o macierzy IBM FS

- hosts: localhost # Za wykonanie modułów ibm.storage_virtualize odpowiada stacja
kontrolna
  gather_facts: false # bo nie obchodzi mnie co wiem o localhost

  tasks:
    - name: "Uwierzytelnienie na jako {{ fs_user }} na {{ fs_cluster }}"
      register: auth_info # auth_info.token pozwala potem nie uważać user i hasła
      ibm.storage_virtualize.ibm_svc_auth:
        clustername: "{{ fs_cluster }}"
        log_path: "{{ fs_log }}"
        username: "{{ fs_user }}"
        password: "{{ fs_pass }}"
      tags:
        - always # Uwierzytelnić zawsze się trzeba

    - name: Tworzenie sklonowanego LUNu pod OSa - źródłem jest LUN common_master_os
      ibm.storage_virtualize.ibm_svc_manage_volume:
        clustername: "{{ fs_cluster }}"
        log_path: "{{ fs_log }}"
        token: "{{ auth_info.token }}"
        name: "{{ my_vdisk }}"
        state: "present"
        pool: "{{ fs_pool }}"
        # size: "{{ my_vdisk_size }}" # to jest do stawiania LUNu od 0
        # unit: "gb"
        type: "thinclone"
        fromsourcevolume: "{{ src_vdisk }}"
        iogrp: "io_grp0"
        novolumegroup: True
```

```
tags:
  - "MKVDISK"
```

- Uruchom playbooka `fs_create_os_lun.yaml` z opcją `--check`, żeby sprawdzić czy wszystko jest poprawne:

```
common30@ender:~/common-04.2024$ ansible-playbook fs_create_os_lun.yaml -e @secrets.enc --check --ask-
vault-pass -t MKVDISK
Vault password:

PLAY [localhost] *****

TASK [Uwierzytelnienie na jako common na fs5200-2.iic] *****
ok: [localhost]

TASK [Tworzenie sklonowanego LUNu pod 0Sa - źródłem jest LUN common_master_os] *****
ok: [localhost]

PLAY RECAP *****
localhost                : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

- Powtórz komendę z poprzedniego punktu, już bez opcji `-check`. Zaloguj się na macierz i komendą `lsdisk` sprawdź czy Twój LUN został utworzony:

```
common30@ender:~/common-04.2024$ ssh common@fs5200-2
(common@fs5200-2) Password:
IBM_FlashSystem:FS5200_2:common>lsdisk -delim : | grep common
7:common_master_os:0:io_grp0:online:0:Pool_SiteA:64.00GB:striped::::60050768128080F71800000
00000000D:0:1:empty:0:no:0:0:Pool_SiteA::no:no:7:common_master_os::::scsi:no:1::no:0:no
10:common30_os:0:io_grp0:online:0:Pool_SiteA:64.00GB:striped::::60050768128080F718000000000
00011:0:1:not_empty:0:no:0:0:Pool_SiteA::yes:no:10:common30_os::::scsi:no:0:thinclone::no:0
:no
```

- Do playbooka `fs_create_os_lun.yaml` dodaj następujący task:

```
- name: Mapowanie LUNu na klaster VIOSów
  ibm.storage_virtualize.ibm_svc_vol_map:
    clustername: "{{ fs_cluster }}"
    log_path: "{{ fs_log }}"
    token: "{{ auth_info.token }}"
    volname: "{{ my_vdisk }}"
    state: "present"
    hostcluster: "{{ vios_cluster }}"
  tags:
    - "VDISKMAP"
```

- Uruchom playbooka tylko z tagiem `VDISKMAP`:

```
common30@ender:~/common-04.2024$ ansible-playbook fs_create_os_lun.yaml -e @secrets.enc --ask-vault-pass -t VDISKMAP
Vault password:

PLAY [localhost] *****

TASK [Uwierzytelnienie na jako common na fs5200-2.iic] *****
ok: [localhost]

TASK [Mapowanie LUNu na klaster VIOSów] *****
changed: [localhost]
```

---

```
PLAY RECAP *****
localhost                : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

---

7. Wykonaj konfigurację magistra na VIOSach i sprawdź komendą `lsmPIO -q` czy Twój LUN jest poprawnie podpięty:

```
root@vios1:/ # lspv
hdisk0          00f9c7fd5e6d0a94          rootvg          active
hdisk1          00f9c7fd6a758a5a          None
hdisk2          none                    None
hdisk3          none                    None
hdisk4          none                    None
root@vios1:/ # cfgmgr
root@vios1:/ # lsmPIO -q
```

Device	Vendor Id	Product Id	Size	Volume Name
hdisk0	IBM	ST9300653SS	279.40GiB	-
hdisk1	IBM	ST9300653SS	279.40GiB	-
hdisk2	IBM	2145	10.00GiB	S824-6013-dummy-hdisk2
hdisk3	IBM	2145	10.00GiB	S824-6013-dummy-hdisk3
hdisk4	IBM	2145	64.00GiB	common_master_os
<b>hdisk5</b>	<b>IBM</b>	<b>2145</b>	<b>64.00GiB</b>	<b>common30_os</b>

---

**Uwaga:** Powtórz tę operację także na drugim VIOSie.

8. Znaną nazwę dysku wpisz do `group_vars/all.yml` na zmienną `my_hdisk`:

