

Know Your tools: Event-Driven Ansible

Dariusz Puchalak

11 czerwca 2024

- ▶ 25+ lat Linux/Unix Sysadmin
- ▶ 15+ lat trener
- ▶ 10+ lat w OSEC
- ▶ 10+ lat z Ansible

- ▶ Od 2009 na rynku
- ▶ doświadczona kadra (ACNI, RHCA)
- ▶ 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.

Sposoby automatyzacji w Ansible.

Pytania?

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 „efetywności” 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.

Zalety Event Driven Automation.

- ▶ poprawia wydajność (działa tylko, kiedy jest taka potrzeba),
- ▶ poprawia bezpieczeństwo (pozwala zareagować zaraz po wykryciu problemu/incydentu),
- ▶ poprawia jakość (reakcja zaraz po zdarzeniu),
- ▶ obniża koszt (automatyzacja pracy ręcznej).

Przykłady użycia.

- ▶ automatyczne zbieranie danych do obsługi zdarzenia,
- ▶ automatyczne restarty aplikacji/systemów,
- ▶ orkiestracja w reakcji na zdarzenie,
- ▶ automatyzacja procesów DevOpsowych (GitOps).

Części składowe rulebooka.

- ▶ Sources - skąd pobieramy zdarzenia
- ▶ Rules - określamy warunki
- ▶ Actions - definiujemy akcje

Sources - collection ansible.eda.

- ▶ `alertmanager.py` - webhook from alertmanager or alike system
- ▶ `aws_cloudtrail.py` - events from an AWS CloudTrail
- ▶ `aws_sqs_queue.py` - AWS SQS queue
- ▶ `azure_service_bus.py` - Azure service bus
- ▶ `file.py` - loading facts from YAML files (initially and when the file changes)
- ▶ `file_watch.py` - watching file system changes
- ▶ `generic.py` - allows you to insert custom data

Sources - collection ansible.eda.

- ▶ `journald.py` - events from systemd journald logs
- ▶ `kafka.py` - events via a kafka topic
- ▶ `pg_listener.py` - PostgreSQL LISTEN and NOTIFY
- ▶ `range.py` - generating events with an increasing index `i`
- ▶ `tick.py` - generating events with an increasing index `i` that never ends
- ▶ `url_check.py` - polls a set of URLs and sends events with their status.
- ▶ `webhook.py` - receiving events via a webhook (must be valid JSON object).

Rules

lista reguł. Reguła składa się min. z:

- ▶ condition - definicja spełnienia warunku uruchomienia akcji
- ▶ action/actions - akcja/akcje

Conditions

lista warunków do spełnienia w celu uruchomienia określonej akcji

Actions.

- ▶ Definicja czynności do wykonania w przypadku spełnienia warunków
 - ▶ run_playbook
 - ▶ run_module
 - ▶ run_job_template
 - ▶ run_workflow_template
 - ▶ set_fact
 - ▶ post_event
 - ▶ retract_fact
 - ▶ print_event
 - ▶ shutdown
 - ▶ debug
 - ▶ none

Przykładowy rulebook

- name: Listen for events on a webhook

hosts: all

sources:

- ansible.eda.webhook:
 - host: 0.0.0.0
 - port: 5000

rules:

- name: Say Hello
 - condition: event.payload.message == "Test"

action:

- run_playbook:
 - name: play1.yml

DEMO

DEMO

OSEC

Dariusz Puchalak

Rulesets

- ▶ name - uniq!
- ▶ sources
- ▶ rules
- ▶ hosts
- ▶ [gather_facts]
- ▶ [default_events_ttl] - ile trzymamy częściowo spełnione warunki
- ▶ execution_strategy - default sequential, lub parallel

Rules

- ▶ name - uniq!
- ▶ condition
- ▶ action/actions
- ▶ [enabled]

Conditions

- ▶ name - uniq!
- ▶ condition
- ▶ action/actions
- ▶ [enabled]

Conditions

typy danych:

- ▶ int
- ▶ string
- ▶ bool
- ▶ float
- ▶ null

Conditions

Operatory porównań:

- ▶ `==`, `!=`, `>`, `>`, `>=`, `<=`,
- ▶ `+`, `-`, `*`,
- ▶ `and`, `or`, `not`, `in`, `not in`, `contains`, `not contains`
- ▶ `is defined`, `is not defined`
- ▶ `is match ....`
- ▶ `is search`, `is regex`, `is select`, `is selectattr`

Conditions

Wiele warunków:

- ▶ all - wiele zdarzeń
- ▶ and - jedno zdarzenie
- ▶ any - dowolne

Pytania?

Dariusz.Puchalak@osec.pl

Dziękuję.
