Ansible
Infrastructure as Code

Cześć!

Prowadzący: Jakub Andrzejewski

DevOps Engineer @ El Passion

Autor prezentacji: Daniel Kossakowski

1.
Ansible? A co to?

Ansible? Infrastructure as Code

Declarative:

- Definiujemy cel
- Co chcemy osiągnąć
- Stanowość
- Wycofywanie zmian

- Terraform

Imperative:

- Definiujemy sposób
- Jak chcemy coś osiągnąć
- Brak stanu
- Zmiany nie są wycofywane

- **Ansible**

Ansible? A co to?

- Narzędzie do zarządzania infrastrukturą
- Zarządzanie konfiguracją oprogramowania
- Zarządzanie konfiguracją systemów
- Ansible łączy się po protokole SSH

Ansible? A dlaczego?

- Dokumentacja
- Agentless
- Minimum zależności: python (3.x)
- Dużo modułów
- Automatyzacja

2.
Instalacja Ansible

Ansible? Instalacja

- RHEL, CentOS, Fedora

```
$ sudo yum install epel-release
$ sudo yum install ansible
```

- Ubuntu, Debian

```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```

3.
Inventory

Inventory? A co to?

- Ansible musi wiedzieć czym zarządzać
- Plik z definicją serwerów
- Adresy IP lub DNS
- Numery portów
- Nazwy użytkowników
- Grupy maszyn
- Dodatkowe parametry

Inventory? Przykład

```
[targets]
localhost                ansible_connection=local
other1.example.com       ansible_connection=ssh       ansible_user=ubuntu
other2.example.com       ansible_connection=ssh       ansible_user=ec2-user
```

Inventory? Przykład

```
[mail-server]
mail1.example.com    ansible_connection=ssh    ansible_user=ubuntu
mail2.example.com    ansible_connection=ssh    ansible_user=ubuntu

[web-server]
web1.example.com     ansible_connection=ssh    ansible_user=ubuntu
web2.example.com     ansible_connection=ssh    ansible_user=ubuntu
```

Inventory? Przykład

```
[all:vars]
ansible_user=ubuntu

[mail-server]
mail1.example.com
mail2.example.com

[web-server]
web1.example.com
web2.example.com
```

Inventory? Hasło?

- Można podać hasło
- Ale haseł nie trzymamy w repo!
- Lepiej użyć klucze SSH
- Publiczny klucz dodajemy do ~/.ssh/authorized_keys na serwerach
- Prywatny klucz trzymamy na maszynie zarządzającej

Inventory? Przykład

```
[all:vars]
ansible_user=ubuntu
ansible_ssh_private_key_file=/home/user/.ssh/id_rsa

[mail-server]
mail1.example.com
mail2.example.com

[web-server]
web1.example.com
web2.example.com
```

Inventory? Lokalizacja

- Globalnie:
  - /etc/ansible
- Lokalnie:
  - Dowolny folder, w którym uruchamiamy ansible
  - Zwyczajowa nazwa: hosts

- Grupa *all* zawiera wszystkie hosty, niezależnie od przypisania do grup

4.
Proste użycie

Proste użycie? Komenda

- Komenda do uruchomienia: ansible
- Polecenia ad-hoc

```
$ ansible -m ping -i hosts all
```

- Parametry:
  - -m ping - nazwa uruchomionego modułu (tu ping)
  - -i hosts - ścieżka do pliku z inventory
  - all - określenie grupy hostów z inventory

Proste użycie? Komenda

- Ten sam ping, ale tylko na dwóch serwerach

```
$ ansible -m ping -i hosts mail-server
```

Proste użycie? Komenda

```
mail1.example.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
mail2.example.com | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Proste użycie? Komenda

```
$ ansible -a "/sbin/reboot" -i hosts all
```

- Parametry:
  - -a “...” - argumenty przekazywane do modułu
  - -i hosts - ścieżka do pliku z inventory
  - all - określenie grupy hostów z inventory
- A jaki moduł będzie użyty?

Proste użycie? Komenda

- Moduł domyślny: ansible.builtin.command
- Dokumentacja: https://docs.ansible.com/ansible/latest/collections/ansible/builtin/command_module.html#command-module
- Taki sam efekt:

```
$ ansible -a "/sbin/reboot" -i hosts all
$ ansible -m ansible.builtin.command -a "/sbin/reboot" -i hosts all
```

5.
Moduły

Moduły? A co to?

- Zbiór funkcji, które “umie” ansible
- Każdy moduł przyjmuje parametry
- Większość modułów ma ograniczenia wywołania z linii komend

- Lista modułów: https://docs.ansible.com/ansible/2.8/modules/list_of_all_modules.html

Moduły? Przykłady

- ansible.builtin.command - uruchamianie komendy na serwerze
- ansible.builtin.copy - kopiowanie pliku lokalnego do serwera
- ansible.builtin.file - zarządzanie plikami na serwerze
- ansible.builtin.lineinfile - zapisywanie jednej linii w pliku
- ansible.builtin.package - zarządzanie zainstalowanymi paczkami
- ansible.builtin.service - zarządzanie usługami (daemons)

6.
Playbook

Playbook? A co to?