```
---
# Sprawy ogólne
my_lpar_name: "common30"          # Hostname i nazwa LPARu. Zamień 00 na numer Twojego
                                użytkownika
my_hdisk: "hdisk5"                # Nazwa mojego LUNu tak jak go widzą VIOSy. Uwaga, w
                                różnych VIOSach mogą być różne nazwy!
my_power: "S824-6013"            # system_name zarządzanego przez HMC pudła

# Sprawy storydzowe
fs_cluster: "fs5200-2.iic"
fs_log: "/tmp/fs-{{ my_lpar_name }}.log"
my_vdisk: "{{ my_lpar_name }}_os" # common00_os
src_vdisk: "common_master_os"    # Wcześniej przygotowany obraz Fedory wyciągany przez Adama
                                Słodowego spod biurka
my_vdisk_size: 64                 # 64 czego określa parametr unit. Zakładamy że GiB
vios_cluster: "S824-6013"        # hoscluster VIOSów zdefiniowany na macierzy
fs_pool: "Pool_SiteA"            # mdiskgrupa

# Flażer-paler
hmc: "hmc10.iic"
```

---

9. Git add, git commit, git push ;-)

## 4 Tworzenie LPARu – `ibm.power_hmc`

W tym ćwiczeniu skupisz się na stworzeniu partycji logicznej. Dzięki temu zapoznasz się z kolekcją `ibm.power_hmc`.

**Uwaga:** Moduły tej kolekcji używają połączenia SSH z hostem HMC. HMC jest zamkniętym systemem, na którym nie można wykonywać wstrzykniętego kodu Pythona, dlatego jedynym hostem, który w rzeczywistości będzie komunikować się z HMC jest stacja kontrolna Ansible. Stąd ustawienie zmiennej `hosts: localhost` przy taskach odwołujących się do HMC.

### 4.1 Instalacja kolekcji `ibm.power_hmc`

Źródłem wszelkiej mądrości w temacie tej kolekcji jest jej dokumentacja:

<https://ibm.github.io/ansible-power-hmc/index.html>.

1. Użyj `ansible-galaxy` do zainstalowania kolekcji `ibm.power_hmc`. Najpierw sprawdź, czy jakaś już nie jest zainstalowana.

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-galaxy collection list | grep "^ibm"
ibm.power_vios                1.3.0
ibm.qradar                    2.1.0
ibm.spectrum_virtualize       2.0.0
ibm.storage_virtualize        2.3.1
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-galaxy collection install
ibm.power_hmc
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/
artifacts/ibm-power_hmc-1.8.2.tar.gz to /home/marcinek/.ansible/tmp/ansible-local-
4481x84nv44j/tmp3bl71ktv/ibm-power_hmc-1.8.2-0gd5u2id
Installing 'ibm.power_hmc:1.8.2' to
'/home/marcinek/.ansible/collections/ansible_collections/ibm/power_hmc'
ibm.power_hmc:1.8.2 was installed successfully
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-galaxy collection list | grep "^ibm"
ibm.power_hmc                1.8.2
ibm.power_vios                1.3.0
ibm.qradar                    2.1.0
ibm.spectrum_virtualize       2.0.0
ibm.storage_virtualize        2.3.1
```

## 4.2 Test kolekcji *ibm.power\_hmc*

W tym ćwiczeniu przetestujesz połączenie z konsolą HMC i zbierzesz fakty na temat konsoli oraz zarządzanego systemu IBM Power.

Będziesz potrzebować użytkownika i hasła na HMC. W ramach ćwiczenia użytkownik nie będzie wymieniony kluczem ssh z HMC, więc Ansible będzie musiał przekazać jego hasło. Dobrą praktyką jest przechowywanie informacji uwierzytelniających w zaszyfrowanych plikach

1. Sprawdź czy w pliku *secrets.enc* są kredki do konta w HMC i razie potrzeby komendą *ansible-vault edit secrets.enc* dopisz je.

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-vault view secrets.enc
Vault password:
---
hmc_user: "common"
hmc_pass: "C0mm0n@Gdyn1@"
common_user_pass: "C0mm0n@Gdyn1@"
```

---

2. Utwórz playbooka *hmc\_info.yaml* o następującej treści:

---

```
---
# Przykładowy playbook zbierający informacje o Power HMC

- hosts: localhost      # Za wykonanie modułów ibm.power_hmc odpowiada stacja kontrolna
  gather_facts: false    # bo nie obchodzi mnie co wiem o localhost

  tasks:
    - name: Odpytywanie HMC
      register: build_info
      ibm.power_hmc.hmc_update_upgrade:
        state: facts
        hmc_host: '{{ hmc }}'
        hmc_auth:
          username: '{{ hmc_user }}'
          password: '{{ hmc_pass }}'
    - name: Wyświetlenie konsoli
      debug:
        var: build_info
```

---

3. Uruchom playbook *hmc\_info.yaml*.

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-playbook hmc_info.yaml -e @secrets.enc
--ask-vault-pass
Vault password:
```

