

Spis treści

O autorach	15
O recenzencie	17
Przedmowa	19
Część I. Wprowadzenie do platformy Docker — kontenery i programowanie w lokalnym środowisku	25
Rozdział 1. Wprowadzenie do platformy Docker	27
Geneza platformy Docker	28
Hosting kolokacyjny	28
Hosting samodzielny	29
Centra danych	29
Wirtualizacja jako sposób na ekonomiczne wykorzystanie zasobów	31
Rosnące zapotrzebowanie na energię	33
Wirtualizacja i chmura obliczeniowa	34
Dalsza optymalizacja zasobów centrów danych przy użyciu kontenerów	36
Podsumowanie	38
Dalsza lektura	38
Rozdział 2. Tworzenie aplikacji z użyciem VirtualBox i kontenerów Docker	39
Wymagania techniczne	40
Problem zanieczyszczania systemu plików hosta	40
Tworzenie maszyn wirtualnych za pomocą programu VirtualBox	41
Wprowadzenie do wirtualizacji	41
Tworzenie maszyny wirtualnej	42
Dodatki do systemu gościa	43
Instalacja programu VirtualBox	44

Kontenery Docker	45
Wprowadzenie do kontenerów	45
Tworzenie aplikacji przy użyciu platformy Docker	47
Pierwsze kroki z platformą Docker	48
Automatyzacja poleceń za pomocą skryptów	49
Podsumowanie	59
Dalsza lektura	60
 Rozdział 3. Udostępnianie kontenerów w serwisie Docker Hub	 61
Wymagania techniczne	62
Wprowadzenie do serwisu Docker Hub	62
Korzystanie z serwisu Docker Hub za pomocą wiersza poleceń	63
Korzystanie z serwisu Docker Hub za pomocą przeglądarki	64
Implementacja kontenera MongoDB w aplikacji	66
Uruchomienie powłoki kontenera	69
Wprowadzenie do architektury mikrouslugowej	71
Skalowalność	72
Komunikacja między kontenerami	72
Implementacja prostej aplikacji mikrouslugowej	75
Udostępnianie kontenerów w serwisie Docker Hub	79
Podsumowanie	82
Dalsza lektura	82
 Rozdział 4. Tworzenie systemów przy użyciu kontenerów	 83
Wymagania techniczne	84
Wprowadzenie do narzędzia Docker Compose	84
Problem ze skryptami	85
Pliki konfiguracyjne narzędzia Docker Compose	86
Dziedziczenie konfiguracji	89
Sekcja depends_on	90
Definiowanie udostępnianych portów	91
Lokalne sieci w platformie Docker	94
Definiowanie sieci za pomocą skryptów	94
Tworzenie sieci za pomocą narzędzia Docker Compose	96
Wiązanie systemów plików hosta i kontenera	97
Optymalizacja wielkości kontenera	98
Skrypt build.sh	100
Inne narzędzia kompozycyjne	101
Docker Swarm	101
Kubernetes	101
Packer	102
Podsumowanie	102
Dalsza lektura	103

Część II. Platforma Docker w środowisku produkcyjnym 105

Rozdział 5. Wdrażanie i uruchamianie kontenerów w środowisku produkcyjnym 107

Wymagania techniczne	108
Przykładowa aplikacja Shiplt Clicker	108
Uruchamianie kontenerów Docker w środowisku produkcyjnym	109
Minimalne środowisko produkcyjne	109
Niezbędne minimum — Docker i Docker Compose na jednym hoście	110
Wsparcie dla platformy Docker	110
Problemy z wdrażaniem na pojedynczym hoście	110
Zarządzane usługi chmurowe	111
Google Kubernetes Engine	111
AWS Elastic Beanstalk	112
AWS ECS i Fargate	112
AWS EKS	112
Microsoft Azure Kubernetes Service	113
DigitalOcean Docker Swarm	113
Tworzenie własnych klastrów Kubernetes	113
Dobieranie właściwej konfiguracji produkcyjnej	114
Ćwiczenie — dołącz do zespołu Shiplt Clicker	116
Ćwiczenie — wybór właściwej metody wdrożenia	119
Ćwiczenie — ocena plików Dockerfile i docker-compose.yml	121
Podsumowanie	121