- To jest prawdziwe zarządzanie
- Skrypty automatyzujące całą konfiguracje
- Wiele wywołań modułów jedną komendą
- Playbook może być jednym plikiem
- Playbook może być zbiorem reguł i wielu plików
- Przykład:
  - Konfiguracja e-mail
  - Konfiguracja httpd
  - Konfiguracja aplikacji

Playbook? Przykład

```
$ cat playbook.yaml
- hosts: all
  become: yes
  tasks:
    - name: Install nginx
      apt:
        name: nginx
        state: present
        update_cache: true

$ ansible-playbook -i hosts playbook.yaml
```

Playbook? Przykład

```
PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.0.1]

TASK [Install nginx] ***********************************************************
ok: [192.168.0.1]

PLAY RECAP *********************************************************************

192.168.0.1          : ok=1 changed=1  unreachable=0 failed=0
```

Playbook? Przykład

```
PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.0.1]

TASK [Install nginx] ***********************************************************
ok: [192.168.0.1]

PLAY RECAP *********************************************************************

192.168.0.1          : ok=2 changed=0  unreachable=0 failed=0
```

Playbook? A co to?

- ansible-playbook uruchamia playbook
- Hosty określone są w playbook
  - W sekcji *hosts* podaje się nazwę grupy w pliku hosts
  - Wpis *all* użyje wszystkie hosty we wszystkich grupach
- Sekcja *become* określa, czy ansible może się przełączyć na root'a
- Sekcja *tasks* zawiera wywołania modułów ansible
  - apt jest odpowiednikiem modułu *ansible.builtin.apt*
  - dla modułów *ansible.builtin.apt* nie trzeba prefixu *ansible.builtin*

Playbook? A co to?

- Przed uruchomieniem taska ansible sprawdza jego status
- Przy drugim uruchomieniu *nginx* nie jest już instalowany
- Takich tasków można utworzyć dużo więcej
- Ale wiele tasków w jednym pliku nie będzie czytelne
- Warto dzielić na pliki (role)

Playbook? Struktura plików

- playbook-webservers.yaml
- playbook-mailservers.yaml
- roles/
  - role-name/ - nazwa roli
    - tasks/ - folder na taski (domyślny: main.yaml)
    - handlers/ - taski odpalane w po zmianach (triggerowane z tasków)
    - files/ - pliki do skopiowania na serwery
    - templates/ - szablony generowanych plików
    - vars/ - pliki ze zmiennymi

Playbook? Role

- Playbook może wywoływać różne role
- Role to zbiór tasków
- Rola realizuje zbliżone cele
- Role mogą być wywoływane tylko dla określonych grup hostów

Playbook? Role

```
- name: apply common configuration to all nodes
  remote_user: ubuntu
  become: true
  become_user: root
  hosts: all
  roles:
        - os-base
        - os-security
```

Playbook? Role

- remote_user: ubuntu - nazwa użytkownika, na którego ansible się loguje
- become: true - zwiększenie uprawnień do roota
- become_user: root
- hosts: all - lista hostów
- roles - lista ról do uruchomienia

ĆWICZENIE!

- Utworzenie maszyn EC2 bazując na dostarczonym kodzie
- Utworzenie pliku hosts z publicznymi adresami IP maszyn
- Utworzenie roli *common* z podstawowymi ustawieniami serwerów
- Utworzenie roli *web-server* do instalacji apache i MySQL

7.
Zmienne

Zmienne? A co to?

- Zmienne możemy przekazywać do playbook'ów (sekcja vars)
- Zmienne możemy przekazywać do ról
- Parametryzacja kodu (np. per rodzaj środowiska)
- Czytelniejszy kod, parametry w jednym miejscu

- Dokumentacja: https://docs.ansible.com/ansible/2.3/playbooks_variables.html

Zmienne? Użycie

- Zmienne możemy umieścić w:
  - pliku hosts
  - playbooku
  - w folderze roli w katalogu vars
  - z linii komend (parametr --extra-vars)
  - folder: group_vars
  - folder: host_vars

Zmienne? Dostęp do zmiennych

- Dostęp do zmiennych: {{ variable_name }}

```
- hosts: all
  become: yes
  vars:
    package_name: nginx
  tasks:
    - name: Install nginx
      apt:
        name: {{ package_name }}
        state: present
```

Zmienne? Dostęp do zmiennych

```
---
# file: /etc/ansible/group_vars/all
# this is the site wide default
ntp_server: default-time.example.com

---
# file: /etc/ansible/group_vars/boston
ntp_server: boston-time.example.com

---
# file: /etc/ansible/host_vars/xyz.boston.example.com
ntp_server: override.example.com
```

8.
Pętle

Pętle? Użycie

- Powtarzalne taski definiowane jednym kodem
- Ansible używa list do pętli
- Źródła list:
  - zmienne
  - bezpośrednio w kodzie
  - z innych tasków

Pętle? Lista w kodzie

```
- name: add several users
  user:
    name: "{{ item }}"
    state: present
    groups: "wheel"
  with_items:
     - testuser1
     - testuser2
```

Pętle? Lista w zmiennej

```
- name: add several users
  user:
    name: "{{ item }}"
    state: present
    groups: "wheel"
  with_items: {{ users }}
```

Pętle? Lista z tasków

```
- hosts: all
  tasks:
    - name: Ansible register with_items example
      shell: "find *.txt"
      args:
        chdir: "/home/user"
    register: with_output

    - shell: "cp {{ item }} {{item}}_bkp"
      with_items:
        - "{{ with_output.stdout_lines }}"
```

Koniec!

Dziękuję za uwagę

Pytania?