
Spis treści

Przedmowa	13
1. Podstawy Pythona dla DevOps	21
Instalowanie i uruchamianie Pythona	21
Powłoka Pythona	22
Jupyter Notebooks	23
Programowanie proceduralne	23
Zmienne	24
Podstawowe operacje arytmetyczne	24
Komentarze	25
Funkcje wbudowane	25
Print	25
Range	26
Sterowanie przepływem kodu	26
If-elif-else	26
Pętle for	28
Pętle while	29
Obsługa wyjątków	29
Obiekty wbudowane	30
Czym jest obiekt?	30
Metody i atrybuty obiektu	31
Sekuencje	31
Funkcje	42
Anatomia funkcji	43
Funkcje jako obiekty	44
Funkcje anonimowe	45
Korzystanie z wyrażeń regularnych	45
Wyszukiwanie	46
Zbiory znaków	46
Klasy znaków	47

Grupy	47
Grupy nazwane	48
Znajdź wszystko	48
Iterator wyszukiwania	48
Podstawianie	49
Kompilowanie	49
Leniwe wartościowanie	50
Generatory	50
Generatory składane	51
Dodatkowe funkcjonalności IPython	51
Korzystanie z IPython do uruchamiania poleceń powłoki Unix	51
Ćwiczenia	53
2. Automatyzacja zadań dotyczących plików i systemu plików	55
Odczytywanie i zapisywanie plików	55
Korzystanie z wyrażeń regularnych do wyszukiwania tekstu	63
Przetwarzanie dużych plików	65
Szyfrowanie tekstu	66
Haszowanie z wykorzystaniem pakietu hashlib	66
Szyfrowanie z wykorzystaniem biblioteki cryptography	67
Moduł os	68
Zarządzanie plikami i katalogami za pomocą modułu os.path	69
Przeglądanie drzew katalogów za pomocą funkcji os.walk	73
Ścieżki jako obiekty modułu pathlib	73
3. Praca w wierszu polecenia	75
Praca w środowisku powłoki	75
Komunikacja z interpreterem za pomocą modułu sys	75
Wykonywanie zadań w systemie operacyjnym z wykorzystaniem modułu os	76
Inicjowanie procesów za pomocą modułu subprocess	77
Tworzenie narzędzi wiersza polecenia	78
Atrybut sys.argv	80
argparse	82
Pakiet click	85
Moduł fire	90
Implementowanie wtyczek	94
Studium przypadku: Turbodoładowanie Pythona	
za pomocą narzędzi wiersza polecenia	95
Kompilator Just-in-Time (JIT) Numba	96
Korzystanie z GPU w Pythonie za pomocą frameworka CUDA	97

Uruchamianie w Pythonie kodu na wielu rdzeniach i w wielu wątkach	98
z wykorzystaniem modułu Numba	100
Klasteryzacja z wykorzystaniem modułu KMeans	101
Ćwiczenia	101
4. Przydatne narzędzia systemu Linux	103
Narzędzia dyskowe	104
Pomiar wydajności	104
Partycje	106
Odczytywanie specyficznych informacji o urządzeniu	107
Narzędzia sieciowe	108
Tunelowanie SSH	108
Pomiar wydajności HTTP za pomocą Apache Benchmark (ab)	109
Testowanie obciążenia za pomocą narzędzia molotov	110
Narzędzia do badania CPU	112
Przeglądanie procesów za pomocą htop	113
Korzystanie z Bash i ZSH	115
Personalizacja powłoki Pythona	116
Rekurencyjny globbing	116
Wyszukiwanie i zamiana z pytaniami o potwierdzenie	117
Usuwanie tymczasowych plików Pythona	118
Wyświetlanie listy procesów i jej filtrowanie	118
Uniksowe znaczniki czasu	119
Łączenie możliwości Pythona z powłoką Bash i ZSH	120
Generator losowych haseł	120
Czy mój moduł istnieje?	121
Zmiana katalogów na ścieżki do modułów	121
Konwersja pliku CSV na JSON	122
Jednowierszowe skrypty w Pythonie	123
Debugery	123
Jak szybko działa ten fragment kodu?	124
strace	125
Ćwiczenia	127
Zadanie związane ze studium przypadku	127
5. Zarządzanie pakietami	129
Dlaczego tworzenie pakietów jest ważne?	130
Kiedy tworzenie pakietu może być niepotrzebne?	130
Wytyczne dotyczące tworzenia pakietów	130
Opisowe wersjonowanie	131
Rejestr zmian	132
Wybór strategii	133