Rozdział 6. Wdrażanie aplikacji przy użyciu Docker Compose 123

Wymagania techniczne	124
Przykładowa aplikacja — Shiplt Clicker v2	124
Dobór sprzętu i systemu operacyjnego dla aplikacji jednoserwerowej	124
Wymagania dla wdrożenia jednoserwerowego	124
Przygotowanie hosta do uruchomienia platformy Docker i narzędzia Docker Compose	125
Instalacja platformy Docker i narzędzia Git	126
Wdrażanie aplikacji przy użyciu plików konfiguracyjnych i skryptów	127
Weryfikacja pliku Dockerfile	127
Weryfikacja pliku docker-compose.yml	129
Przygotowanie produkcyjnego pliku .env	131
Skrypty	132
Ćwiczenie — przechowywanie plików aplikacyjnych poza serwerem produkcyjnym	135
Ćwiczenie — zabezpieczenie środowiska produkcyjnego	135
Monitorowanie niewielkich aplikacji — dzienniki i alarmy	136
Dzienniki	136
Alarmy	137
Ograniczenia aplikacji jednoserwerowych	137
Brak automatycznego przełączania awaryjnego	138
Brak skalowalności w poziomie wraz ze wzrostem obciążenia	138
Niestabilność działania z powodu błędnej konfiguracji	138
Katastrofalne skutki awarii w przypadku braku kopii zapasowej	139

Studium przypadku — migracja z systemu CoreOS i usługi DigitalOcean do CentOS 7 i AWS	139
Podsumowanie	139
Dalsza lektura	140
Rozdział 7. Ciągłe wdrażanie oprogramowania przy użyciu systemu Jenkins	141
Wymagania techniczne	142
Przykładowa aplikacja — Shiplt Clicker v3	142
Wykorzystanie systemu Jenkins w procesie ciągłej integracji oprogramowania	143
Pułapki, których powinien unikać	143
Wykorzystanie istniejącego serwera	144
Instalacja systemu Jenkins	144
Ciągłe wdrażanie oprogramowania przy użyciu systemu Jenkins	148
Plik Jenkinsfile i połączenie z serwerem	148
Testowanie systemu Jenkins i platformy Docker za pomocą skryptu procesowego	148
Łączenie z serwerem produkcyjnym za pomocą usługi SSH	150
Modyfikowanie konfiguracji za pomocą systemu Jenkins	154
Umieszczenie pliku Jenkinsfile w serwisie GitHub	154
Zmiana źródeł wszystkich repozytoriów	156
Definiowanie zmiennych środowiskowych dla serwera produkcyjnego	157
Budowanie kontenerów i umieszczanie ich w serwisie Docker Hub	158
Umieszczanie kontenerów w serwisie Docker Hub i wdrażanie ich w środowisku produkcyjnym	159
Wdrażanie różnych odgałęzień oprogramowania w kilku środowiskach	162
Utworzenie środowiska testowego	162
Definiowanie zmiennych środowiskowych dla serwera testowego	163
Wymuszenie wdrożenia testowego odgałęzienia projektu	163
Złożoność i ograniczenia skalowalności systemu Jenkins	165
Zarządzanie wieloma hostami	165
Złożoność skryptów	166
Kiedy wiadomo, że została osiągnięta granica?	166
Podsumowanie	167
Dalsza lektura	167
Rozdział 8. Wdrażanie kontenerów Docker przy użyciu platformy Kubernetes	169
Wymagania techniczne	170
Opcje lokalnej instalacji platformy Kubernetes	170
Docker Desktop i platforma Kubernetes	170
Minikube	171
Sprawdzenie poprawności działania platformy Kubernetes	172
Wdrożenie przykładowej aplikacji Shiplt Clicker v4	172
Instalacja programu Helm	172
Lokalne wdrożenie aplikacji Shiplt Clicker i kontrolera NGINX Ingress Controller	173
Dobór dystrybucji platformy Kubernetes	175
Google Kubernetes Engine	175
AWS EKS	175
Red Hat OpenShift	176
Microsoft Azure Kubernetes Service	176
Inne opcje	177