---

---

```
PLAY [localhost]
```

```
*****
```

```
TASK [Odpytywanie HMC]
```

```
*****
```

```
ok: [localhost]
```

```
TASK [Wyświetlenie konsoli]
```

```
*****
```

```
ok: [localhost] => {
  "build_info": {
    "ansible_facts": {
      "discovered_interpreter_python": "/usr/bin/python3"
    },
    "build_info": {
      "BASEVERSION": "V10R2",
      "HMCBUILDLEVEL": "2211152246",
      "RELEASE": "2",
      "SERVICEPACK": "1030",
      "VERSION": "10"
    },
    "changed": false,
    "failed": false
  }
}
```

```
PLAY RECAP
```

```
*****
```

```
***
```

```
localhost      : ok=2    changed=0    unreachable=0    failed=0    skipped=0
rescued=0      ignored=0
```

---

**Uwaga:** Ansible może marudzić, że klucz ssh do HMC nie jest znany. Wystarczy połączyć się z HMC po ssh i zaakceptować tenże.

4. Pamiętaj! *git add*, *git commit*, *git push*.

## 4.3 Tworzenie i aktywacja LPARu

W tym ćwiczeniu, w końcu wystrzelisz swój LPAR.

### Założenia:

- Nazwa Twojego LPARu to *username* przydzielony na początku warsztatów. Upewnij się, że zmienna *my\_lpar\_name* jest właśnie tak ustawiona.
- Nie mamy czasu na wyciąganie z *ansible\_facts* nazwy Twojego dysku do zamapowania podczas tworzenia LPARu, dlatego musisz go znać i wpisać „na twardo” do *group\_vars/all.yaml* na zmiennej *my\_hdisk*. Ten krok był w ćwiczeniu 3.3.
- Na twardo też wpisujemy parametry takie jak pamięć czy procesor.
- Ćwiczenie będziemy wykonywać po kolei, żeby uniknąć ewentualnych problemów z równoczesnym wykonywaniem operacji na HMC lub VIOSach.

- Upewnij się, że zmienne *my\_lpar\_name* i *my\_hdisk* są poprawnie ustawione w *group\_vars/all.yaml*. W razie potrzeby, Zaloguj się na S824-6013-vios2 (albo 1) na użytkownika *ansible* i wyszukaj nazwę swojego dysku utworzonego w ćwiczeniu 3.3. Poniższy przykład wyszukuje dysk użytkownika *common30*:

```
root@vios2:/ # lsmpio -q | grep common30
hdisk5          IBM          2145          64.00GiB common30_os
```

- Utwórz playbooka *hmc\_create\_lpar.yaml*.

**Uwaga:** Zwróć uwagę na jeden szczegół: Dysk `{{ my_hdisk }}` jest mapowany tylko z jednego VIOSa. Moduł *ibm.power\_hmc.powervm\_lpar\_instance* jest na tyle uprzejmy, że zrobi to mapowanie na obu VIOSach :-)

```
---
# Przykładowy playbook zbierający informacje o Power HMC

- hosts: localhost      # Za wykonanie modułów ibm.storage_virtualize odpowiada stacja
  kontrolna
  gather_facts: false # bo nie obchodzi mnie co wiem o localhost

  tasks:
    - name: Tworzenie LPARu
      ibm.power_hmc.powervm_lpar_instance:
        hmc_host: '{{ hmc }}'
        hmc_auth:
          username: '{{ hmc_user }}'
```

```

    password: '{{ hmc_pass }}'
    system_name: "{{ my_power }}"
    vm_name: "{{ my_lpar_name }}"
    min_proc_unit: 0.2
    proc_unit: 0.2
    max_proc_unit: 1.0
    proc: 2          # Liczba procesorów wirtualnych, bo jest ustawione proc_unit.
    os_type: "aix_linux"
    volume_config:
      # - vios_name: "vios1"      # jak to widać na HMC, a nie hostname viosa
      #   volume_name: "{{ my_hdisk }}"
      - vios_name: "vios2"
        volume_name: "{{ my_hdisk }}"
    virt_network_config:
      - network_name: "VLAN1-ETHERNET0"
    state: present

```

3. Uruchom playbooka. Poniższy przykład pokazuje playbook uruchomiony z opcją -vvv co podnosi poziom gadatliwości Ansibla, stąd nieco przydługi wydruk. Jest to bardzo pomocne przy debugowaniu:

```

marcinek@ender:~/prog/ansible/common-04.2024$ ansible-playbook hmc_create_lpar.yaml -e @secrets.enc --ask-vault-pass -vvv
ansible-playbook [core 2.16.5]
  config file = /home/marcinek/prog/ansible/common-04.2024/ansible.cfg
  configured module search path = ['/home/marcinek/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.12/site-packages/ansible
  ansible collection location = /home/marcinek/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible-playbook
  python version = 3.12.2 (main, Feb 21 2024, 00:00:00) [GCC 13.2.1 20231205 (Red Hat 13.2.1-6)] (/usr/bin/python3)
  jinja version = 3.1.3
  libyaml = True
Using /home/marcinek/prog/ansible/common-04.2024/ansible.cfg as config file
Vault password:
host_list declined parsing /home/marcinek/prog/ansible/common-04.2024/hosts as it did not pass its verify_file() method
script declined parsing /home/marcinek/prog/ansible/common-04.2024/hosts as it did not pass its verify_file() method
auto declined parsing /home/marcinek/prog/ansible/common-04.2024/hosts as it did not pass its verify_file() method
Parsed /home/marcinek/prog/ansible/common-04.2024/hosts inventory source with ini plugin
Skipping callback 'default', as we already have a stdout callback.
Skipping callback 'minimal', as we already have a stdout callback.
Skipping callback 'oneline', as we already have a stdout callback.

PLAYBOOK: hmc_create_lpar.yaml *****
1 plays in hmc_create_lpar.yaml

PLAY [localhost] *****

TASK [Tworzenie LPARu] *****
task path: /home/marcinek/prog/ansible/common-04.2024/hmc_create_lpar.yaml:8

[ ... ]

changed: [localhost] => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": true,
  "invocation": {
    "module_args": {
      "action": null,
      "advanced_info": null,
      "all_resources": null,
      "delete_vdisks": null,
      "hmc_auth": {
        "password": "VALUE_SPECIFIED_IN_NO_LOG_PARAMETER",

