Próbny egzamin EX294 i strategia na dzień egzaminu

Poziom: średnio

Dotyczy: ansible-core 2.16 (RHEL 10), RHEL 9 i 10 jako węzły zarządzane, dni 29 i 30 programu

Czas: 4 godziny egzaminu próbnego, 1 godzina analizy, egzamin około 4 godzin

Dzień 29 to egzamin próbny na czas na świeżym laboratorium, dzień 30 to prawdziwy egzamin EX294. Poniżej lista dziewiętnastu własnych zadań w stylu celów, strategia pracy z playbookami, sprawdzanie trwałości po restarcie i to, co robimy po wyniku.

ZAŁOŻENIA SCENARIUSZA

wszystkie cztery maszyny wracają do migawki `czysty-start` ze strony Plan 30 dni i laboratorium RHEL w Zielonej, czyli bez śladów ćwiczeń z tygodni 3 i 4. Węzłem sterującym jest `rhlab01`, a węzły zarządzane to `rhlab02`, `rhlab03` i `rhlab04`. Strona Red Hat o EX294 mówi o RHEL 9, więc kto chce dokładnie takich warunków, robi próbę na `rhlab03` jako węźle sterującym. Pliki z poprzednich tygodni (archiwum kolekcji `zielona-lab-1.0.0.tar.gz`, repozytorium bare na `rhlab02`) odtwarzamy według Tydzień 4 do EX294: błędy, sekrety, role i automatyzacja zadań RHCSA. Czas trwania egzaminu, cztery godziny, pochodzi od partnerów szkoleniowych, a nie ze strony Red Hat.

Skróty użyte na tej stronie (9)

Skrót	Co znaczy
EX200	numer egzaminu praktycznego RHCSA, na którym kandydat konfiguruje żywy system RHEL bez dostępu do internetu
EX294	numer egzaminu praktycznego RHCE w ścieżce Ansible: kandydat pisze playbooki, które Red Hat uruchamia na świeżych systemach
EX374	numer egzaminu specjalisty Red Hat o tworzeniu automatyzacji w Ansible Automation Platform
EX467	numer egzaminu specjalisty Red Hat o zarządzaniu automatyzacją w Ansible Automation Platform

Skrót	Co znaczy
RHCE	Red Hat Certified Engineer: certyfikat Red Hat za automatyzację administracji; w ścieżce Ansible to zdany EX200 i zdany EX294
RHCSA	Red Hat Certified System Administrator: certyfikat Red Hat za praktyczną obsługę systemu RHEL, zdaje się go egzaminem EX200
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SELinux	Security-Enhanced Linux: dodatkowa warstwa uprawnień w jądrze RHEL
YAML	YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

Fiszka

Komendy wykonujemy w bash jako `tomek` na węźle sterującym w katalogu `~/ansible`. Służą do sprawdzania pracy w trakcie i na końcu, więc nie zmieniają systemów, poza restartem. Nazwy takie jak `NazwaPlaybooka` czy `NazwaHosta` zastępujemy własnymi.

◫◫ TYLKO ODCZYT

Sprawdzenie samej składni YAML: yaml.safe_load

```
python3 -c 'import sys, yaml; yaml.safe_load(open(sys.argv[1])); print("YAML poprawny")'
NazwaPliku.yml
```

Odpowiada na pytanie, czy plik jest poprawnym YAML, a nie czy jest poprawnym playbookiem. Przydaje się przy plikach zmiennych i szablonach, które `--syntax-check` nie bada.

◫◫ **YAML** = YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach · **python3** = uruchom interpreter Pythona w wersji 3 albo wskazany skrypt

◫◫ ZMIENIA SYSTEM

Sprawdzenie, czy drugie uruchomienie nic nie zmienia: grep changed

```
ansible-playbook NazwaPlaybooka.yml | grep -E "changed=|failed="
```

Przy poprawnym playbooku drugie uruchomienie pokazuje `changed=0` i `failed=0`. Zadania, które zawsze zgłaszają zmianę, poprawiamy przez `changed_when` albo lepszy moduł.

ansible-playbook = uruchom scenariusz Ansible

TYLKO ODCZYT

Przykłady użycia modułu z dokumentacji: ansible-doc

```
ansible-doc NazwaModulu | sed -n '/^EXAMPLES/,/^RETURN/p'
```

