

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Extending Puppet - rozszerzanie elementów ekosystemu Puppet

PUPPET/DEV

Czas trwania: 3 dni

Puppet i inne powiązane z nim technologie udowadniają swoją przydatność jako kompleksowe narzędzia DevOps. Pozwalają na określenie konfiguracji całych środowisk, działających na nich aplikacji i usług

Cele szkolenia

- Uczestnik nabeździe umiejętności pozwalające rozszerzyć funkcjonalność Puppet, Facter, Hiera oraz MCollective, przy pomocy programowania interfejsów programistycznych w Ruby
- Dodatkowo uczestnik nabeździe umiejętności testowania nowego kodu zarówno przy użyciu testów jednostkowych oraz akceptacyjnych oraz pozna mechanizmy ciągłej budowy oraz inspekcji kodu dla Puppet

Zalety

- Aby wykorzystać w pełni możliwości jakie daje ekosystem Puppet, często niezbędne okazuje się rozszerzenie go o dodatkowe funkcjonalności dostosowane do używanych systemów i aplikacji
- Puppet daje możliwość prostego rozszerzenia go o dodatkowe funkcjonalności przy użyciu wtyczek

Dla kogo?

- Szkolenie adresowane jest do zaawansowanych użytkowników Puppet, programistów oraz do administratorów, którzy chcą poznać sposoby rozszerzenia i dostosowania funkcjonalności Puppet do wymagań projektów i środowisk

Wymagania

- Wymagana jest wiedza na temat podstaw systemu Puppet odpowiadająca szkoleniu Puppet Essentials
- Mile widziana jest także podstawowa znajomość Ruby



Program

1. Przegląd ekosystemu Puppet pod kątem możliwości rozszerzania o niestandardowe elementy
 - a. Opis schematu i układ modułu Puppet
 - b. Mechanizm pluginsync
 - c. Moment uruchamiania różnych elementów składowych Puppet
 - d. Mechanizmy testowania jednostkowego RSpec
 - e. Mechanizmy testowania akceptacyjnego Beaker
 - f. Konwencja Puppet_X
2. Zasady pisania reużywalnych modułów
3. Dodatkowe fakty w narzędziu Facter
 - a. Tworzenie nowych faktów
 - b. Testowanie manualne i automatyczne
4. Funkcje w manifestach Puppet
 - a. Tworzenie nowych funkcji
 - b. Testowanie manualne i automatyczne
5. Dodatkowe typy w manifestach Puppet
 - a. Parametry i właściwości w typach
 - b. Walidacja i dostosowanie danych z manifestów
 - c. Testowanie automatyczne przy użyciu RSpec
6. Providery dla typów
 - a. Mechanizm dobierania właściwej implementacji
 - b. Dziedziczenie providerów
 - c. Mechanizmy prefetch, flush i instances
 - d. Testowanie automatyczne przy wykorzystaniu RSpec i Beaker
7. Procesory raportów Puppet
8. Rozszerzanie Hiera o dodatkowe źródła danych
9. Rozszerzenia dla MCollective
 - a. Klient
 - b. Serwer
10. Wykorzystanie PuppetDB poprzez API
11. Wykorzystanie API klasyfikatora Puppet Enterprise
12. Rozwiązywanie problemów