```



```

        "username": "VALUE_SPECIFIED_IN_NO_LOG_PARAMETER"
    },
    "hmc_host": "hmc10.iic",
    "iPLSource": null,
    "install_settings": null,
    "keylock": null,
    "max_mem": null,
    "max_proc": null,
    "max_proc_unit": 1.0,
    "max_virtual_slots": null,
    "mem": null,
    "min_mem": null,
    "min_proc": null,
    "min_proc_unit": 0.2,
    "npiv_config": null,
    "os_type": "aix_linux",
    "physical_io": null,
    "proc": 2,
    "proc_compatibility_mode": null,
    "proc_mode": null,
    "proc_unit": 0.2,
    "prof_name": null,
    "restart_option": null,
    "retain_vios_cfg": null,
    "shared_proc_pool": null,
    "shutdown_option": null,
    "state": "present",
    "system_name": "S824-6013",
    "virt_network_config": [
        {
            "network_name": "VLAN1-ETHERNET0",
            "slot_number": null
        }
    ]
}

[ ... ]

"partition_info": {
    "AllocatedVirtualProcessors": 2,
    "AssociatedManagedSystem": "S824-6013",
    "CurrentMemory": 2048,
    "CurrentProcessingUnits": 0.2,
    "CurrentProcessors": 2,
    "CurrentUncappedWeight": 128,
    "Description": null,
    "HasDedicatedProcessors": "false",
    "HasPhysicalIO": "false",
    "IsVirtualServiceAttentionLEDOn": "false",
    "LastActivatedProfile": "default_profile",
    "MemoryMode": "Dedicated",
    "MigrationState": "Not_Migrating",
    "OperatingSystemType": "AIX/Linux",
    "OperatingSystemVersion": "Unknown",
    "PartitionID": 3,
    "PartitionName": "*****_master",
    "PartitionState": "not activated",
    "PartitionType": "AIX/Linux",
    "PowerManagementMode": null,
    "ProgressState": null,
    "RMCState": "inactive",
    "ReferenceCode": "",
    "RemoteRestartState": "Invalid",
    "ResourceMonitoringIPAddress": null,
    "SharingMode": "uncapped",
    "SystemName": "S824-6013"
}
}

PLAY RECAP *****
localhost          : ok=1    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    i
gnored=0

```

4. Pora na aktywację LPARu. Zaloguj przez SSH się do HMC w dwóch oknach. W pierwszym otworzysz konsolę do Twojego LPARu, w drugim, wydasz komendę do uruchomienia go.
5. W pierwszej sesji wydaj komendę `vtmenu` i wybierz swój LPAR:

```

-----
Partitions On Managed System: S824-6013
05/400 Partitions not listed
-----
1)    common_master          Not Activated
2)    vios1                  Running
3)    vios2                  Running
4)    wsaix01                Running
5)    wsaix02                Running

```

```
Enter Number of Running Partition (q to quit): 1
```

Zostanie otwarta konsola szeregową do Twojej partycji. To się da zrobić nawet do wyłączonej partycji.

#### 6. W drugiej sesji znajdź nazwę i ID Twojej partycji:

```

common@5817vhmc-iic:~> lssyscfg -r lpar -m S824-6013 -F lpar_id,name
14,WSA_EDI_Clients
13,EDI_Wroclaw
12,EDI_Gliwice
101,vios2
100,vios1
7,wsas01
2,wsaix02
1,wsaix01
3,common_master

```

#### 7. Aktywuj tę partycję:

```

common@5817vhmc-iic:~> chsysstate -m S824-6013 -r lpar -o on --id 3 -f default_profile -b
sms

```

Jeśli wszystko się powiodło, możesz zamknąć tę sesję i wrócić do tej z konsolą. LPAR powinien być odpalony do SMS.

#### 8. W menu SMS wybierz opcję: *Select boot options*:

```

PowerPC Firmware
Version FW860.B3 (SV860_245)
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.
-----
Main Menu
1.  Select Language
2.  Setup Remote IPL (Initial Program Load)
3.  I/O Device Information
4.  Select Console
5.  Select Boot Options

```

---

-----  
 Navigation Keys:

X = eXit System Management Services  
 -----

Type menu item number and press Enter or select Navigation key:

---

9. W menu *Multiboot* wybierz 1. *Select install/boot device*.
10. W menu *Select Device Type* wybierz 3. *Hard Drive*.
11. W menu *Media Type* wybierz 1. *SCSI*.
12. W menu *Select Adapter* wybierz 3. *List all devices*.
13. W menu *Select device* wybierz jeden z podanych dysków. Są to dwie ścieżki do Twojego LUNu. Nieważne, z której podniesiesz system.