Wypisuje tylko sekcję z przykładami, czyli to, co najczęściej kopiujemy z dokumentacji. Nazwę modułu znajdujemy przez `ansible-doc -l | grep -i NazwaFragmentu`.

ansible-doc = od Ansible i documentation: pokazuje dokumentację modułu wraz z przykładami, bez dostępu do internetu

TYLKO ODCZYT

Wypisanie zmiennej hosta razem ze skąd pochodzi: debug

```
ansible NazwaHosta -m ansible.builtin.debug -a "var=hostvars[inventory_hostname]"
```

Pokazuje wszystkie zmienne i fakty hosta w jednym miejscu. Wynik jest długi, więc zawężamy go do jednego klucza, na przykład `var=ansible_facts['default_ipv4']`.

ansible = bazowe polecenie Ansible do wykonania pojedynczej komendy ad-hoc na grupie maszyn, bez playbooka

TYLKO ODCZYT

Odczytanie jednego faktu: setup z filtrem

```
ansible NazwaHosta -m ansible.builtin.setup -a "filter=ansible_default_ipv4"
```

Faktów używamy w szablonach przez `ansible_facts['default_ipv4']['address']`, więc nazwę klucza najpierw odczytujemy.

ansible = bazowe polecenie Ansible do wykonania pojedynczej komendy ad-hoc na grupie maszyn, bez playbooka

TYLKO ODCZYT

Wyszukanie poleceń zamiast modułów w playbookach: grep

```
grep -nE "ansible.builtin.(command|shell):|^ +(command|shell):" *.yml
```

Pokazuje miejsca, w których użyliśmy `command` albo `shell` zamiast modułu. Każde z nich sprawdzamy, bo moduł jest zwykle bardziej powtarzalny.

grep = global regular expression print: szukaj wzorca w tekście

TYLKO ODCZYT

Restart wszystkich węzłów z czekaniem na powrót: reboot

```
ansible lab -b -m ansible.builtin.reboot -a "reboot_timeout=600"
```

Moduł `reboot` restartuje węzeł i czeka, aż znów odpowiada. To najprostszy sposób sprawdzenia trwałości konfiguracji bez ręcznego chodzenia po maszynach.

ansible = bazowe polecenie Ansible do wykonania pojedynczej komendy ad-hoc na grupie maszyn, bez playbooka

TYLKO ODCZYT

Sprawdzenie plików fstab na wszystkich węzłach: findmnt --verify

```
ansible lab -b -m ansible.builtin.command -a "findmnt --verify"
```

Wynik bez błędów po restarcie oznacza, że wpisy z `/etc/fstab` są poprawne. Wynik z błędem pokazuje wiersz do poprawy.

ansible = bazowe polecenie Ansible do wykonania pojedynczej komendy ad-hoc na grupie maszyn, bez playbooka

TYLKO ODCZYT

Sprawdzenie, że usługa wstanie po restarcie: systemctl is-enabled

```
ansible web -b -m ansible.builtin.command -a "systemctl is-enabled httpd"
```

Odpowiedź `enabled` jest potrzebna razem z `active`, bo sama działająca usługa mogła zostać uruchomiona ręcznie.

ansible = bazowe polecenie Ansible do wykonania pojedynczej komendy ad-hoc na grupie maszyn, bez playbooka

□□ TYLKO ODCZYT

Sprawdzenie, co jest w repozytorium: git status

```
git status --short
git log --oneline
```

Pierwsza komenda pokazuje pliki nieśledzone albo zmienione, a druga zapisane zmiany. Przed końcem egzaminu zmiany powinny być zapisane i wysłane.

□□ **git** = system kontroli wersji (nazwa to brytyjski slang: nieznośny typ)

□□ ZMIENIA SYSTEM

Ustawienie zegara ostrzegawczego: systemd-run

```
systemd-run --user --on-active=210m --unit=egzamin-alarm bash -c 'echo "30 minut do konca" | wall'
```

Zegar uruchamia się po 210 minutach i wypisuje komunikat na terminalu. W trakcie egzaminu zwykle nie mamy takiego zegara pod ręką, więc ustawiamy własny albo patrzymy na zegarek.

