

Spis treści

Wstęp 13

Podziękowania 15

O książce 17

O autorze 21

CZĘŚĆ 1. WYJAŚNIENIE, CZYM SĄ KONTENERY I OBRAZY DOCKERA 23

1. Zanim zaczniesz 25

1.1. Dlaczego kontenery podbiją świat? 26

1.1.1. Przenoszenie aplikacji do chmury 27

1.1.2. Modernizacja starych aplikacji 28

1.1.3. Tworzenie nowej aplikacji chmurowej 29

1.1.4. Innowacje techniczne: rozwiązania bezserwerowe i inne 32

1.1.5. Cyfrowe przetwarzanie przy użyciu rozwoju i operacji 33

1.2. Dla kogo jest przeznaczona ta książka? 34

1.3. Tworzenie środowiska roboczego 35

1.3.1. Instalacja Dockera 35

1.3.2. Weryfikacja działania Dockera 37

1.3.3. Pobieranie kodów źródłowych do książki 38

1.3.4. Pamiętanie o poleceniach czyszczących 38

1.4. Jak być efektywnym od zaraz? 39

2. Docker — przedstawienie i przykład Witaj, świecie! 41

2.1. Uruchamianie aplikacji Witaj, świecie! w kontenerze 41

2.2. A zatem czym jest kontener? 45

2.3. Nawiązywanie połączenia z kontenerem jak ze zdalnym komputerem 47

2.4. Uruchamianie witryn internetowych w kontenerze 51

2.5. Jak Docker wykonuje kontenery? 55

2.6. Laboratorium. Poznawanie systemu plików kontenera 58

3. *Tworzenie własnych obrazów Dockera* 61
 - 3.1. Stosowanie obrazu kontenera z serwisu Docker Hub 61
 - 3.2. Pisanie pierwszego pliku Dockerfile 66
 - 3.3. Budowanie własnego obrazu kontenera 68
 - 3.4. Obrazy Dockera i warstwy obrazów 71
 - 3.5. Optymalizacja plików Dockerfile pod kątem wykorzystania pamięci podręcznej warstw obrazów 74
 - 3.6. Laboratorium 77
4. *Pakowanie aplikacji w formie kodu źródłowego do obrazów Dockera* 79
 - 4.1. Po co komu serwer budowania, skoro można użyć pliku Dockerfile? 80
 - 4.2. Przegląd aplikacji: kod źródłowy Javy 84
 - 4.3. Przegląd aplikacji: kody źródłowe aplikacji Node.js 89
 - 4.4. Przegląd aplikacji: kod źródłowy Go 92
 - 4.5. Wieloetapowe pliki Dockerfile 96
 - 4.6. Laboratorium 97
5. *Udostępnianie obrazów w serwisie Docker Hub oraz innych rejestrach* 99
 - 5.1. Korzystanie z rejestrów, repozytoriów i znaczników obrazów 100
 - 5.2. Umieszczanie własnych obrazów w rejestrze Docker Hub 101
 - 5.3. Uruchamianie i stosowanie własnego rejestru Dockera 106
 - 5.4. Efektywne stosowanie znaczników obrazów 111
 - 5.5. Przekształcanie oficjalnych obrazów w złote obrazy 113
 - 5.6. Laboratorium 115
6. *Stosowanie woluminów Dockera do trwałego przechowywania danych* 117
 - 6.1. Dlaczego dane w kontenerach nie są trwałe? 118
 - 6.2. Uruchamianie kontenerów z woluminami Dockera 123
 - 6.3. Uruchamianie kontenerów z dowiązaniami systemu plików 129
 - 6.4. Ograniczenia dowiązań 132
 - 6.5. Konstrukcja systemu plików kontenera 136
 - 6.6. Laboratorium 138