---

```

Select Device
Device  Current  Device
Number  Position  Name
1.      2      Interpartition Logical LAN
          ( loc=U8286.42A.21C7FDV-V4-C2-T1 )
2.      1      SCSI 63 GB Harddisk, part=1 ( )
          ( loc=U8286.42A.21C7FDV-V4-C3-T1-L8100000000000000 )
3.      -      SCSI 63 GB Harddisk, part=1 ( )
          ( loc=U8286.42A.21C7FDV-V4-C4-T1-L8100000000000000 )
  
```

-----  
 Navigation keys:

M = return to Main Menu

ESC key = return to previous screen

X = eXit System Management Services  
 -----

Type menu item number and press Enter or select Navigation key: **2**

---

14. W menu *Select Task* wybierz *Normal Boot*. (w zasadzie możesz też wybrać *Service*, bo Linux ma to w nosie).

---

```
Select Task

SCSI 63 GB Harddisk, part=1 ( )
  ( loc=U8286.42A.21C7FDV-V4-C3-T1-L8100000000000000 )

1.  Information
2.  Normal Mode Boot
3.  Service Mode Boot

-----
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen          X = eXit System Management Services
-----
Type menu item number and press Enter or select Navigation key:2
```

---

15. Tak, jesteśmy pewni, że opuszczamy SMS:

---

```
Are you sure you want to exit System Management Services?
1.  Yes
2.  No

-----
Navigation Keys:
X = eXit System Management Services
-----
Type menu item number and press Enter or select Navigation key:1
```

---

## 16. Jedziemy z Fedorą!

```
+-----+
|*Fedora Linux (6.8.6-300.fc40.ppc64le) 40 (Server Edition) |
| Fedora Linux (6.8.0-0.rc6.49.fc40.ppc64le) 40 (Server Edition Prerelease) |
| Fedora Linux (0-rescue-733ca739dd3b4e97a40a7701dd6a07f6) 40 (Server Editio>|
|
|
|
|
|
|
|
|
|
|
+-----+
```

Use the ^ and v keys to select which entry is highlighted.  
 Press enter to boot the selected OS, 'e' to edit the commands  
 before booting or 'c' for a command-line.  
 The highlighted entry will be executed automatically in 0s.

17. Poczekaj chwilę, aż zgłosi się ekran logowania do systemu i zaloguj się na użytkownika `common/C0mm0n@Gdyn1@`. System podnosi sieć z DHCP, ale wstanie z kulawym hostname, więc warto go zmienić. Potem wyświetl sobie IP Twojego LPARu i zaloguj się na niego ze stacji kontrolnej Ansible przez ssh.

```
Fedora Linux 40 (Server Edition)
Kernel 6.8.6-300.fc40.ppc64le on an ppc64le (hvc0)

Web console: https://common00.iic:9090/ or https://10.10.3.125:9090/

common00 login: common
Password:
[ 95.935819] systemd-journald[862]:
/var/log/journal/733ca739dd3b4e97a40a7701dd6a07f6/user-1000.journal: Journal file uses a
different sequence number ID, rotating.
Last login: Sun Apr 14 22:39:24 from 10.10.13.14
[common@common00 ~]$ sudo hostnamectl set-hostname common30.iic
[common@common00 ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: env2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UNKNOWN group default
qlen 1000
    link/ether 3e:db:a9:53:5a:02 brd ff:ff:ff:ff:ff:ff
    inet 10.10.3.125/16 brd 10.10.255.255 scope global dynamic noprefixroute env2
        valid_lft 7077sec preferred_lft 7077sec
    inet6 fe80::3cdb:a9ff:fe53:5a02/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

18. Git add, git commit, git push! Miłej zabawy z Twoją nową Fedorą na IBM Power ;-)

## 5 Majsterkowanie w VIOS

W tym ćwiczeniu dowiesz się jak zainstalować i przetestować kolekcję *ibm.power\_vios*. W chwili tworzenia tego warsztatu, kolekcja *ibm.power\_vios* nie umożliwia modyfikowania mapowań, dlatego to ćwiczenie jest opcjonalne.

### 5.1 Instalacja kolekcji *ibm.power\_vios*

Pełny podręcznik instalacji kolekcji *ibm.power\_vios* znajduje się tu:

<https://ibm.github.io/ansible-power-vios/installation.html>. Poniżej znajduje się krótka, i wystarczająca lista kroków.

1. Zaloguj się do stacji ćwiczeniowej na swoje konto i przejdź do katalogu z Twoim projektem Ansible.
2. Sprawdź czy kolekcja nie jest już przypadkiem zainstalowana:

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-galaxy collection list | grep -i \
"^ibm"
ibm.qradar                2.1.0
ibm.spectrum_virtualize   2.0.0
ibm.storage_virtualize    2.3.1
```