□□ **systemd-run** = od systemd i run: uruchamia polecenie jako tymczasową jednostkę systemd, na przykład z opóźnieniem albo jako zegar

Dzień 29: egzamin próbny na czas

□□ Z ŻYCIA WZIĘTE

Środa, 9:00. Ola kładzie na biurku Tomka zegar kuchenny i wyłącza mu zakładki z dokumentacją w przeglądarce. „Cztery godziny. Masz `man`, `ansible-doc` i `/usr/share/doc`. Nie pytasz mnie o nic, dopóki nie wybije czas”. Tomek wraca do migawki `czysty-start` na wszystkich czterech maszynach i czyta listę zadań raz, od początku do końca, zanim dotknie klawiatury.

Zasady

1. Cztery godziny na zegarze, bez przerw na szukanie w internecie. Dozwolone: `man`, `ansible-doc`, `ansible-navigator doc`, katalog `/usr/share/doc` i własne notatki w `~/ansible`. Niedozwolone: nasza baza wiedzy, wyszukiwarka i notatki spoza laboratorium.
2. Wszystko robimy Ansible: pliki z nazwami podanymi w zadaniach, uruchamiane z katalogu `~/ansible`. Ręczne zmiany na węzłach nie liczą się, bo egzaminator uruchamia nasze playbooki na świeżych systemach.
3. Zadanie zaliczamy w całości albo wcale, bez punktów częściowych. Przyjmujemy własną skalę 100 punktów i próg zdania 70, co jest tylko naszym założeniem do treningu. Prawdziwego progu strona Red Hat nie podaje, a część źródeł wtórnych mówi o 210 punktach z 300.
4. Po zakończeniu restartujemy wszystkie węzły i sprawdzamy trwałość (zadanie 19). Nie poprawiamy niczego ręcznie po zegarze.

Lista zadań

Nr	Zadanie	Cel egzaminu	Pkt
1	`ansible.cfg` (inwentarz `./inventory`, użytkownik `tomek`, `roles_path` z katalogiem systemowym ról, `collections_path` z katalogiem systemowym, `sudo` bez hasła), `ansible-navigator.yml` (bez kontenera, wynik w terminalu), klucze SSH i `sudo` na węzłach	konfiguracja Ansible, węzły zarządzane	5
2	Inwentarz: `web` (`rhlab04`), `dane` (`rhlab02`, `rhlab03`), `lab` z obu grup. Zmienna `srodowisko: egzamin` dla `lab` i `kontakt: it@zielona.example` dla `web` w `group_vars`	inwentarze, zmienne	4
3	Skrypt `adhoc.sh` (wykonywalny) z poleceniami ad hoc: instalacja `tmux` na `lab` i włączenie z uruchomieniem `chronyd` na `lab`	polecenia ad hoc	4
4	`pakiety.yml`: na `web` pakiety `httpd` i `mod_ssl`, na `dane` pakiety `rsync` i `tmux`, na wszystkich usunięty `telnet`	pakiety	5

Nr	Zadanie	Cel egzaminu	Pkt
5	`www.yml`: na `web` działający i włączony `httpd`, stała reguła zapory dla `http`, strona `index.html` z szablonu `index.html.j2` o treści `Zielona <srodowisko>: <FQDN> (<adres IPv4>)`, restart `httpd` tylko po zmianie strony	szablony, usługi, zapora, handlery	6
6	`hosts.yml` i `hosts.j2`: na `lab` plik `/etc/hosts.zielona` z wierszem `adres FQDN nazwa` dla każdego hosta z `lab`, wygenerowany z faktów	szablony, fakty	7
7	Zaszyfrowany `vars/sejf.yml` ze zmienną `haslo_uzytkownikow` (hasło do sejfu w `~/.vault_pass`) i jawny `vars/uzytkownicy.yml` z listą `ola` (UID 3101, admin), `marek` (UID 3102), `kasia` (UID 3103)	Vault	4
8	`uzytkownicy.yml`: na `lab` grupa `zespol`, wszystkie trzy konta w tej grupie z hasłem z sejfu, a konta z `admin: true` dodatkowo w grupie `wheel`, ale tylko na hostach z grupy `web`. Hasło ustawiane tylko przy tworzeniu konta	użytkownicy i grupy, pętle, warunki	8
9	Rola `baner` w `roles/baner` (zadanie, szablon `baner.j2`, wartość domyślna), zapisująca `/etc/issue.net` z tekstem `Zielona <srodowisko>: dostep tylko dla uprawnionych`, wywołana z `baner.yml` na `lab`	role, szablony	5
10	Kolekcja `zielona.lab` z archiwum `zielona-lab-1.0.0.tar.gz` zainstalowana do `./collections` i `kolekcja.yml` wywołujący na `web` rolę `zielona.lab.motd`	kolekcje	5

