AWX i Semaphore

Poziom: operacyjny
Dotyczy: AWX (upstream Ansible Automation Platform), Semaphore UI **Czas: ~25 min**

Żadne z tych dwóch narzędzi nie zastępuje Ansible — oba są interfejsem i API postawionym nad tym samym silnikiem, żeby uruchamianie playbooków dało się oddać zespołowi bez konta SSH na każdej maszynie i bez konieczności znajomości linii poleceń.

Skróty użyte na tej stronie (7)

Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
AWX	otwarta wersja Ansible Tower: przeglądarkowy panel do uruchamiania Ansible
GET	Metoda HTTP, którą prosimy serwer o odczyt zasobu bez zmieniania czegokolwiek, np. o wartość wilgotności z czujnika
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
REST	sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera
WMS	Warehouse Management System: system zarządzania magazynem

Z życia wzięte

Z ŻYCIA WZIĘTE

Marek z magazynu prosi codziennie o to samo: żeby ktoś z IT ręcznie uruchomił skrypt czyszczący stare pliki tymczasowe z WMS na `ora01`, zanim zacznie się zmiana o 6:00. Zamiast pamiętać o tym co rano, Ola zakłada w AWX szablon zadania z harmonogramem, do którego Marek ma dostęp tylko do podglądu historii uruchomień. Skrypt rusza sam, a IT dowiaduje się o nim tylko wtedy, gdy coś pójdzie nie tak.

Czym są AWX i Semaphore

Pojęcie: AWX. Otwartoźródłowy projekt będący podstawą komercyjnej Ansible Automation Platform — webowy interfejs i REST API postawione nad Ansible, dodające to, czego samemu `ansible-playbook` z linii poleceń brakuje w środowisku zespołowym: kto może uruchomić który playbook, na jakich maszynach, z jakim harmonogramem, i pełną historię tego, co się wykonało, kiedy i z jakim wynikiem.



Uzytkownik klika szablon zadania w AWX albo Semaphore, interfejs uruchamia playbook silnikiem Ansible, a ten łączy się po SSH z maszynami docelowymi (kliknięcie otwiera pełny rozmiar)

Pojęcie: Semaphore. Lżejsza alternatywa o podobnym celu — również webowy interfejs i API nad Ansible (a dodatkowo nad Terraform), z mniejszymi wymaganiami sprzętowymi i prostszą instalacją niż pełny AWX, kosztem części zaawansowanych funkcji zarządzania uprawnieniami i integracji dostępnych w AWX.

Oba narzędzia operują na tych samych czterech pojęciach, tylko nazwanych czasem nieco inaczej w interfejsie.

Pojęcie: projekt. Powiązanie z repozytorium kodu (Git), z którego AWX albo Semaphore pobiera playbooki, role i pliki pomocnicze przed każdym uruchomieniem — dzięki temu treść automatyzacji żyje w tym samym repozytorium, co reszta infrastruktury jako kod, a nie jest wklejana ręcznie do interfejsu.

Pojęcie: inwentarz (w kontekście AWX/Semaphore). Ten sam koncepcyjnie inwentarz co w czystym Ansible (patrz Ansible: podstawy operacyjne), tylko utrzymywany i edytowany przez UI zamiast pliku tekstowego na dysku administratora — może być statyczny, wklejony wprost, albo dynamiczny, podłączony do zewnętrznego źródła (np. API dostawcy chmury). Maszyny bywają pogrupowane funkcjonalnie, na przykład w grupę operacyjną skupiającą serwery, którymi zajmuje się dany zespół.

Pojęcie: poświadczenia (w kontekście AWX/Semaphore). Magazyn kluczy SSH, haseł i tokenów API, analogiczny do magazynu poświadczeń Jenkinsa (patrz Poświadczenia i bezpieczeństwo) — playbook odwołuje się do poświadczenia przez jego nazwę, nigdy nie

mając bezpośredniego dostępu do jego wartości, a użytkownik uruchamiający zadanie może w ogóle nie mieć uprawnień do podejrzenia samego klucza.

Pojęcie: szablon zadania (job template). Zapisana konfiguracja gotowa do uruchomienia — konkretny playbook z konkretnego projektu, uruchamiany na konkretnym inwentarzu, z konkretnym zestawem poświadczeń. To właśnie szablon zadania klika użytkownik, żeby wykonać automatyzację — nie musi przy tym znać ani treści playbooka, ani składni Ansible.

