

SZKOLENIE PODSTAWOWE

Helm - menedżer pakietów aplikacji w Kubernetes

HELM

Czas trwania: 1 dni (8h)

Helm - jako natywny menedżer pakietów aplikacji w Kubernetes

Cele szkolenia

- Poznanie możliwości szablonów w języku Go, służących do tworzenia dynamicznych plików konfiguracyjnych
- Wprowadzenie do obsługi narzędzia Helm - natywnego menedżera pakietów aplikacji w orkiestratorze Kubernetes

Zalety

- Szkolenie dostarcza dużą dawkę praktycznych informacji oraz przykładów, po których szablony języka Go staną proste i przejrzyste
- Zdobywana wiedza ma dużą wartość merytoryczną i może być zastosowana podczas rozwiązywania problemów w rzeczywistych projektach
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretność umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how

Dla kogo?

- Szkolenie dedykowane przede wszystkim deweloperom i administratorom, jednocześnie szeroko wprowadzające każdego zainteresowanego tworzeniem, uruchamianiem i zarządzaniem aplikacjami wykorzystującymi orkiestrator Kubernetes

Wymagania

- Minimum roczne doświadczenie w zarządzaniu aplikacjami używając orkiestratora Kubernetes lub odbycie szkolenia Kubernetes w Praktyce

Program



1. Podstawy Helm
 - a. Czym jest Helm i jak wpływa na zarządzanie aplikacjami w orkiestratorze Kubernetes
 - b. Historia i przyszłość
 - c. Co zmieniło się pomiędzy wersją 2 oraz 3
2. Omówienie podstawowych poleceń powłoki związanych z Helm
 - a. helm install/uninstall - jak zainstalować lub też usunąć aplikacje z klastra Kubernetes
 - b. helm upgrade - jako sposób na aktualizacje konfiguracji
 - c. helm get - komenda wyświetlająca dokładne informacje o zainstalowanych obiektach powiązanych z paczką
 - d. helm history - sprawdzenie historii aktualizacji aplikacji
 - e. helm rollback - przywrócenie jednej z poprzednich wersji konfiguracji
 - f. helm repo - polecenie zarządzające zdalnymi repozytoriami paczek
3. Struktura plików paczki aplikacji
 - a. Chart.yaml - plik w formacie YAML zawierający informacje ogólne o paczce jak jej wersja czy nazwa
 - b. values.yaml - plik ze zmiennymi paczki
 - c. charts/ - katalog, w którym zdefiniowane są wszystkie zależności aplikacji
 - d. crds/ - katalog z obiektami rozszerzonymi API klastra Kubernetes
 - e. templates/ - katalog z szablonami, które w połączeniu ze zmiennymi wygenerują manifesty aplikacji
4. YAML w detalach
 - a. struktury danych
 - b. typy zmiennych
 - c. wieloliniowe ciągi znaków
5. Praca z szablonami z użyciem obiektów wbudowanych
 - a. Release - obiekt opisujący uruchomioną aplikację
 - b. Values/Chart - jako sposób na odwołanie się do wartości zdefiniowanych w pliku values.yaml/Chart.yaml
 - c. Files - funkcja pozwalająca na zaciągnięcie zawartości plików i wygenerowaniu na ich podstawie obiektów typu ConfigMap lub Secret
 - d. Capabilities - obiekt umożliwiający sprawdzenie: wersji API platformy Kubernetes lub dostępnych wersji obiektów API w klastrze, w którym instalowana jest aplikacja
 - e. Template - funkcja dająca dostęp do informacji o lokalizacji szablonów
6. Instrukcje warunkowe oraz pętle przydatne przy tworzeniu szablonów
 - a. if/else - instrukcje umożliwiające warunkowe dodawanie zawartości manifestów
 - b. range - pętla pozwalająca na iteracje elementów dowolnej listy lub też słownika
 - c. with - instrukcja definiująca obecny zakres struktury zmiennych
7. Konfiguracja
 - a. .helmignore jako sposób na ignorowanie określonych plików w strukturze paczki aplikacji
 - b. Jak przypiąć powiadomienia lub inne akcje do cyklu życia aplikacji: instalacji, aktualizacji lub usunięcia
 - c. Repozytoria paczek oprogramowania - przegląd aplikacji oraz rozwiązań służących do przechowywania paczek aplikacji
 - d. Zarządzanie uprawnieniami użytkowników korzystających z Helm wersji v3



e. Podsumowanie najlepszych praktyk związanych z zarządzaniem aplikacjami przy użyciu Helm