Nr	Zadanie	Cel egzaminu	Pkt
11	`czas.yml`: na `lab` rola systemowa `timesync` z serwerami `0.pl.pool.ntp.org` i `1.pl.pool.ntp.org` z `iburst`	role systemowe RHEL	5
12	`magazyn.yml`: na `lab` grupa `vgDane` z `/dev/sdb`, wolumen `lvDane` 2 GB, XFS, montowanie w `/srv/dane` na stałe. Gdy utworzenie grupy się nie uda, komunikat `Brak dysku sdb` i dalsze wykonanie	magazyn, systemy plików, obsługa błędów	8
13	`cron.yml`: na `dane` zadanie cron użytkownika `root` o nazwie `Kopia testowa`, codziennie o 2:15, dopisujące datę do `/var/log/kopia-test.log`	zadania zaplanowane	4
14	`selinux.yml`: na `web` port 8888 z etykietą `http_port_t` i `Listen 8888` w konfiguracji `httpd`, stała reguła zapory dla 8888/tcp, trwale włączony przełącznik `httpd_can_network_connect`, katalog `/srv/strona` z etykietą `httpd_sys_content_t`	bezpieczeństwo, SELinux	7
15	`kopia.yml`: na `lab` archiwum `tar.gz` plików `/etc/hosts` i `/etc/hostname` w `/var/tmp/kopia-<nazwa hosta>.tar.gz`, pobrane do `~/kopie` na węźle sterującym	archiwizacja	4
16	`bledy.yml`: na `lab` instalacja pakietu `ncdu` (spoza repozytoriów systemu). Gdy się nie uda, komunikat `Pakiet niedostepny`, a zawsze plik `/var/tmp/bledy-koniec`	obsługa błędów	5
17	`raport.yml`: na `lab` plik `/root/raport.txt` z FQDN, wersją jądra (z polecenia `uname -r` zapisanego w zmiennej) i pamięcią w MB, pobrany do `~/raporty`	zmienne, fakty, szablony	4

Nr	Zadanie	Cel egzaminu	Pkt
18	Repozytorium `ssh://tomek@rhlab02/home/tomek/repo/egzamin.git`: wszystkie playbooki, szablony i role zapisane i wysłane, bez pliku hasła do sejfu	Git	4
19	`po-restarcie.yml`: restart wszystkich węzłów modułem `reboot` i sprawdzenie, że po powrocie działa `chronyd` na `lab`, `httpd` na `web` i jest zamontowane `/srv/dane` na `lab`	trwałość konfiguracji	6

Suma punktów: 100. Zadanie 10 wymaga archiwum z dnia 26, więc przed próbą przygotowujemy je w katalogu `~/` i sprawdzamy, że nie ma go jeszcze w `~/ansible/collections`.

Sprawdzenie

Po czterech godzinach sprawdzamy każde zadanie osobno komendami z odpowiedzi. Zadanie liczymy, gdy wszystkie jego komendy dają oczekiwany wynik, a drugie uruchomienie playbooka nie zmienia niczego (`changed=0`).

