

ANSIBLE

- POZIOM ZAAWANSOWANY



Kategoria	Czas trwania	Termin	Cena
DevOps	3 dni	ustalamy indywidualnie	ustalamy indywidualnie

Program szkolenia:

Poniżej przedstawiamy przykładowy program szkolenia, który może zostać zmodyfikowany zgodnie z oczekiwaniami oraz poziomem grupy szkoleniowej. Przed przygotowaniem docelowego programu szkolenia, przeprowadzamy rozmowę techniczną, w której bierze udział trener oraz osoba techniczna lub cały zespół developerów reprezentujący klienta, w celu ustalenia szczegółów szkolenia.

Uwaga: zakładamy podstawową znajomość Ansible, YAML, Jinja, podstawy administracji systemami Linux, umiejętność pracy w terminalu (konsoli), edycji plików na serwerach zdalnych (np vim, lub Visual Studio Code z pluginem Remote Explorer), biegłe użycie ssh.

1. Omówienie systemów orkiestracji i Ansible

2. Przygotowanie środowiska do pracy z Ansible

- sposoby instalacji Ansible
- instalacja Ansible w 'sandbox', kwestie utrzymania środowiska przez wiele lat
- praca z modułami uruchamianymi z linii komend

3. Praca z Ansible

- praca z zmiennymi (variables) – znaczenie umiejscowienia zmiennych względem ich 'zasięgu' w projekcie (variable precedence)
- role i kolekcje

- 'Powtórzenie' przygotowanie "od zera do bohatera" deploymentu aplikacji
 - tworzenie playbooków, przygotowywanie szablonów (template) konfiguracji, połączenie w całość wdrożenia aplikacji z bazą danych i loadbalancerem
 - w jednym przebiegu
- warunkowe wykonywanie zadań (when)
- cykliczne wykonywanie zadań (loop)
- użycie tagów (lub dlaczego ich nie używam)
- parametryzacja zadań
- dostosowywanie konfiguracji per środowisko
- re-użycie już wytworzonych komponentów

■ 4. Debugging

- debugging pracy z Ansible (brakujące zmienne, błędy wykonania manifestów, błędy w zadaniach)
- debugging na poziomie kodu Ansible - "ręczne" zmiany w naszym orkiestratorze w kodzie Python, sposoby uruchomienia modułów jako kodu pythona
- opcja użycia Ansible jako biblioteki Python (bez wrapera ansible)
- testowanie ról z Ansible Molecule

■ 5. Ansible Galaxy

- omówienie repozytorium ról i kolekcji
- historia projektu Ansible, a kompatybilność wsteczna
- wersjonowanie ról i kolekcji

■ 6. Interfejs graficzny AWX/Tower

- koncepcja pracy w małej i dużej organizacji
 - omówienie modelu uprawnień RBAC
- przykłady uruchamiania naszego kodu IaaS za pomocą interfejsu graficznego (web)
- instalacja AWX
- konfiguracja AWX ręczna i automatyczna z poziomu kodu (SaaS)

■ 7. Koncepcje pracy z wieloma środowiskami

- dev, test, QA, UAT, PROD - jak to wszystko połączyć i utrzymać!
- rozwijanie i utrzymanie projektów na przestrzeni lat
- układ katalogów w projekcie
- konfiguracja Ansible
- kwestia lokalizacji zmiennych - gdzie je definiować aby było to elastyczne i wygodne rozwiązanie

■ 8. Inventory - dynamiczne, statyczne, hybrydowe

- układ inventory, podział na podfoldery
- dynamiczne inventory – przykłady użycia
- jak utrzymać inventory dla wielu projektów i środowisk

■ 9. Sekrety w IaasC

- Ansible-Vault i przykłady automatycznego ładowania sekretów
- koncepcja utrzymania sekretów w systemach zewnętrznych

■ 10. Ciekawostki i dobre praktyki

- lookupy
- delegowanie zadań
- powtarzanie wykonania zadań
- "tagowanie" zadań
- bloki i grupowanie zadań
- zarządzanie kolejnością wykonywania zadań, `run_once`, `pre_post_tasks`, `include_*`
- raporty wykonania, jak je "poprawiać", jak poprawnie analizować 'zmiany', znaczenie trybu `check-mode` (i potencjalne pułapki)
- "przyspieszanie Ansible", np `mitogen`, `pipelining`, itp
- "bezpieczne restarty" klastrów przy pomocy Ansible

■ 11. Orkiestracja systemu Windows (przykłady)

■ 12. Dodatkowo

- możliwość przeprowadzenia części laboratoriów w środowisku testowym (AWS lub Azure) klienta
- program może być dostosowany dynamicznie do potrzeb grupy/klienta