Pojęcie: harmonogram. Cykliczne uruchomienie szablonu zadania bez ręcznej interwencji — na przykład codzienne zadanie sprzątające stare pliki tymczasowe na grupie serwerów, uruchamiane automatycznie o ustalonej porze, zamiast pamiętania o nim ręcznie.

Różnice między AWX a Semaphore

Cecha	AWX	Semaphore
Silniki obsługiwane natywnie	Ansible	Ansible, Terraform, inne (przez wtyczki)
Model uprawnień	rozbudowany, wielopoziomowy (organizacje, zespoły, role)	prostszy, mniej granularny
Wymagania instalacyjne	pełna platforma (Kubernetes/kontenery, baza danych, kolejka zadań)	pojedynczy binarny proces albo lekki kontener
Typowe zastosowanie	duże środowisko z wieloma zespołami i restrykcyjnym rozdziałem uprawnień	mniejszy zespół, potrzeba szybkiego startu bez ciężkiej infrastruktury

Uruchomienie zadania z poziomu API

Zamiast klikać w interfejsie, szablon zadania da się uruchomić programowo, co jest przydatne przy integracji z innymi systemami (np. wywołanie z etapu pipeline'u Jenkinsa opisanego na stronie Pipeline: Jenkinsfile, etapy, artefakty).

Uruchomienie szablonu zadania AWX przez REST API

```
curl -s -X POST https://awx.zielona.example/api/v2/job_templates/12/launch/ \
  -H "Authorization: Bearer TokenApi" \
  -H "Content-Type: application/json" \
  -d '{"limit": "app01.ipa.zielona.example"}'
```

Przykład odpowiedzi:

```
{
  "job": 4821,
  "status": "pending",
  "job_template": 12
}
```

Odpowiedź zwraca identyfikator uruchomionego zadania (`job`), po którym da się później sprawdzić jego status i log wykonania osobnym zapytaniem GET do tego samego API.

Częste błędy

Nadawanie szerszych uprawnień inwentarzowi, niż wymaga zadanie. Grupa użytkowników z dostępem do uruchamiania zadań na całym inwentarzu produkcyjnym, gdy realnie potrzebuje dostępu tylko do jednej podgrupy serwerów, zwiększa skutki potencjalnej pomyłki przy wybieraniu celu uruchomienia.

Edytowanie playbooka bezpośrednio w interfejsie AWX/Semaphore zamiast w repozytorium projektu. Część instalacji na to pozwala technicznie, ale wtedy zmiana nie przechodzi code review i ginie przy najbliższej synchronizacji projektu z repozytorium, które nadpisuje lokalne zmiany.

Brak `--limit` (odpowiednika w polu „Limit" szablonu zadania) przy pierwszym uruchomieniu nowego playbooka. Tak samo jak przy gołym Ansible, testowanie od razu na całym inwentarzu jest ryzykowniejsze niż na jednej maszynie.

□□ Ćwiczenia

>

> Zadania sprawdzające.

1. Wyjaśniamy, co dokładnie dodają AWX i Semaphore ponad to, co daje sam `ansible-playbook` z linii poleceń.
2. Zespół ma dziesięć osób w kilku podzespołach i potrzebuje precyzyjnego rozdziału, kto może uruchamiać które szablony zadań na których inwentarzach. Wskazujemy, które z dwóch narzędzi lepiej pasuje do tego wymagania i uzasadniamy.
3. Opisujemy, co się stanie, jeśli ktoś zmieni treść playbooka bezpośrednio w interfejsie zamiast w repozytorium projektu.
4. Piszemy, jakie trzy elementy trzeba mieć skonfigurowane w AWX, zanim szablon zadania w ogóle da się uruchomić.

Powiązane: Ansible, AWX, Semaphore, Rudder, Terraform: co do czego · Ansible: podstawy operacyjne · Poświadczenia i bezpieczeństwo

---

Revision #36
Created 2026-09-25 07:40:06 UTC by Tomasz Konieczny
Updated 2026-10-26 14:13:30 UTC by Tomasz Konieczny