Komendy sprawdzające do zadań 1 do 19

```
cd ~/ansible
# 1
ansible-config dump --only-changed
ansible all -m ansible.builtin.ping
ansible all -b -m ansible.builtin.command -a "id"
# 2
ansible-inventory --graph --vars
# 3
test -x adhoc.sh && bash -n adhoc.sh && cat adhoc.sh
ansible lab -b -m ansible.builtin.command -a "systemctl is-enabled chronyd"
# 4
ansible web -m ansible.builtin.command -a "rpm -q httpd mod_ssl"
ansible dane -m ansible.builtin.command -a "rpm -q rsync tmux"
ansible lab -m ansible.builtin.command -a "rpm -q telnet"
```

```
# 5
curl -s http://rhlab04
ansible web -b -m ansible.builtin.command -a "firewall-cmd --permanent --list-services"
# 6
ansible lab -m ansible.builtin.command -a "cat /etc/hosts.zielona"
# 7
head -1 vars/sejf.yml
ansible-vault view vars/sejf.yml
# 8
ansible lab -m ansible.builtin.command -a "id ola"
ansible lab -m ansible.builtin.command -a "getent group wheel zespol"
# 9
ansible lab -m ansible.builtin.command -a "cat /etc/issue.net"
# 10
ansible-galaxy collection list -p ./collections
ansible web -m ansible.builtin.command -a "cat /etc/motd"
# 11
ansible lab -m ansible.builtin.command -a "grep -E '^(server|pool)' /etc/chrony.conf"
# 12
ansible lab -b -m ansible.builtin.command -a "findmnt /srv/dane"
ansible lab -b -m ansible.builtin.command -a "grep dane /etc/fstab"
# 13
ansible dane -b -m ansible.builtin.command -a "crontab -l -u root"
# 14
ansible web -b -m ansible.builtin.command -a "semanage port -l"
ansible web -b -m ansible.builtin.command -a "getsebool httpd_can_network_connect"
ansible web -b -m ansible.builtin.command -a "ls -dZ /srv/strona"
# 15
ls ~/kopie
# 16
ansible lab -m ansible.builtin.command -a "ls /var/tmp/bledy-koniec"
# 17
ls ~/raporty
# 18
git -C ~/ansible log --oneline
git -C ~/ansible status --short
# 19
ansible-playbook po-restarcie.yml
```

Dla zadania 10 lista kolekcji ma pokazać `zielona.lab`, a komenda `cat /etc/motd` tekst z roli kolekcji. Zadanie 16 jest zaliczone też wtedy, gdy `ncdu` okazał się dostępny: wtedy ważne jest, że plik końca istnieje.

Przykładowe rozwiązania najtrudniejszych zadań: 6, 8, 14 i 19

Zadanie 6, plik `hosts.yml`:

```
---
- name: Plik hosts laboratorium
  hosts: lab
  become: true
  tasks:
    - name: Plik hosts.zielona z szablonu
      ansible.builtin.template:
        src: hosts.j2
        dest: /etc/hosts.zielona
        owner: root
        group: root
        mode: "0644"
```

Zadanie 6, plik `templates/hosts.j2`:

```
# Zarzadzane przez Ansible
{% for h in groups['lab'] %}
{{ hostvars[h]['ansible_facts']['default_ipv4']['address'] }} {{
hostvars[h]['ansible_facts']['fqdn'] }} {{ hostvars[h]['ansible_facts']['hostname'] }}
{% endfor %}
```

Szablon korzysta z faktów innych hostów, więc play musi je zebrać dla wszystkich (`gather_facts` jest domyślnie włączone). Zadanie 8, plik `uzytkownicy.yml` z `vars/uzytkownicy.yml` w postaci listy słowników z polami `nazwa`, `uid` i `admin`:

```
---
- name: Konta zespolu
  hosts: lab
  become: true
  vars_files:
    - vars/uzytkownicy.yml
```

```
    - vars/sejf.yml
  tasks:
    - name: Grupa zespol
      ansible.builtin.group:
        name: zespol
        state: present

    - name: Konta zespolu
      ansible.builtin.user:
        name: "{{ item.nazwa }}"
        uid: "{{ item.uid }}"
        groups: "{{ ['zespol'] + (['wheel'] if item.admin and inventory_hostname in
groups['web'] else []) }}"
        append: true
        password: "{{ haslo_uzytkownikow | password_hash('sha512') }}"
        update_password: on_create
        state: present
      loop: "{{ uzytkownicy }}"
      loop_control:
        label: "{{ item.nazwa }}"
      no_log: true
```