Pojęcia stosowane w platformie Kubernetes	178
Obiekty	178
Mapy ConfigMap	179
Pody	180
Węzły	180
Usługi	180
Kontrolery wejściowe	181
Skrytki	182
Przestrzenie nazw	187
Konfigurowanie usługi AWS EKS za pomocą szablonu CloudFormation	187
Wprowadzenie do szablonów AWS EKS Quick Start CloudFormation	188
Przygotowanie konta AWS	188
Szablony AWS EKS Quick Start CloudFormation	192
Konfigurowanie klastra EKS	195
Wdrożenie aplikacji w klastrze AWS EKS i ograniczenie zasobów	197
Konfigurowanie ograniczeń chroniących przed wyciekami pamięci i przeciążeniem procesora	197
Przygotowanie aplikacji Shiplt Clicker do korzystania z kontrolera ALB	198
Wdrożenie aplikacji Shiplt Clicker w klastrze EKS	198
Repozytorium AWS Elastic Container Registry w klastrze AWS EKS	199
Tworzenie repozytorium ECR	200
Rozdzielanie środowisk za pomocą etykiet i przestrzeni nazw	202
Przykład — oznaczenie etykietami środowisk w domyślnej przestrzeni nazw	202
Środowiska programistyczne, akceptacyjne, testowe i produkcyjne	203
Podsumowanie	204
Dalsza lektura	204
 Rozdział 9. Ciągłe wdrażanie oprogramowania w chmurze przy użyciu platformy Spinnaker	 207
Wymagania techniczne	208
Zaktualizowana wersja aplikacji Shiplt Clicker v5	208
Usprawnienie platformy Kubernetes pod kątem utrzymywania aplikacji	209
Zarządzanie klastrem EKS za pomocą lokalnej stacji roboczej	209
Diagnozowanie problemów z połączeniem narzędzia kubectl z klastrem	210
Przełączanie pomiędzy kontekstem klastra i lokalnej stacji	210
Sprawdzenie poprawności działania kontrolera wejściowego ALB	211
Przygotowanie domeny Route 53 i certyfikatu	211
Utworzenie i wdrożenie aplikacji Shiplt Clicker v5	212
Platforma Spinnaker — kiedy i dlaczego są niezbędne bardziej zaawansowane wdrożenia	215
Wprowadzenie do platformy Spinnaker	215
Instalacja platformy Spinnaker w klastrze AWS EKS za pomocą programu Helm	217
Komunikacja z platformą Spinnaker za pomocą proxy kubectll	218
Udostępnianie platformy za pomocą kontrolera wejściowego ALB	218
Konfiguracja platformy Spinnaker za pomocą programu Halyard	220
Połączenie platformy Spinnaker z systemem Jenkins	220
Integracja systemu Jenkins z platformą Spinnaker i repozytorium ECR	221
Połączenie platformy Spinnaker z serwisem GitHub	226

Połączenie platformy Spinnaker z serwisem Docker Hub	226
Diagnozowanie problemów z platformą Spinnaker	227
Prosta strategia wdrożenia aplikacji Shiplt Clicker za pomocą platformy Spinnaker	228
Definiowanie aplikacji w platformie Spinnaker	228
Definiowanie procesu w platformie Spinnaker	229
Utworzenie wpisu DNS dla kontrolera wejściowego	234
Funkcjonalności wdrożeniowe i testowe platformy Spinnaker	235
Wdrożenie kanarkowe	235
Wdrożenie „czerwone/czarne”	235
Anulowanie wdrożenia	236
Testowanie aplikacji	236
Podsumowanie	237
Dalsza lektura	237
Rozdział 10. Monitorowanie kontenerów Docker przy użyciu systemów Prometheus, Grafana i Jaeger	239
Wymagania techniczne	240
Wdrożenie demonstracyjnej aplikacji Shiplt Clicker v7	240
Dzienniki kontenerów Docker i programów uruchomieniowych	243
Dzienniki kontenerów Docker	243
Cechy idealnego systemu zarządzania dziennikami	244
Diagnozowanie problemów z warstwą sterowania platformy Kubernetes na podstawie dzienników	245
Zapisywanie dzienników w usłudze CloudWatch Logs	246
Długotrwałe przechowywanie dzienników w usłudze S3	247
Analiza dzienników za pomocą usług CloudWatch Insights i Amazon Athena	248
Ćwiczenie — sprawdzenie liczby uruchomień gry Shiplt Clicker	249
Testy dostępności, gotowości i uruchamiania w platformie Kubernetes	249
Sprawdzanie za pomocą testów dostępności, czy kontener odpowiada na zapytania	250
Sprawdzanie za pomocą testów gotowości, czy usługa może przetwarzać ruch	250
Przystosowanie aplikacji Shiplt Clicker do osobnych testów dostępności i gotowości	251
Ćwiczenie — wymuszenie negatywnego wyniku testu gotowości aplikacji Shiplt Clicker	252
Zbieranie wskaźników i wysyłanie alarmów za pomocą systemu Prometheus	252
Historia systemu	253
Zapytania i interfejs WWW	253
Definiowanie w aplikacji wskaźników dla systemu Prometheus	254
Odczytywanie niestandardowych wskaźników	256
Konfiguracja alarmów	256
Wysyłanie powiadomień za pomocą modułu Alertmanager	258
Szczegóły zapytań i zewnętrznego monitoringu	260
Wizualizacja danych operacyjnych za pomocą systemu Grafana	260
Dostęp do systemu	260
Dodawanie paneli opracowanych przez społeczność użytkowników	261
Utworzenie nowego panelu z niestandardowym zapytaniem	262

