

Automatyzacja Ansible w świecie AI.

Dariusz Puchalak

16 września 2025

- ▶ 30 lat Linux/Unix Sysadmin
- ▶ 18 lat trener
- ▶ 10+ lat w OSEC
- ▶ 10+ lat z Ansible

- ▶ Od 2009 na rynku
- ▶ doświadczona kadra
- ▶ specjalizacja open-source
- ▶ subskrypcje, szkolenia, konsultacje

Ścieżka edukacyjna Ansible.

- ▶ Lista aktualnych kursów, egzaminów:
<https://osec.pl/red-hat-ansible-automation/>
- ▶ RH294 - RHCE - Podstawowy kurs Ansible
- ▶ DO417 - Ansible dla inżynierów Windows (alternatywa dla RH294)
- ▶ DO457 - Ansible dla inżynierów sieciowych (alternatywa dla RH294)
- ▶ RH358 - „Dawne RH300 @RHEL7”, czyli usługi Linuksowe ręcznie i Ansiblem
- ▶ DO374 - „Dawne Advanced Ansible”, czyli zaawansowane elementy Ansible (kontynuacja RH294)
- ▶ DO467 - „Dawny Tower”, czyli jak używać Ansible Automation Platform
- ▶ DO274 - Introduction to Event-Driven Ansible (w RHAAP)

Spis treści.

Wprowadzenie do automatyzacji.

Ansible.

Pytania?

Automatyzacja.

„Automatyzacja – znaczne ograniczenie lub zastąpienie (proces zastępowania) ludzkiej pracy fizycznej i umysłowej przez pracę maszyn działających na zasadzie samoregulacji i wykonujących określone czynności bez udziału człowieka (czyli samoczynnych).” - Wikipedia

Cloud computing.

Cloud computing is "a paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand," - ISO/IEC 22123-1:2023(E) - Information technology - Cloud computing - Part 1: Vocabulary

AI.

Artificial intelligence (AI) is the capability of computational systems to perform tasks typically associated with human intelligence, such as learning, reasoning, problem-solving, perception, and decision-making. - Wikipedia

Automatyzacja w IT.

- ▶ IT process automation (runbook automation). - to co robi Ansible.
- ▶ Robotic process automation (RPA).

Infrastructure as code

Infrastructure as code (IaC) is the process of managing and provisioning computer data center resources through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. The IT infrastructure managed by this process comprises both physical equipment, such as bare-metal servers, as well as virtual machines, and associated configuration resources. The definitions may be in a version control system, rather than maintaining the code through manual processes. The code in the definition files may use either scripts or declarative definitions, but IaC more often employs declarative approaches. Wikipedia

Automatyzacja w IT - skalowanie.

- ▶ Microsoft's Desired State Configuration (DSC) is a declarative configuration platform.
- ▶ OpenStack Heat is a service to orchestrate composite cloud applications using a declarative template format through an OpenStack-native REST API.
- ▶ Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
- ▶ Ansible - Declarative approach to achieving a task that uses a description of the final state rather than a description of the sequence of steps necessary to achieve that state.

Wymagania do automatyzacji „infrastruktury”.

- ▶ Łatwość rozumienia i modyfikacji przez człowieka.
- ▶ Możliwość cofnięcia się w czasie, do poprzednich „poprawnych” konfiguracji. - czyli wersjonowanie.
- ▶ Systemy IT powinny być w oczekiwanym przez nas stanie (zachowywać się w oczekiwany sposób).
- ▶ Powtarzalność - eliminacja błędów ludzkich.

Problemy z AI.