Zadanie 14:

```
---
- name: SELinux i httpd na web
  hosts: web
  become: true
  tasks:
    - name: Pakiety
      ansible.builtin.dnf:
        name:
          - httpd
          - policycoreutils-python-utils
        state: present

    - name: Dodatkowy port nasluchu
      ansible.builtin.lineinfile:
        path: /etc/httpd/conf/httpd.conf
```

```
    line: "Listen 8888"
    insertafter: '^Listen 80$'
  notify: Restart httpd

- name: Etykieta portu 8888
  community.general.seport:
    ports: "8888"
    proto: tcp
    setype: http_port_t
    state: present

- name: Port 8888 w zaporze
  ansible.posix.firewalld:
    port: 8888/tcp
    permanent: true
    immediate: true
    state: enabled

- name: Przelacznik httpd_can_network_connect
  ansible.posix.seboolean:
    name: httpd_can_network_connect
    state: true
    persistent: true

- name: Katalog strony
  ansible.builtin.file:
    path: /srv/strona
    state: directory
    mode: "0755"

- name: Regula etykiety katalogu strony
  community.general.sefcontext:
    target: '/srv/strona(/.*)?'
    setype: httpd_sys_content_t
    state: present

- name: Zastosowanie etykiet
  ansible.builtin.command: restorecon -Rv /srv/strona
```

```
      register: restorecon_wynik
      changed_when: "'Relabeled' in restorecon_wynik.stdout"

    - name: Wlaczenie i start httpd
      ansible.builtin.service:
        name: httpd
        state: started
        enabled: true

  handlers:
    - name: Restart httpd
      ansible.builtin.service:
        name: httpd
        state: restarted
```

Zadanie 19:

```
---
- name: Restart i sprawdzenie trwalosci
  hosts: lab
  become: true
  tasks:
    - name: Restart wezlow
      ansible.builtin.reboot:
        reboot_timeout: 600

    - name: Zebranie stanu uslug
      ansible.builtin.service_facts:

    - name: Zebranie punktow montowania
      ansible.builtin.setup:
        gather_subset:
          - mounts

    - name: Chronyd dziala
      ansible.builtin.assert:
        that:
          - ansible_facts['services']['chronyd.service']['state'] == 'running'
        fail_msg: "chronyd nie dziala po restarcie"
```

```
    - name: Magazyn zamontowany
      ansible.builtin.assert:
        that:
          - ansible_facts['mounts'] | selectattr('mount', 'equalto', '/srv/dane') | list |
length == 1
        fail_msg: "/srv/dane nie jest zamontowany po restarcie"

    - name: Httpd dziala na web
      ansible.builtin.assert:
        that:
          - ansible_facts['services']['httpd.service']['state'] == 'running'
        fail_msg: "httpd nie dziala po restarcie"
      when: inventory_hostname in groups['web']
```

Moduł `assert` kończy się błędem z naszym komunikatem, gdy warunek jest fałszywy, więc zamiast oglądać wyniki ręcznie, dostajemy jednoznaczne `ok` albo `FAILED`.

Analiza po egzaminie próbnym

Po zamknięciu zegara robimy jedną godzinę analizy, nie więcej. Dla każdego niezaliczonego zadania zapisujemy w dzienniku trzy rzeczy: co poszło nie tak (składnia, moduł, zmienna, uprawnienia, trwałość), gdzie szukalibyśmy odpowiedzi na prawdziwym egzaminie (`ansible-doc`, `man`, `/usr/share/doc`) i ile minut stracono. Na koniec wybieramy dwa tematy do powtórki, tylko dwa, i robimy je na dzień przed egzaminem.

Najczęstsze braki po pierwszej próbie to: brak `become: true` w play, brak kolekcji `ansible.posix`, zła nazwa grupy w `hosts:`, zapomniany `notify` albo handler o innej nazwie, nieujęty w cudzysłów napis z dwukropkiem, niestała reguła zapory (bez `permanent: true`) i pominięcie restartu na końcu.

Dzień 30: strategia na egzamin EX294

> **□□ Z ŻYCIA WZIĘTE**

> Czwartek, 8:30. Tomek sprawdza trzeci raz, czy ma dokument tożsamości i czy kamera widzi biurko, choć Ola mówi, że wszystko jest dobrze. Na kartce przy klawiaturze ma pięć zdań, które wypisali razem poprzedniego dnia: zacząć od konfiguracji, uruchamiać po każdym zadaniu, zapisywać w Git co kilka zadań, na koniec restart i na końcu nie ruszać niczego, czego nie sprawdzamy.

