

## Spis treści

O autorze (9)

O korektorze merytorycznym (10)

Przedmowa (11)

Wstęp (15)

Rozdział 1. Czym są mikrousługi? (17)

Geneza architektury SOA (18)

Podejście monolityczne (19)

Podejście mikrousługowe (22)

Zalety mikrousług (24)

Rozdzielenie zakresów odpowiedzialności (24)

Mniejsze projekty (24)

Skalowanie i wdrażanie (25)

Wady mikrousług (26)

Nielogiczny podział aplikacji (26)

Więcej interakcji sieciowych (27)

Powielanie danych (27)

Problemy z kompatybilnością (28)

Skomplikowane testy (28)

Implementacja mikrousług w języku Python (29)

Standard WSGI (29)

Biblioteki Greenlet i Gevent (31)

Platformy Twisted i Tornado (33)

Moduł asyncio (34)

Wydajność kodu (36)

Podsumowanie (38)

## Rozdział 2. Platforma Flask (39)

Jaka wersja Pythona? (40)

Obsługa zapytań w platformie Flask (41)

Kierowanie zapytań (44)

Zapytanie (47)

Odpowiedź (49)

Wbudowane funkcjonalności platformy Flask (50)

Obiekt session (51)

Zmienne globalne (51)

Sygnały (52)

Rozszerzenia i pośredniki (53)

Szablony (55)

Konfiguracja (56)

Konspekty (58)

Obsługa błędów i diagnostyka kodu (59)

Szkielet mikrousługi (62)

Podsumowanie (64)

## Rozdział 3. Cykl doskonały: kodowanie, testowanie, dokumentowanie (65)

Rodzaje testów (67)

Testy jednostkowe (67)

Testy funkcjonalne (70)

Testy integracyjne (72)

Testy obciążeniowe (72)

Testy całościowe (75)

Pakiet WebTest (76)

Narzędzia pytest i tox (78)

Dokumentacja programistyczna (80)

Ciągła integracja (84)

System Travis CI (85)

System ReadTheDocs (86)

System Coveralls (86)

Podsumowanie (88)

## Rozdział 4. Aplikacja Runnerly (89)

Aplikacja Runnerly (89)

Historie użytkowników (90)

Struktura monolityczna (91)

Model (92)

Widok i szablon (93)

Zadania wykonywane w tle aplikacji (96)

Uwierzytelnianie i autoryzowanie użytkowników (99)

Zebranie elementów w monolityczną całość (102)

Dzielenie monolitu (104)

Usługa danych (105)

Standard Open API 2.0 (106)

Dalszy podział aplikacji (108)

Podsumowanie (110)

## Rozdział 5. Interakcje z innymi usługami (111)

### Wywołania synchroniczne (112)

Sesje w aplikacji Flask (113)

Pula połączeń (116)

Pamięć podręczna i nagłówki HTTP (117)

Przyspieszanie przesyłania danych (120)

Wnioski (124)

### Wywołania asynchroniczne (125)

Kolejki zadań (125)

Kolejki tematyczne (126)

Publikowanie i subskrybowanie komunikatów (130)

Wywołania RPC w protokole AMQP (130)

Wnioski (131)

### Testy (131)

Imitowanie wywołań synchronicznych (131)

Imitowanie wywołań asynchronicznych (133)

### Podsumowanie (135)

## Rozdział 6. Monitorowanie usług (137)

### Centralizacja dzienników (138)

Konfiguracja systemu Graylog (139)

Wysyłanie logów do systemu Graylog (142)

Dodatkowe pola (145)

### Wskaźniki wydajnościowe (146)

Wskaźniki systemowe (146)

Wskaźniki wydajnościowe kodu (148)

Wskaźniki wydajnościowe serwera WWW (150)

Podsumowanie (151)

## Rozdział 7. Zabezpieczanie usług (153)

Protokół OAuth2 (154)

Uwierzytelnienie oparte na tokenach (156)

Standard JWT (156)

Biblioteka PyJWT (158)

Uwierzytelnianie za pomocą certyfikatu X.509 (160)

Mikrousluga TokenDealer (162)

Stosowanie usługi TokenDealer (165)

Zapora WAF (167)

Platforma OpenResty: serwer Nginx i język Lua (169)

Zabezpieczanie kodu (174)

Sprawdzanie odbieranych zapytań (175)

Ograniczanie zakresu działania aplikacji (178)

Analizator Bandit (179)

Podsumowanie (181)

## Rozdział 8. Wszystko razem (183)

Tworzenie interfejsu za pomocą biblioteki ReactJS (184)

Język JSX (185)

Komponenty ReactJS (186)

Biblioteka ReactJS i platforma Flask (189)

Bower, npm i Babel (190)

Współdzielenie zasobów między domenami (193)

Uwierzytelnianie użytkowników i autoryzowanie zapytań (195)

Komunikacja z usługą danych (195)

Uzyskiwanie tokena Strava (196)

Uwierzytelnienie w kodzie JavaScript (198)

Podsumowanie (199)

## Rozdział 9. Spakowanie i uruchomienie Runnerly (201)

Narzędzia pakujące (202)

Kilka definicji (203)

Pakowanie projektów (204)

Wersje projektu (211)

Udostępnianie projektu (213)

Rozpowszechnianie projektu (215)

Uruchamianie mikrousług (218)

Zarządzanie procesami (220)

Podsumowanie (223)

## Rozdział 10. Usługi kontenerowe (225)

Czym jest Docker? (226)

Docker od podstaw (227)

Uruchamianie aplikacji Flask na platformie Docker (229)

Kompletny system - OpenResty, Circus i Flask (231)

Platforma OpenResty (232)

Menedżer Circus (233)

Wdrożenia kontenerowe (236)

Docker Compose (237)

Klastrowanie i prowizjonowanie kontenerów (239)

Podsumowanie (241)

## Rozdział 11. Instalacja w chmurze AWS (243)

Chmura AWS (244)

Kierowanie zapytań - Route53, ELB i AutoScaling (245)

Wykonywanie kodu - EC2 i Lambda (246)

Gromadzenie danych - EBS, S3, RDS, ElasticCache i CloudFront (247)

Powiadamianie - SES, SQS i SNS (248)

Prowizjonowanie i uruchamianie - CloudFormation i ECS (250)

Podstawy wdrażania mikrousług w chmurze AWS (250)

Utworzenie konta w chmurze AWS (251)

Instalowanie instancji EC2 z systemem CoreOS (253)

Wdrażanie klastrów przy użyciu usługi ECS (257)

Usługa Route53 (262)

Podsumowanie (263)

## Rozdział 12. Co dalej? (265)

Iteratory i generatory (266)

Koprocedury (269)

Biblioteka asyncio (270)

Platforma aiohttp (271)

Platforma Sanic (272)

Model asynchroniczny i synchroniczny (273)

Podsumowanie (275)

## Skorowidz (276)