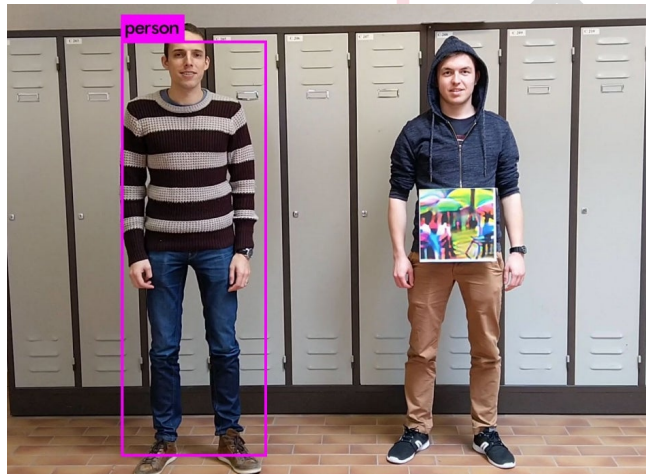
Donald Trump: Hi siri, how many miles did I ran today?

Siri: Ok, sending missiles to [#Iran](#) today



Rysunek: Satyryczne pokazanie problemów z AI.

AI - Artificial Ignorance?



Rysunek: AI nie widzi człowieka.

Problemy ze złym wykorzystaniem AI.

- ▶ „My voice is my passport” - nie rozumienie technologii i jej ograniczeń.
- ▶ Halucynacje.
- ▶ Inny sposób „myślenia” niż ludzie.
- ▶ Instrukcje dla modelu AI w danych zewnętrznych.
- ▶ FSD Tesli.
- ▶
- ▶ Odpowiedni rodzaj/model do odpowiedniego celu.

AI a automatyzacja infrastruktury.

- ▶ Dobrze dobrane do zadania rozwiązanie AI potrafi być równie dobre co człowiek, a nawet wielu z nas przewyższać.
- ▶ AI nie zastąpi „tradycyjnego” podejścia deklaratywnego.
- ▶ AI świetnie wspomogę procesy tworzenia, testowania automatyzacji.
- ▶ AI może świetnie wspomóc analizę błędów.

Cechy i zalety Ansible.

- ▶ Automatyzuje praktycznie wszystko.
- ▶ Agentless - korzysta z natywnych mechanizmów:
 - ▶ SSH + Python
 - ▶ WinRM + PowerShell
 - ▶ API
- ▶ Łatwa integracja z innymi systemami.
- ▶ Elastyczna automatyzacja.
- ▶ Idempotentność.

- ▶ Deklarujemy oczekiwany stan zarządzanych systemów.
- ▶ W sposób zoptymalizowany do czytania (pliki tekstowe).
- ▶ W bardzo prosty sposób.
- ▶ Automatyzujemy Linuksy, Windowsy, urządzenia sieciowe, firewalle, bazy danych, wirtualizacje, cloud, ...
- ▶ Ansible wykonując playbook (nasz „kod”) doprowadza systemy do oczekiwanego stanu (wykonując zmiany tylko gdy są potrzebne - np. restart usługi tylko po zmianie jej konfiguracji).

Architektura Ansible.

- ▶ Control nodes - tutaj uruchamiamy Ansible.
- ▶ Managed nodes - te systemu automatyzujemy.
- ▶ Inventory - określa co automatyzujemy:
 - ▶ statyczne,
 - ▶ dynamiczne (pobierające informacje o systemach z zew. źródła).
- ▶ Task - kawałek kodu z parametrami.
- ▶ Playbook - zbiór tasków uruchamianych na hostach z inventory.

Moduły - deklarowanie stanu.

- name: User 'johnd' with a specific uid and a group created.
ansible.builtin.user:
 name: johnd
 comment: John Doe
 uid: 1040
 group: admin
- name: User 'james' removed.
ansible.builtin.user:
 name: james
 state: absent

Przykładowe dane (zmienne).

default_password: Pa\$\$w0rd

users:

- fullname: Joe Test
name: test1
groups:
 - test1
 - test2description: Test user 1
- fullname: Joe Doe
name: test2
groups:
 - test2
 - adminsdescription: Test user 2

Deklarowanie stanu ze zmiennych.

```
- name: Local Windows users created.  
  win_user:  
    account_disabled: no  
    account_locked: no  
    name: "{{ item.name }}"  
    fullname: "{{ item.fullname | default(omit) }}"  
    description: "{{ item.description | default(omit) }}"  
    groups: "{{ item.groups | join(',') }}"  
    password: "{{ item.password | default(default_password) }}"  
    groups_action: add  
    password_expired: no  
    password_never_expires: yes  
    state: present  
    update_password: on_create  
    user_cannot_change_password: yes  
  with_items: "{{ users }}"
```

Sterowanie kodu zmiennymi - zalety i cechy.