Przed egzaminem

- Sprawdzamy przy rezerwacji, czy egzamin odbywa się na RHEL 9 i jakich wersji narzędzi używa, bo strona Red Hat mówi o RHEL 9, a zmiana na RHEL 10 nie została ogłoszona. Gdy w opisie zmian pojawi się RHEL 10, układ ćwiczeń zostaje ten sam, a różnice dotyczą wersji `ansible-core` i ścieżek systemowych.
- Dla egzaminu zdalnego sprawdzamy stanowisko wcześniej: kamerę, mikrofon, łącze, oświetlenie i uporządkowane biurko. Szczegóły opisuje strona Red Hat o egzaminach zdalnych, więc tych wymagań nie zgadujemy.
- Pamiętamy, że egzamin jest praktyczny: internet nie jest dostępny, a dokumentacja dostarczana z produktem jest dostępna w systemie.
- Nie ćwiczymy nowych rzeczy dzień przed egzaminem. Robimy jedną krótką powtórkę dwóch słabych tematów i idziemy spać.

Kolejność pracy na egzaminie

1. Czytamy całą listę zadań raz, zanim zaczniemy pisać, i oznaczamy te, które zależą od innych (konfiguracja i inwentarz są pierwsze, zadania na ich podstawie po nich).
2. Zaczynamy od konfiguracji: `ansible.cfg`, inwentarz, klucze i `sudo`. Bez tego żadne zadanie się nie uruchomi, więc błąd tu kosztuje wszystko.
3. Potem idziemy w kolejności od łatwych do trudnych: ad hoc, pakiety, usługi, użytkownicy. Zadania z szablonami, rolami i magazynem zostawiamy na środek. Zadaniom, które nie wychodzą po dziesięciu minutach, dajemy znacznik i wracamy do nich na końcu.
4. Po każdym zadaniu uruchamiamy playbook, potem `--syntax-check`, potem drugi raz i sprawdzamy `changed=0`. Zadanie jest skończone, gdy przeszło wszystkie trzy kroki.
5. Dokumentację czytamy w kolejności: `ansible-doc` modułu, jego przykłady, potem `/usr/share/doc`. Szukamy przykładu, kopiujemy, zmieniamy nazwy.
6. Co kilka zadań zapisujemy zmiany: `git add`, `git commit`, `git push`. Jeśli coś się wysypie, jest do czego wrócić.
7. Zostawiamy co najmniej 20 minut na końcowy restart i sprawdzenie. Najpierw restart modułem `reboot`, potem komendy kontrolne, a poprawki dopiero po wynikach.

Trwałość i powtarzalność

Egzaminator uruchamia playbooki na świeżych systemach, więc sprawdzana jest powtarzalność, a nie stan naszych maszyn. Oznacza to cztery nawyki:

- **Każde zadanie ma własne `name`, nazwę pliku zgodną z poleceniem i `hosts:` zgodne z treścią.** Różnica w nazwie pliku zwykle kosztuje całe zadanie.
- **Wszystko, co ma przetrwać restart, jest trwałe:** `enabled: true` dla usług, `permanent: true` dla zapory, `persistent: true` dla przełączników SELinux, wpis `mount` w `fstab`, reguły `sefcontext` zamiast samego `chcon`.
- **Moduły zamiast `command` i `shell`.** Polecenia dopuszczamy tylko tam, gdzie modułu brak (na przykład `restorecon`), i wtedy dodajemy `changed_when`.
- **Sekrety nie leżą w repozytorium w postaci jawnej.** Plik hasła do sejfu jest poza repozytorium albo na liście `.gitignore`, a zadania z hasłami mają `no_log: true`.

Kiedy coś nie działa

Objaw	Pierwszy ruch
Playbook nie startuje	`ansible-playbook NazwaPlaybooka.yml --syntax-check`, potem sprawdzenie wcięć i nazw modułów
`UNREACHABLE`	`ansible NazwaHosta -m ansible.builtin.ping -vvv`, klucz SSH i nazwa w inwentarzu
`Missing sudo password`	plik w `/etc/sudoers.d/` na węźle albo `become` w `ansible.cfg`

