

SZKOLENIE ŚREDNIO ZAAWANSOWANE

Terraform - Automatyzacja wdrożeń (Infrastructure as Code) w chmurze AWS

DEVOPS/TF

Czas trwania: 2 dni

Cele szkolenia

.....

- Przeprowadzenie uczestnika w przekrojowy sposób przez różne aspekty pracy z Terraform, skupiając się przy tym na zrozumieniu najlepszych praktyk, rolę narzędzia Terraform w domenie DevOps/Infrastructure As Code (IaC) oraz zdobyciu praktycznych umiejętności
- Zrozumie koncepcję Infrastructure As Code, będzie znał zalety oraz wady tego podejścia
- Nauka użycia Terraform we właściwy sposób
- Poznanie najlepszych praktyk pracy z Terraform
- Nauka wdrażania infrastruktury przy podejściu IAC u wiodących dostawców platform IaaS, PaaS oraz SaaS
- Zrozumienie różnicy pomiędzy Terraform a CloudFormation (AWS native) oraz innymi narzędziami IAC (Chef, Ansible, Puppet, ...)
- Nauka zarządzania infrastrukturą przy pomocy Terraform
- Nauka testowania i walidowania infrastruktury IT przy pomocy Terraform

Zalety

.....

- 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?

.....

- Programiści, architekci oraz administratorzy, którzy chcieliby nauczyć się zarządzać infrastrukturą IT w sposób automatyczny, wykorzystując przy tym najlepsze praktyki SDLC, takie jak CI, CD, code review, automatyczne testy itp.
- Programiści, architekci oraz administratorzy, którzy chcieliby zrozumieć, na czym polega koncepcja Infrastructure as Code



- Programiści, architekci oraz administratorzy, którzy chcieliby poszerzyć swoje kompetencje DevOps
- Programiści, architekci oraz administratorzy, którzy chcieliby zdobyć jedną z kluczowych umiejętności przy pracy w środowiskach Cloud
- Programiści, architekci oraz administratorzy, którzy chcieliby być w stanie zaprojektować system IT na poziomie infrastruktury (system produkcyjny lub np. POC) i komponentów, począwszy od warstwy sieci, poprzez bezpieczeństwo, aż po aplikacje/mikroserwisy
- Programiści, architekci oraz administratorzy, którzy chcieliby poznać narzędzie, które pozwoli dostarczać blue printy systemów IT w postaci kodu źródłowego
- Programiści, architekci oraz administratorzy, którzy chcieliby poszerzyć swój warsztat o kompetencję IAC/Terraform, zrozumieć jak znane im praktyki SDLC mogą być zastosowane do projektowania infrastruktury systemów

Wymagania

- Uczestnicy powinni znać co najmniej jedną chmurę (np. AWS, Azure, GCP) oraz orientować się w tematyce DevOps
- Znajomość Terraform nie jest wymagana
- Ćwiczenia będą realizowane w oparciu o serwisy w chmurze AWS



Program

1. Wprowadzenie do Terraform
 - a. DevOps
 - b. Co to jest Infrastructure as Code?
 - c. Zalety podejścia IAC
 - d. Jak działa Terraform
 - e. Terraform a inne narzędzia IAC (porównanie) - Chef, Puppet, Ansible, CloudFormation
2. Pierwsze kroki z Terraform
 - a. Konfiguracja konta AWS
 - Konfiguracja AWS CLI
 - b. Instalacja Terraform 0.12
 - c. Przedstawienie funkcjonalności AWS (AWS provider) dostępnej w Terraform
 - Data sources vs resources
 - Przegląd dostępnych obiektów źródłowych
 - Przegląd dostępnych obiektów do zarządzania stanem infrastruktury
 - d. Instalacja oraz deinstalacja przykładowych obiektów chmury AWS (np. EC2, RDS, VPC)
 - e. Stan infrastruktury w Terraform (State File)
 - Czym jest plik stanu infrastruktury?
 - Jak edytować plik stanu infrastruktury?
 - f. Zmienne (var), zmienne lokalne (locals), podstawy modułów, funkcje oraz wyrażenia w Terraform
 - g. Obiekty
 - Meta-argumenty - depends on, count, for_each, provider, provisioner, connection
 - Cykl życia (lifecycle) - modyfikowanie domyślnych zachowań przy pomocy akcji cyklu życia
 - h. Wyrażenia warunkowe
 - i. Pętle for
3. Jak projektować infrastrukturę systemów w Terraform
 - a. Moduły - wprowadzenie (parametry wejściowe, sposób wersjonowania oraz parametry wyjściowe)
 - b. Moduły - jak projektować oraz korzystać efektywnie
 - c. Struktura projektów w Terraform, różne podejścia
 - d. Jak i co testować w Terraform?
 - Unit Testy, Testy integracyjne oraz testy E2E
 - e. Jak zarządzać stanem optymalnie?
 - Przechowywanie stanu lokalnie vs Terraform backend
 - Omówienie dostępnych backendów (s3, AzureRM, Artifactory, ...)
 - f. Kwestie bezpieczeństwa
 - Przechowywanie sekretów/hasel
 - Bezpieczne przechowywanie pliku stanu
 - g. Jak pracować w zespole nad projektami Terraform?
4. Podsumowanie