Monitorowanie wydajności aplikacji za pomocą systemu Jaeger	264
Interfejs OpenTracing API	264
Wprowadzenie do systemu Jaeger	265
Instalacja klienta systemu Jaeger w aplikacji Shiplt Clicker	267
Instalacja rozszerzenia Jaeger Operator	270
Podsumowanie	272
Dalsza lektura	272
Rozdział 11. Skalowanie i testy obciążeniowe aplikacji w środowisku Docker	275
Wymagania techniczne	276
Nowa aplikacja Shiplt Clicker v8	276
Skalowanie klastra Kubernetes	278
Ręczne skalowanie klastra	279
Dynamiczne skalowanie klastra	281
Siatka usług Envoy i jej zastosowania	285
Zarządzanie ruchem w sieci	286
Instalacja siatki Envoy	287
Testowanie skalowalności i wydajności aplikacji za pomocą narzędzia k6	291
Rejestrowanie i odtwarzanie sesji	292
Ręczne tworzenie realistycznego testu	293
Wykonanie testu obciążeniowego	297
Podsumowanie	298
Dalsza lektura	299
Część III. Bezpieczeństwo kontenerów Docker	301
Rozdział 12. Wprowadzenie do bezpieczeństwa kontenerów	303
Wymagania techniczne	304
Wirtualizacja i modele bezpieczeństwa hiperwizora	304
Wirtualizacja i pierścienie ochronne	304
Platforma Docker i pierścienie ochronne	306
Kontenerowe modele bezpieczeństwa	308
Docker Engine, containerd i zabezpieczenia w systemie Linux	309
Przestrzeń PID	310
Przestrzeń MNT	311
Przestrzeń NET	311
Przestrzeń IPC	311
Przestrzeń UTS	311
Przestrzeń USER	312
Uwaga dotycząca grup cgroups	312
Dobre praktyki w skrócie	312
Regularnie instaluj poprawki	313
Zabezpieczaj gniazdo sieciowe	313
Nie uruchamiaj błędnego kodu	315
Zawsze twórz konto użytkownika z minimalnymi uprawnieniami	315
Podsumowanie	315

Rozdział 13. Podstawy bezpieczeństwa i dobre praktyki korzystania z platformy Docker	317
Wymagania techniczne	318
Bezpieczeństwo obrazów kontenerów	318
Weryfikacja obrazu	320
Minimalny obraz bazowy	322
Ograniczanie uprawnień	323
Zapobieganie wyciekowi danych	324
Bezpieczne korzystanie z poleceń w platformie Docker	326
Polecenia COPY i ADD — jaka jest różnica?	326
Kopiowanie rekurencyjne — bądź ostrożny	327
Bezpieczeństwo procesu budowania kontenerów	328
Wieloetapowy proces budowania obrazu	329
Ograniczanie możliwości i zasobów wdrażanego kontenera	330
Ograniczanie zasobów	330
Ograniczanie możliwości	331
Podsumowanie	332
 Rozdział 14. Zaawansowane zabezpieczenia: skrytki, poufne polecenia, znaczniki i etykiety	 333
Wymagania techniczne	334
Bezpieczne przechowywanie poufnych danych w platformie Docker	334
Dziennik Raft	335
Tworzenie, edytowanie i usuwanie skrytek	336
Tworzenie skrytek	336
Odczytywanie skrytek	336
Usuwanie skrytek	337
Skrytki w akcji — przykłady	338
Zabezpieczanie kontenerów za pomocą znaczników	340
Umieszczanie w etykietach metadanych aplikacji	341
Podsumowanie	342
 Rozdział 15. Skanowanie, monitorowanie i zewnętrzne narzędzia	 343
Wymagania techniczne	344
Skanowanie i monitorowanie a bezpieczeństwo kontenerów w chmurze i środowisku programistycznym	344
Skanowanie obrazów kontenerów za pomocą programu Anchore Engine	345
Chef InSpec	349
Lokalny monitoring platformy Docker za pomocą natywnego narzędzia stats	350
Agregowanie danych monitoringowych w chmurze za pomocą narzędzia Datadog	353
Zabezpieczanie kontenerów w chmurze AWS	356
Alarmy bezpieczeństwa w usłudze AWS GuardDuty	357
Zabezpieczanie kontenerów w chmurze Azure	358
Monitorowanie kontenerów w chmurze Azure	358
Zabezpieczanie kontenerów przy użyciu usługi Security Center	359

Zabezpieczanie kontenerów w chmurze GCP	360
Analiza kontenerów i uwierzytelnienie binarne	361
Wykrywanie ataków za pomocą usługi Security Command Center	362
Podsumowanie	363
Dalsza lektura	364
 Rozdział 16. Wnioski — koniec drogi, ale nie podróży	 365
Wymagania techniczne	365
Kontenery w skrócie	366
Czego się dowiedziałeś o tworzeniu aplikacji	366
Wzorce projektowe	366
Poszerzenie wiedzy o tworzeniu i utrzymaniu aplikacji	369
Inżynieria chaosu i tworzenie niezawodnych systemów produkcyjnych	369
Bezpieczeństwo i dalsze kroki	371
Metasploit i testy penetracyjne	371
Podsumowanie	373