

Kod szkolenia: **PUPPET/ADV**

Tytuł szkolenia: **Zaawansowane techniki użycia Puppet**

Puppet Advanced

Dni: 2

Opis:

Adresaci szkolenia

Szkolenie adresowane jest do programistów i administratorów, którzy chcieliby uspołnić swoją wiedzę na temat języka i ekosystemu Puppet i poszerzyć ją o zaawansowane techniki użycia Puppeta i powiązanych technologii.

Cel szkolenia

Uczestnik uspołni posiadaną wiedzę na temat podstaw oprogramowania Puppet oraz rozszerzy ją o zaawansowane techniki programowania manifestów i mechanizmy oraz systemy związane z ekosystemem Puppet. Uczestnik będzie w stanie skonfigurować i zarządzać złożoną infrastrukturą, spójną dla wielu środowisk i aplikacji.

Mocne strony szkolenia

W stale zmieniającym się środowisku, często składającym się z wielu poziomów akceptacji i różnych projektów wymagane jest sprawne, szybkie ale też przewidywalne wdrażanie w formie iteracji. Wybór metodyki DevOps, a w szczególności stosowanie oprogramowania Puppet, pozwala spełnić te wymagania, lecz wymaga umiejętności rozbudowy i utrzymywania kodu wdrożeniowego w długim okresie czasu, dla wielu projektów i środowisk. Zagadnienia poznane na szkoleniu dostarczą uczestnikom wiedzy i narzędzi, pozwalających spełnić takie wymagania dostarczając pełne, spójne, elastyczne i sprawne środowisko wdrożeniowe.

Wymagania

Szkolenie przeznaczone jest zarówno dla programistów jak i administratorów, którzy znają podstawy oprogramowania Puppet. Zakładana jest wiedza porównywalna ze szkoleniem Puppet Essentials. Umiejętność programowania w języku Ruby może być przydatna ale nie jest obowiązkowa.

Parametry szkolenia

2*8 godzin (2*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Program szkolenia:

1. Powtórzenie podstaw języka Puppet ze szkolenia Puppet Essentials.
2. Utrzymywanie złożonej struktury projektów i środowisk
3. Sposoby wdrażania samego kodu wdrożeniowego
4. Zaawansowane elementy składni manifestów
 - I. eksportowane zasoby
 - II. zaawansowane metody definiowanie kolejności wykonania i weryfikacja tej kolejności
 - III. programistyczne tworzenie zasobów
 - IV. dziedziczenie klas
 - V. separacja kodu wdrożeniowego na prywatny i publiczny
5. Zaawansowana praca z plikami i ich częściami przy pomocy concat, augeas i file_line
6. Wykorzystanie konwencji roli i profili dla organizacji kodu wdrożeniowego
7. Najlepsze praktyki dla dziedziczenia klas, podziału na moduły
8. Tworzenie niestandardowych funkcji i faktów
9. Instrumentacja i zarządzanie zwrotne przy pomocy mcollective
10. Automatyzacja instalacji maszyn fizycznym przy użyciu Razor
11. Wprowadzenie do testowania kodu Puppet przy pomocy testów dymnych, jednostkowych i akceptacyjnych
12. Re-używalne moduły Puppet i udostępnianie ich dla społeczności
13. Sposoby szukania i rozwiązywania problemów w kodzie wdrożeniowym
14. Używanie danych generowanych przez Puppet
15. Przydatne moduły takie jak stdlib, docker_platform oraz trocla