- ▶ Idempotentność – właściwość pewnych operacji, która pozwala na ich wielokrotne stosowanie bez zmiany wyniku.
- ▶ W idealnym środowisku można sterować całą infrastrukturą i procesami automatyzacji tylko poprzez zmienne.
- ▶ Infrastructure as code
- ▶ Łatwe wersjonowanie (bo wszystko w plikach tekstowych).
- ▶ Czytelne dla człowieka (nie wymaga programowania) i maszyny - pliki YAML.
- ▶ Prostota - gdy się task „wywali” dla danego hosta, to już więcej nic dla niego nie wykonujemy.

Zmienne.

- ▶ variable precedence == „nadpisywanie zmiennych ogólnych przez bardziej szczegółowe”
- ▶ https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_variables.html#understanding-variable-precedence

Proces wprowadzenia nowego heterogenicznego systemu.

W idealnym środowisku można sterować całą infrastrukturą i procesami automatyzacji tylko poprzez zmienne. IT dzieli się robotą:

- ▶ Sieciowcy „ogarniają” konfigurację sieci.
- ▶ Bezpieczniaki - firewalle, WAFy, itp.
- ▶ Systemowcy odpowiednio Linuksy/Windowsy.
- ▶ Bazodanowcy - bazę danych.
- ▶ Operatorzy Clouda - konfiguruja to co ma trafić do clouda.
- ▶ Typowe rzeczy jak monitoring, backup z reguły nie wymagają (zbytnej) dodatkowej konfiguracji.
- ▶ Inne według potrzeb.

Jak mamy Ansible podpięte pod WebUI (AWX, RHAAP, Rundeck) to dowolna osoba z uprawnieniami może uruchomić Ansible który to po prostu zautomatyzuje.

Można powtarzać wielokrotnie - Ansible wprowadza zmiany tylko gdy musi.

Dalsze kroki.

Mając środowisko testowe uruchamiamy kolejne playbooki:

- ▶ testy akceptacyjne funkcjonalne
- ▶ testy akceptacyjne nie funkcjonalne jak:
 - ▶ bezpieczeństwo
 - ▶ wydajność
- ▶ Te same playbooki służą do uruchomienia w QA.
- ▶ Te same playbooki służą do uruchomienia produkcyjnego.
- ▶ Łatwo można zrobić z tego pipeline CI/CD.
- ▶ Mamy od razu większą część do planów Disaster Recovery.
- ▶ Kolejne systemy wykorzystują to co już mamy wdrożone i wytestowane.
- ▶ Nowe wersje wykorzystują to co już mamy (czasem po wprowadzeniu niewielkich modyfikacji).

Self service.

Jak mamy Ansible podpięte pod WebUI (AWX, RHAAP, Rundeck) to dowolna osoba z uprawnieniami może uruchomić Ansible który to po prostu zautomatyzuje dany proces.

- ▶ 1sza linia wsparcia może uruchomić zdefiniowaną wcześniej diagnostykę (nie musi posiadać kont na diagnozowanych systemach)
- ▶ 1sza linia wsparcia może uruchomić zdefiniowane akcje naprawcze.
- ▶ Pełna rozliczalność wykonywanych akcji.
- ▶ „Bezpieczniaki” mogą uruchomić swoje playbooki sprawdzające stan zgodności systemów IT z wymaganiami, regulacjami.
- ▶ Można stworzyć playbooki realizujące techniczna część wielu procedur.