3. Jak widać, nie. Doinstaluj ją.

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-galaxy collection install \
ibm.power_vios
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/
artifacts/ibm-power_vios-1.3.0.tar.gz to /home/marcinek/.ansible/tmp/ansible-local-
1552ke8fkg8/tmpfu_v12z/ibm-power_vios-1.3.0-dwa2co7b
Installing 'ibm.power_vios:1.3.0' to
'/home/marcinek/.ansible/collections/ansible_collections/ibm/power_vios'
ibm.power_vios:1.3.0 was installed successfully
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-galaxy collection list | grep -i \
"^ibm"
ibm.power_vios            1.3.0
ibm.qradar                2.1.0
ibm.spectrum_virtualize   2.0.0
ibm.storage_virtualize    2.3.1
```

Może się też zdarzyć, że masz już tę kolekcję, np. w starszej wersji. Możesz wymusić jej instalację w nowszej poprzez dodanie parametru `--force` do polecenia instalacji.

## 5.2 Przygotowanie serwerów VIOS do pracy z Ansible

Według dokumentacji IBM VIOS 4.1 jest przygotowany do pracy z Ansiblem. Wcześniejsze wersje muszą być dostosowane.

W środowisku tych warsztatów, serwery VIOS zostały przygotowane dla Ciebie. Poniższe kroki są zamieszczone tu z kronikarskiego obowiązku, na wypadek gdyby ktoś chciał powtórzyć te ćwiczenia u siebie, choć osobiście nie polecam robienia ich na produkcji ;-)

1. Na wszystkich serwerach VIOS, które będą konfigurowane Ansiblem, załóż użytkownika *ansible* z następującymi uprawnieniami:

---

```
$ oem_setup_env

# mkuser roles=PAdmin,CacheAdm,FSAdmin,pkgadm \
    default_roles=PAdmin,CacheAdm,FSAdmin,pkgadm ansible
```

---

2. Nadaj założonym kontom hasła, oraz opcjonalnie, wymień się kluczami ssh. AIX lubi wymusić zmianę hasła przy pierwszym logowaniu, dlatego warto najpier się zalogować interaktywnie i ustawić coś sensownego:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ssh ansible@s824-6013-vios1
ansible@s824-6013-vios1's password:
[compat]: 3004-610 You are required to change your password.
    Please choose a new one.
The most recent software update has modified the current system rules.
These modifications have not been deployed on the system. To view the
modifications and deploy, run the 'rulescfgset' command.
WARNING: Your password has expired.
You must change your password now and login again!
Changing password for "ansible"
ansible's Old password:
ansible's New password:
Enter the new password again:
Connection to s824-6013-vios1 closed.
```

---

Operację powtórz na drugim VIOSie. Dla wygody wymień klucze pomiędzy kontem z którego puszczasz playbooki a użytkownikami *ansible* na VIOSach.

3. Zmodyfikuj plik *hosts* tak, żeby zawierał sekcję *[viosy:vars]* gdzie można podać grupowe zmienne. W tym wypadku ustaw zmienną *ansible\_user=ansible*. Sekcję ze zmiennymi umieść pod sekcją grupy.

---

```
[viosy]
s824-6013-vios2.iic
s824-6013-vios1.iic

[viosy:vars]
ansible_user=ansible
```

---

4. „Pingnij” Ansiblem viosy:

---

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible viosy -m ping
[WARNING]: No python interpreters found for host s824-6013-vios1.iic (tried ['python3.12',
'python3.11', 'python3.10', 'python3.9', 'python3.8', 'python3.7', 'python3.6',
'/usr/bin/python3',
'/usr/libexec/platform-python', 'python2.7', '/usr/bin/python', 'python'])
s824-6013-vios1.iic | FAILED! => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "module_stderr": "Shared connection to s824-6013-vios1.iic closed.\r\n",
  "module_stdout": "/bin/sh: /usr/bin/python: not found.\r\n",
  "msg": "MODULE FAILURE\nSee stdout/stderr for the exact error",
  "rc": 127
}
[WARNING]: No python interpreters found for host s824-6013-vios2.iic (tried ['python3.12',
'python3.11', 'python3.10', 'python3.9', 'python3.8', 'python3.7', 'python3.6',
'/usr/bin/python3',
'/usr/libexec/platform-python', 'python2.7', '/usr/bin/python', 'python'])
s824-6013-vios2.iic | FAILED! => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "module_stderr": "Shared connection to s824-6013-vios2.iic closed.\r\n",
  "module_stdout": "/bin/sh: /usr/bin/python: not found.\r\n",
  "msg": "MODULE FAILURE\nSee stdout/stderr for the exact error",
  "rc": 127
}
```

---



5. Nie miało prawa się udać. Standardowa instalacja VIOSa nie ma pythona. Dlatego trzeba zainstalować... DNF, który jako pre-requisite instaluje odpowiednią wersję Pythona. Pełna instrukcja instalacji znajduje się tu: <https://developer.ibm.com/tutorials/awb-configuring-dnf-create-local-repos-ibm-aix/>. Sprowadza się do ściągnięcia skryptu `dnf_aixtoolbox.sh` i puszczenia go.