Objaw	Pierwszy ruch
Nie znaleziono modułu	`ansible-doc -l
Zmienna niezdefiniowana	`ansible NazwaHosta -m ansible.builtin.debug -a "var=NazwaZmiennej"`, plik `group_vars` obok inwentarza
Zadanie zawsze zgłasza zmianę	`changed_when`, moduł zamiast `command`, `creates` albo `removes`
Po restarcie czegoś brak	porównanie z listą trwałości powyżej, zwykle brak `enabled`, `permanent` albo `persistent`
Utknęliśmy na jednym zadaniu	znacznik, następne zadanie, powrót po wszystkich innych

Po wyniku

Wynik pochodzi od Red Hat, strona egzaminu mówi, że oficjalne wyniki są dostępne tylko w Red Hat Certification Central i zwykle w ciągu trzech amerykańskich dni roboczych. Raport pokazuje łączną liczbę punktów, bez podziału na zadania, więc przy niezdanym egzaminie musimy sami odtworzyć w pamięci, co nie wyszło, najlepiej tego samego dnia. Zasady ponownego podejścia i ich koszt sprawdzamy w opisie egzaminu i w regulaminie, bo ich nie potwierdzono.

Po zdaniu EX294 i wcześniej EX200 tytuł RHCE w ścieżce Ansible jest kompletny. Dalsze kroki, które wymienia sama strona egzaminu, to specjalizacje Red Hat Certified Specialist: Developing Automation with Ansible Automation Platform (EX374) oraz Managing Automation with Ansible Automation Platform (EX467). Opisy tych egzaminów i ich zakres sprawdzamy na stronach Red Hat, a nie z pamięci.

Częste błędy

Objaw	Przyczyna	Co robimy
Playbook działa raz, a przy drugim uruchomieniu zgłasza zmiany	`command` i `shell` bez `changed_when` albo `creates`	moduł zamiast polecenia albo `changed_when`
Po restarcie nie ma zapory albo etykiety	`permanent: false`, `persistent: false` albo sam `chcon`	`permanent: true`, `persistent: true`, `sefcontext` z `restorecon`
Playbook z hasłem nie działa na egzaminie	plik z hasłem sejfu został w poprzednim środowisku	plik hasła tworzymy sami według treści zadania
Szablon pokazuje puste miejsca	fakty nie zebrane albo zły klucz	`gather_facts: true`, `setup` z filtrem i poprawna nazwa klucza
Rola systemowa nie została znaleziona	`roles_path` bez katalogu systemowego	`ansible-galaxy role list` i dopisanie ścieżki
Zadanie z dyskiem psuje cały playbook	dysk ma inną nazwę albo go nie ma	`block` i `rescue`, sprawdzenie `ansible_facts['devices']`
Wcięcie w YAML psuje zadanie	tabulatory zamiast spacji	edytor z widocznymi białymi znakami, `--syntax-check`
Zmiany nie trafiły do repozytorium	brak `git push` albo brak `git add` na nowych plikach	`git status --short` na końcu

Co dalej

Po EX294 zostaje utrzymanie nawyku: dwa playbooki w tygodniu na własnym laboratorium i raz na kwartał powtórka z restartem. Wiedza o RHCSA starzeje się razem z wersją RHEL, a certyfikat jest ważny trzy lata, więc w tym czasie warto przejść na nowy zestaw celów. Stronę o całym programie, laboratorium i mapie dni opisuje Plan 30 dni i laboratorium RHEL w Zielonej, a opowieść o tym, skąd się wzięły te egzaminy, RHCSA i RHCE: egzaminy, w których nie ma ani jednego pytania

testowego.

Źródła: strona Red Hat o egzaminie EX294 (cele, narzędzia, brak internetu na egzaminie, sprawdzanie playbooków na świeżych systemach, wyniki zwykle w ciągu trzech amerykańskich dni roboczych, rekomendowane kolejne egzaminy EX374 i EX467, stan na 10 października 2026); dokumentacja Ansible (moduły `ansible.builtin`, `ansible.posix`, `community.general`); Red Hat, Considerations in adopting RHEL 10. Czas czterech godzin pochodzi od partnerów szkoleniowych i nie jest potwierdzony przez Red Hat. Własna skala punktów i próg 70 to założenie treningowe.

Revision #8
Created 2026-10-10 13:47:42 UTC by Tomasz Konieczny
Updated 2026-10-26 14:25:27 UTC by Tomasz Konieczny