Więcej informacji do RH AAP (Red Hat Ansible Automation Platform):

- ▶ Youtube
- ▶ Prezentacja

Składamy z klocków.

- ▶ Moduły - 9235
- ▶ Role - 140
- ▶ Plug-iny:
 - ▶ filter - 257
 - ▶ lookup - 119
 - ▶ test - 111
- ▶ Kolekcje: 91 <https://docs.ansible.com/ansible/latest/collections/index.html#list-of-collections>

Gotowce do Ansible.

- ▶ RH Automation Hub

<https://console.redhat.com/ansible/ansible-dashboard>:

- ▶ Partnerzy - 86
- ▶ Kolekcje - 176

- ▶ Ansible Galaxy (publiczna biblioteka) <https://galaxy.ansible.com/ui/>

- ▶ Kolekcje - 3687
- ▶ Role - 36328

Automatyzacja Linuksów.

- ▶ RedHat - <https://linux-system-roles.github.io/>
- ▶ Debian/Ubuntu - <https://docs.debops.org/>

Push automation..

Tradycyjne użycie Ansible (ansible-playbook, ansible, ansible-navigator).

Pull automation.

ansible-pull - odwrócenie metody push. Zaciągany jest playbook z GITa i uruchamiany lokalnie. Stosowane najczęściej:

- ▶ dla systemów które bywają wyłączone, niedostępne czasowo,
- ▶ w celu zwiększenie „efektywności” automatyzacji,
- ▶ stosowane tam gdzie nie ma zależności z innymi systemami.

REPL automation.

REPL console for executing Ansible tasks.

REPL - read-eval-print loop.

Interaktywna automatyzacja systemów - stosowana sporadycznie.

Event Driven Automation.

ansible-rulebook - automatyzacja uruchamiana zdarzeniowo.

Pozwala wykonać określone akcje w reakcji na zdefiniowane zdarzenie/a zgodnie z filozofią Ansible.

Więcej informacji:

- ▶ Youtube
- ▶ Prezentacja

Filtry

- ▶ Jak uniksowy mechanizm „pipe”
- ▶ Używane przede wszystkim do zmiany danych.
- ▶ `ansible-doc -t filter -l`
- ▶ https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_filters.html

Więcej informacji:

- ▶ Youtube
- ▶ Prezentacja

Testy

- ▶ Używane przede wszystkim do porównań.
- ▶ Testy dot. plików/katalogów działają tylko na kontrolerze
- ▶ `ansible-doc -t test -l`
- ▶ https://docs.ansible.com/ansible/latest/playbook_guide/playbooks_tests.html

Więcej informacji:

- ▶ Youtube
- ▶ Prezentacja

Lookup/query

- ▶ Dają dla dostęp do danych pobieranych z zewnątrz
- ▶ `ansible-doc -t lookup -l`
- ▶ lookup i query zwracają te same dane, ale w innym formacie.
- ▶ lookup zwraca łańcuch danych rozdzielonych przecinkami
- ▶ query zwraca listę

Więcej informacji:

- ▶ Youtube
- ▶ Prezentacja

Przykłady.

- ▶ `https://dariusz.puchalak.net/Ansible/ssh-config-files.yml`
- ▶ `https://dariusz.puchalak.net/Ansible/versions.yml`
- ▶ `https://dariusz.puchalak.net/Ansible/Windows_reboot.yml`
- ▶ `https://dariusz.puchalak.net/Ansible/environment.yml`
- ▶ `https://dariusz.puchalak.net/Ansible/OSEC_Forum_2023/ddns_update.yml`
- ▶ `ansible-doc -t filter urlsplit`
- ▶ `ansible-doc cisco.ios.ios_l3_interfaces`
- ▶ `Ansible-demo/vmware-configure.yml`

Pytania?

Dariusz.Puchalak@osec.pl

Dziękuję.