CZĘŚĆ 2. URUCHAMIANIE ROZPROSZONYCH APLIKACJI W KONTENERACH 141

7. *Uruchamianie aplikacji wielokontenerowych przy użyciu Docker Compose* 143
 - 7.1. Anatomia pliku Docker Compose 144
 - 7.2. Uruchamianie aplikacji wielokontenerowych przy użyciu Compose 148
 - 7.3. Jak Docker podłącza do siebie komponenty 155
 - 7.4. Konfigurowanie aplikacji w Docker Compose 158
 - 7.5. Problemy, jakie rozwiązuje Docker Compose 163
 - 7.6. Laboratorium 164
8. *Wspieranie niezawodności przy użyciu testów stanu i zależności* 167
 - 8.1. Umieszczanie testów stanu w obrazach Dockera 168
 - 8.2. Uruchamianie kontenerów przy użyciu testów zależności 174
 - 8.3. Pisanie własnych narzędzi na potrzeby testów aplikacji 178
 - 8.4. Definiowanie testów stanu i testów zależności w plikach Docker Compose 183
 - 8.5. W jaki sposób testy wspierają samonaprawiające się aplikacje 187
 - 8.6. Laboratorium 189
9. *Zapewnianie możliwości obserwowania poprzez użycie skonteneryzowanego monitoringu* 191
 - 9.1. Stos monitorujący dla skonteneryzowanych aplikacji 192
 - 9.2. Udostępnianie metryk aplikacji 197
 - 9.3. Uruchamianie kontenera Prometheusa w celu gromadzenia metryk 202
 - 9.4. Uruchamianie kontenera Grafana w celu wizualizacji metryk 208
 - 9.5. Możliwości obserwowania i ich poziomy 216
 - 9.6. Laboratorium 217
10. *Uruchamianie wielu środowisk przy użyciu Docker Compose* 219
 - 10.1. Wdrażanie wielu aplikacji przy użyciu Docker Compose 220
 - 10.2. Stosowanie Docker Compose z plikami przesłaniającymi 223
 - 10.3. Wstrzykiwanie konfiguracji przy wykorzystaniu zmiennych środowiskowych i danych poufnych 231

10.4. Redukcja powtórzeń przy użyciu pól rozszerzeń 238

10.5. Proces stosowania konfiguracji przez Dockera 240

10.6. Laboratorium 242

11. **Budowanie i testowanie aplikacji przy użyciu Dockera i Docker Compose 245**

11.1. Jak działa proces ciągłej integracji w przypadku stosowania Dockera? 246

11.2. Uruchamianie infrastruktury do budowania w oparciu o Dockera 247

11.3. Zapisywanie ustawień budowania w plikach Docker Compose 257

11.4. Tworzenie zadań ciągłej integracji, których jedyną zależnością jest Docker 262

10.5. Zastosowanie kontenerów w procesie ciągłej integracji 266

11.6. Laboratorium 268

CZĘŚĆ 3. DZIAŁANIE W DUŻEJ SKALI Z UŻYCIEM ORKIESTRATORA KONTENERÓW277

12. **Przedstawienie orkiestracji: Docker Swarm i Kubernetes 271**

12.1. Czym jest orkiestrator kontenerów? 272

12.2. Tworzenie klastra Docker Swarm 273

12.3. Uruchamianie aplikacji jako usług Docker Swarm 278

12.4. Zarządzanie ruchem sieciowym w klastrze 284

12.5. Wybór pomiędzy Docker Swarm a Kubernetes 292

12.6. Laboratorium 295

13. **Wdrażanie aplikacji rozproszonych jako stosów w Docker Swarm 297**

13.1. Stosowanie Docker Compose do wdrożeń w środowiskach produkcyjnych 298

13.2. Zarządzanie konfiguracją aplikacji przy użyciu obiektów konfiguracyjnych 303

13.3. Zarządzanie wrażliwymi ustawieniami przy wykorzystaniu danych poufnych 309

13.4. Przechowywanie danych na woluminach w klastrach Swarm 313

13.5. Przedstawienie sposobu, w jaki klastry zarządzają stosami 318

13.6. Laboratorium 320

14.	<i>Automatyzacja wydawania wersji aplikacji przy użyciu aktualizacji i wycofywania</i>	321
14.1.	Aktualizowanie aplikacji przy użyciu Dockera	322
14.2.	Konfigurowanie wprowadzania zmian w środowiskach produkcyjnych przy użyciu Compose	328
14.3.	Konfigurowanie wycofywania usług	332
14.4.	Zarządzanie przestojami klastra	339
14.5.	Zapewnianie wysokiej dostępności w klastrach Swarm	343
14.6.	Laboratorium	344
15.	<i>Konfiguracja Dockera pod kątem bezpiecznego dostępu zdalnego i operacji CI/CD</i>	347
15.1.	Opcje punktów końcowych API Dockera	348
15.2.	Konfigurowanie bezpiecznych zdalnych połączeń z Dockerem	353
15.3.	Stosowanie kontekstów Dockera do pracy ze zdalnymi silnikami	362
15.4.	Dodawanie wdrażania ciągłego do potoku CI	365
15.5.	Model dostępu stosowany w Dockerze	372
15.6.	Laboratorium	373
16.	<i>Budowanie obrazów Dockera działających wszędzie: w systemach Linux i Windows, na procesorach Intel i Arm</i>	375
16.1.	Dlaczego obrazy wieloarchitekturowe są ważne?	376
16.2.	Budowanie obrazów wieloarchitekturowych z użyciem jednego lub kilku plików Dockerfile	380
16.3.	Przesyłanie do rejestrów obrazów wieloarchitekturowych i manifestów	386
16.4.	Budowanie obrazów wieloarchitekturowych przy użyciu Docker Buildx	392
16.5.	Określanie, czy obrazy wieloarchitekturowe pasują do naszych planów	398
16.6.	Laboratorium	399
CZĘŚĆ 4.	PRZYGOTOWYWANIE KONTENERÓW DO ZASTOSOWAŃ PRODUKCYJNYCH	401
17.	<i>Optymalizacja obrazów Dockera pod kątem wielkości, szybkości działania i bezpieczeństwa</i>	403
17.1.	Optymalizacja obrazów Dockera	404
17.2.	Wybór odpowiednich obrazów bazowych	409

