



Wprowadzenie do DevOps i zarządzania systemami rozproszonymi

mgr inż. Jakub Woźniak

Zarządzanie Systemami Rozproszonymi
Instytut Informatyki
Politechnika Poznańska



Plan wykładu

- Czym jest DevOps?
- DevOps vs. Agile, SRE i Platform Engineering
- System rozproszony w praktyce
- Implementacja DevOps
- Configuration Management as Code
- Ansible - wprowadzenie, model, historia
- Ansible vs Chef, Puppet
- Przykłady playbooków do konfiguracji



Czym jest DevOps?

Definicja: Zbiór praktyk, mających na celu usunięcie przeszkód komunikacyjnych pomiędzy zespołami developerskimi i operacyjnymi.

Cel: Zautomatyzowanie procesów wdrożeniowych, testów, monitorowania.

Ważne daty

2007: Konferencja Agile – Patrick Debois inicjuje dyskusję.

2009: DevOps Days – początek ruchu.



DevOps vs Agile, SRE i Platform Engineering

- **Agile** – podejście do zarządzania projektami promujące iteracyjny sposób zarządzania wymaganiami.
- **SRE** - podejście do zarządzania infrastrukturą i operacjami przez migrację praktyk z wytwarzania oprogramowania
- **Platform Engineering** - trend wskazujący na potrzebę wytworzenia platformy developerskiej integrującej ekosystem narzędzi DevOps



System rozproszony w praktyce

- Skalowalność: Zwiększanie mocy obliczeniowej.
- Dostępność: Redundancja, zapobieganie awariom.
- Wydajność: Rozpraszanie obciążeń.
- Przykłady: Netflix, Amazon Web Services.

System rozproszony jest naturalnym krokiem ewolucyjnym przy skalowaniu aplikacji. System rozproszony wprowadza złożoność, która jest poważnym wyzwaniem przy utrzymaniu stabilności ciągle rozwijającej się aplikacji.



Implementacja DevOps

DevOps promuje ideę wspólnych narzędzi dla różnych ról. Duży nacisk kładziemy na automatyzację procesów i obserwowalność.

- CI/CD: automatyczne testowanie i wdrażanie, automatyzacja procesów wsadowych
- Testy: jednostkowe, integracyjne, end to end, wydajnościowe
- Monitoring: stała kontrola wydajności i stabilności.



Configuration Management as Code

Mnogość maszyn do utrzymania stanowi następujące problem:

- spójność konfiguracji
- czas potrzeby na wdrożenie
- błędy przy wprowadzaniu konfiguracji

Zarządzanie konfiguracją jako kod pozwala na uniknięcie tych problemów, przez deklaracyjny opis konfiguracji i wdrożenie jej przy pomocy narzędzia, które interpretuje konfiguracje i dokonuje rekonceyliacji stanów maszyn.



Ansible – wprowadzenie, model, historia

Ansible to narzędzie wytworzone w 2012 roku i rozwijane współcześnie przez firmę Red Hat.

Ansible używa języka YAML do opisu konfiguracji (nazywane Playbookiem), działa w modelu bezagentowym (wszystko dzieje się z maszyny zarządzającej po SSH lub WinRM), do działania wymaga interpretera języka Python.

Ansible jest udostępnione na licencji Open Source.



Ansible vs Chef, Puppet

- Ansible: Agentless, YAML.
- Chef: Agent-based, Ruby.
- Puppet: Deklaratywny, agent-based.

Jak wygląda przykładowa konfiguracja Ansible?

```
---
- name: Intro to Ansible Playbooks
  hosts: all

  tasks:
  - name: Copy file hosts with permissions
    ansible.builtin.copy:
      src: ./hosts
      dest: /tmp/hosts_backup
      mode: '0644'
  - name: Add the user 'bob'
    ansible.builtin.user:
      name: bob
      become: yes
      become_method: sudo
  - name: Upgrade all apt packages
    apt:
      force_apt_get: yes
      upgrade: dist
      become: yes
```

- Name – nazwa playbooka
- Hosts – grupa hostów do zarządzania
- Tasks – lista zadań
 - Name: nazwa zadania
 - Nazwa modułu
 - Parametry modułu

źródło: <https://spacelift.io/blog/ansible-tutorial>