```
root@vios1:/tmp # ./dnf_aixtoolbox.sh -d
Attempting download of dnf_bundle_aix_71_72.tar ...
Saving to 'dnf_bundle_aix_71_72.tar'...
208 MB received in 2 minutes (1.72 MB/sec)

Extracting dnf_bundle_aix_71_72.tar ...
x ca-certificates-2023.2.60-0.aix7.1.ppc.rpm, 991206 bytes, 1936 tape blocks
x dnf-4.2.17-32_4.aix7.1.noarch.rpm, 11135 bytes, 22 tape blocks
x dnf-automatic-4.2.17-32_4.aix7.1.noarch.rpm, 12450 bytes, 25 tape blocks
x dnf-data-4.2.17-32_4.aix7.1.noarch.rpm, 23693 bytes, 47 tape blocks

[ ... ]

28:python3-hawkey-0.39.1-32_3      ##### [ 93%]
29:python3-libdnf-0.39.1-32_3     ##### [ 97%]
30:rpm-python3-4.15.1-32_2        ##### [100%]
```

dnf installed successfully.

Please run 'dnf update' to update packages to the latest level.

Please note, RPM packages are downloaded in dnf cache /var/cache/dnf.

RPM packages install files go under the path /opt.

Hence it is recommended to always keep at least 512MB of free space in /var & /opt to avoid any download and installation/update failures.

## 6. Zaktualizuj pakiety rpm:

```
root@vios2:/ # dnf update
AIX generic repository      1.9 MB/s | 20 MB    00:10
AIX noarch repository      518 kB/s | 3.1 MB  00:06
AIX 7.2 specific repository 1.3 MB/s | 1.3 MB  00:01
Last metadata expiration check: 0:00:02 ago on Fri Apr 12 21:56:59 CEST 2024.
Dependencies resolved.

=====
Package                        Architecture  Version      Repository      Size
=====
Upgrading:
ca-certificates                ppc          2023.2.60-2  AIX_Toolbox     968 k
libcomps                       ppc          0.1.15-101  AIX_Toolbox     622 k
libmodulemd                    ppc          1.5.2-100   AIX_Toolbox     1.2 M
librepo                        ppc          1.11.0-103  AIX_Toolbox     336 k
libsmartcols                   ppc          2.34-101    AIX_Toolbox     614 k
libzstd                        ppc          1.5.2-3     AIX_Toolbox     1.5 M
p11-kit                        ppc          0.24.1-1    AIX_Toolbox     3.7 M

[ ... ]

Installed:
pinentry-1.0.0-1.ppc          bash-5.2.15-1.ppc          bzip2-1.0.8-2.ppc          check-0.13.0-1.ppc
curl-8.6.0-1.ppc              cyrus-sasl-2.1.28-1.ppc    db-1:5.3.28-1.ppc          expat-2.5.0-1.ppc
gdbm-1.23-1.ppc               gettext-0.21-2.ppc        glib2-2.76.3-2.ppc        gmp-6.3.0-1.ppc
gnupg2-2.4.3-1.ppc           gnutls-3.8.2-1.ppc        gpvme-1.13.1-101.ppc      info-7.0.2-1.ppc
json-c-0.17-1.ppc             krb5-libs-1.21.2-1.ppc    libassuan-2.5.6-1.ppc     libffi-3.4.4-2.ppc
libgcrypt-1.10.2-1.ppc       libgpg-error-1.45-1.ppc   libconv-1.17-1.ppc        libksba-1.6.3-1.ppc
libnghttp2-1.58.0-1.ppc      libssh2-1.10.0-2.ppc     libtasn1-4.19.0-1.ppc     libtextstyle-0.21-2.ppc
libunistring-1.1-1.ppc       libxml2-2.10.4-1.ppc     libyaml-0.2.5-1.ppc       ncurses-6.4-1.ppc
nettle-3.9.1-1.ppc           npth-1.5-1.ppc            openldap-2.5.16-1.ppc     pcre2-10.40-1.ppc
readline-8.2-1.ppc           sqlite-3.41.2-1.ppc       xz-libs-5.4.3-1.ppc       zlib-1.2.13-1.ppc
```

```
libgcc-1:10-2.ppc      libgcc10-10.3.0-6.ppc  libgomp-1:10-2.ppc    libgomp10-10.3.0-6.ppc
libstdc++-1:10-2.ppc  libstdc++10-10.3.0-6.ppc
```

```
Complete!
```

## 7. Zainstaluj pakiet `sudo`:

```
root@vios2:/ # dnf install sudo
```

## 8. Dodaj użytkownika `ansible` do bezhasłowych sudoersów. Utwórz plik `/etc/sudoers.d/ansible` o następującej zawartości:

```
root@vios1:/ # cat /etc/sudoers.d/ansible
ansible ALL=(ALL) NOPASSWD: ALL
```

## 9. Dodaj ścieżkę `/opt/freeware/bin` do systemowej zmiennej `$PATH`. (Z jakiegoś tajemniczego powodu Ansible nie używa zmiennych z `~/.profile`). W Pliku `/etc/environment` dopisz:

```
[ ... ]
PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin:/usr/java8_64/jre/bin:/usr/
java8_64/bin:/opt/freeware/bin
[ ... ]
```

## 10. „Pingnij” VIOSy ponownie:

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible viosy -m ping
[WARNING]: Platform aix on host s824-6013-vios2.iic is using the discovered Python
interpreter at /opt/freeware/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.16/reference_appendices/interpreter_discovery.html for more information.
s824-6013-vios2.iic | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/opt/freeware/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
[WARNING]: Platform aix on host s824-6013-vios1.iic is using the discovered Python
interpreter at /opt/freeware/bin/python3.9, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.16/reference_appendices/interpreter_discovery.html for more information.
s824-6013-vios1.iic | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/opt/freeware/bin/python3.9"
  },
  "changed": false,
  "ping": "pong"
}
```

## 5.3 Test kolekcji *ibm.power\_vios*

Żeby przetestować przetestować moduł `ibm.power_vios.mapping_facts` należy:

1. W katalogu Twojego projektu, utwórz plik `vios_info.yaml` o następującej zawartości:

```
---
# Przykładowy playbook zbierający informacje o VIOS

- hosts: viosy
  remote_user: ansible
  gather_facts: True

  tasks:
    - name: Odpytywanie VIOSa o mapowania
      ibm.power_vios.mapping_facts:

    - name: Wyświetlenie informacji o mapowaniach
      debug:
        var: ansible_facts.mappings
```

2. Uruchom playbooka:

```
marcinek@ender:~/prog/ansible/common-04.2024$ ansible-playbook vios_info.yaml
```

```
PLAY [viosy] *****

TASK [Gathering Facts] *****
[WARNING]: Platform aix on host s824-6013-vios2.iic is using the discovered Python interpreter at
/opt/freeware/bin/python3.9, but future installation of another Python interpreter could change the meaning of that
path. See https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more
information.
ok: [s824-6013-vios2.iic]
[WARNING]: Platform aix on host s824-6013-vios1.iic is using the discovered Python interpreter at
/opt/freeware/bin/python3.9, but future installation of another Python interpreter could change the meaning of that
path. See https://docs.ansible.com/ansible-core/2.16/reference_appendices/interpreter_discovery.html for more
information.
ok: [s824-6013-vios1.iic]

TASK [Odpytywanie VIOSa o mapowania] *****
ok: [s824-6013-vios2.iic]
ok: [s824-6013-vios1.iic]

TASK [Wyświetlenie informacji o mapowaniach] *****
ok: [s824-6013-vios2.iic] => {
  "ansible_facts.mappings": {
    "vscsi": {}
  }
}
ok: [s824-6013-vios1.iic] => {
  "ansible_facts.mappings": {
    "vscsi": {
      "vhost1": {
        "clientid": 2,
        "physloc": "U8286.42A.21C7FDV-V100-C12",
        "vtlds": {
          "cd4aix2": {
            "backing": "/var/vio/VMLibrary/aix7321.iso",
            "lun": "0x8100000000000000",
            "status": "Available"
          }
        }
      }
    }
  }
}
```

```
    },  
    "vhost8": {  
      "clientid": 9,  
      "physloc": "U8286.42A.21C7FDV-V100-C19",  
      "vtlds": {  
        "cd4red1": {  
          "backing": "/var/vio/VMLibrary/aix732lgold.iso",  
          "lun": "0x8100000000000000",  
          "status": "Available"  
        }  
      }  
    },  
    "vhost9": {  
      "clientid": 10,  
      "physloc": "U8286.42A.21C7FDV-V100-C20",  
      "vtlds": {  
        "cd4red2": {  
          "backing": "/var/vio/VMLibrary/aix732lgold.iso",  
          "lun": "0x8100000000000000",  
          "status": "Available"  
        }  
      }  
    }  
  }  
}  
  
}
```

```
PLAY RECAP *****
s824-6013-vios1.iic : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
s824-6013-vios2.iic : ok=3    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

### 3. Git add, git commit, git push :-)

## 6 Użyteczne informacje i linki

Tu znajdziesz użyteczne linki do dokumentacji.

- Setup projektu Ansible:  
[https://docs.ansible.com/ansible/latest/tips\\_tricks/sample\\_setup.html](https://docs.ansible.com/ansible/latest/tips_tricks/sample_setup.html)
- Bardzo fajny RedBook: <https://www.redbooks.ibm.com/redpieces/pdfs/sg248551.pdf>
- 
- Kolekcje modułów Ansible od IBM: <https://galaxy.ansible.com/ui/namespaces/ibm/>

Te, których używasz w tych ćwiczeniach:

- HMC: [https://galaxy.ansible.com/ui/repo/published/ibm/power\\_hmc/](https://galaxy.ansible.com/ui/repo/published/ibm/power_hmc/) oraz <https://ibm.github.io/ansible-power-hmc/index.html>.
- Storage Virtualize:  
[https://galaxy.ansible.com/ui/repo/published/ibm/storage\\_virtualize/](https://galaxy.ansible.com/ui/repo/published/ibm/storage_virtualize/)
- ibm.power\_vios
  - Instalacja: <https://ibm.github.io/ansible-power-vios/installation.html>
  - Konfiguracja VIOS: <https://ibm.github.io/ansible-power-vios/quickstart.html>
- DNF na AIXie i VIOSie: <https://developer.ibm.com/tutorials/awb-configuring-dnf-create-local-repos-ibm-aix/>
- 
-