17.3. Minimalizacja liczby i wielkości warstw 415

17.4. Podnoszenie wieloetapowych operacji budowania
na wyższy poziom 419

17.5. Dlaczego optymalizacja ma znaczenie? 423

17.6. Laboratorium 425

18. *Zarządzanie konfiguracją aplikacji w kontenerach* 427

18.1. Wielowarstwowe podejście do konfigurowania aplikacji 428

18.2. Pakowanie konfiguracji dla każdego środowiska 432

18.3. Wczytywanie konfiguracji ze środowiska uruchomieniowego 437

18.4. Konfigurowanie starych aplikacji w taki sam sposób jak nowych 442

18.5. Dlaczego opłaca się stosować elastyczny model konfiguracji? 448

18.6. Laboratorium 451

19. *Zapisywanie dzienników aplikacji i zarządzanie nimi
w Dockerze* 453

19.1. Witamy w świecie stderr i stdout! 454

19.2. Przekazywanie komunikatów z innych źródeł
do standardowego strumienia wyjściowego 459

19.3. Gromadzenie i przekazywanie wpisów z dzienników kontenerów 464

19.4. Zarządzanie wyświetlaniem i gromadzeniem wpisów 472

19.5. Poziomy rejestrowania komunikatów 475

19.6. Laboratorium 478

20. *Kontrola ruchu HTTP do kontenerów
przy użyciu odwrotnego serwera pośredniczącego* 479

20.1. Czym jest odwrotny serwer pośredniczący? 480

20.2. Obsługa trasowania i SSL w odwrotnym serwerze pośredniczącym 486

20.3. Poprawianie wydajności i niezawodności przy użyciu serwera
pośredniczącego 491

20.4. Chmurowy odwrotny serwer pośredniczący 497

20.5. Wzorce, jakie można tworzyć, używając odwrotnych serwerów
pośredniczących 505

20.6. Laboratorium 507

21. *Komunikacja asynchroniczna
przy użyciu kolejki komunikatów* 509

21.1. Czym jest asynchroniczne przesyłanie komunikatów? 510

21.2. Stosowanie chmurowych kolejek komunikatów 514

- 21.3. Konsumowanie i obsługa komunikatów 519
- 21.4. Dodawanie nowych możliwości przy użyciu programów obsługi komunikatów 522
- 21.5. Wzorce stosowania asynchronicznych komunikatów 527
- 21.6. Laboratorium 529

22. *Końca nie widać* 531

- 22.1. Uruchamiaj własne rozwiązania stanowiące potwierdzenie koncepcji 531
- 22.2. Poszukaj możliwości stosowania Dockera w swojej firmie 532
- 22.3. Planowanie ścieżki produkcji 534
- 22.4. Poznaj społeczność Dockera 535

A *Rozwiązania laboratoriów* 537

- Rozdział 2. 537
- Rozdział 3. 539
- Rozdział 4. 539
- Rozdział 5. 540
- Rozdział 6. 541
- Rozdział 7. 542
- Rozdział 8. 543
- Rozdział 9. 543
- Rozdział 10. 545
- Rozdział 11. 545
- Rozdział 12. 546
- Rozdział 13. 546
- Rozdział 14. 546
- Rozdział 15. 549
- Rozdział 16. 550
- Rozdział 17. 552
- Rozdział 18. 552
- Rozdział 19. 553
- Rozdział 20. 555
- Rozdział 21. 556