Sposoby tworzenia pakietów	133
Natywny pakiet Pythona	133
Pakiety w stylu Debiana	139
Pakiety RPM	145
Zarządzanie za pomocą systemd	151
Procesy długotrwałe	152
Konfiguracja	152
Plik modułu systemd	154
Instalacja modułu	155
Obsługa logów	156
Ćwiczenia	158
Zadanie związane ze studium przypadku	158
6. Continuous Integration i Continuous Deployment	159
Studium przypadku: konwersja źle utrzymywanej witryny	
bazującej na WordPressie do Hugo	159
Konfigurowanie Hugo	160
Konwersja witryny WordPress na posty Hugo	161
Utworzenie indeksu Algolia i jego uaktualnienie	163
Automatyzacja za pomocą Makefile	165
Instalacja z wykorzystaniem AWS CodePipeline	165
Studium przypadku: instalacja aplikacji Python App Engine	
za pomocą mechanizmu Google Cloud Build	166
Studium przypadku: NFSOPS	173
7. Monitorowanie i logowanie	175
Kluczowe pojęcia dotyczące budowania niezawodnych systemów	175
Niezmienne zasady DevOps	176
Centralne logowanie	176
Studium przypadku: produkcyjna baza danych zabija dyski twarde	177
Zbudować czy kupić?	178
Odporność na błędy	178
Monitorowanie	180
Graphite	180
StatsD	181
Prometheus	181
Oprządkowanie	185
Konwencje nazewnictwa	188
Logowanie	189
Dlaczego konfigurowanie logowania jest trudne?	189
basicconfig	189

Głębsza konfiguracja	190
Powszechne wzorce	194
Stos ELK	195
Logstash	197
Elasticsearch i Kibana	198
Ćwiczenia	201
Zadanie związane ze studium przypadku	202
8. Pytest dla DevOps	203
Testowanie za pomocą frameworka pytest	203
Pierwsze kroki z pytest	204
Testowanie z wykorzystaniem pytest	204
Różnice w stosunku do unittest	206
Cechy frameworka pytest	207
conftest.py	208
Niezwyczajna funkcja assert	208
Parametryzacja	209
Fikstury	211
Pierwsze kroki	211
Fikstury wbudowane	213
Testowanie infrastruktury	215
Co to jest walidacja systemowa?	216
Wprowadzenie do projektu Testinfra	217
Nawiązywanie połączeń ze zdalnymi węzłami	217
Funkcje i fikstury specjalne	220
Przykłady	222
Testowanie notatników Jupyter Notebooks z wykorzystaniem frameworka pytest	224
Ćwiczenia	225
Zadanie związane ze studium przypadku	225
9. Chmura obliczeniowa	227
Podstawy chmury obliczeniowej	228
Rodzaje chmur obliczeniowych	230
Rodzaje usług chmury obliczeniowej	231
IaaS	231
MaaS	235
PaaS	235
Przetwarzanie bezserwerowe	236
SaaS	239
Infrastruktura jako kod	239
Ciągłe dostawy	240

Wirtualizacja i kontenery	240
Wirtualizacja sprzętowa	240
Sieci SDN	241
Magazyny SDS	241
Kontenery	241
Wyzwania i możliwości przetwarzania rozproszonego	243
Współbieżność, wydajność i zarządzanie procesami w dobie chmury obliczeniowej	244
Zarządzanie procesami	245
Zarządzanie procesami z wykorzystaniem modułu subprocess	245
Korzystanie z modułu multiprocessing do rozwiązywania problemów	247
Rozwidlanie procesów za pomocą funkcji pool()	248
FaaS i tryb bezserwerowy	250
Wysokowydajny Python z wykorzystaniem pakietu Numba	251
Korzystanie z kompilatora Just in Time biblioteki Numba	251
Korzystanie z wysokowydajnych serwerów	252
Wniosek	252
Ćwiczenia	253
Studia przypadków	253

10. Infrastruktura jako kod 255

Klasyfikacja narzędzi automatyzacji infrastruktury	256
Dostarczanie ręczne	258
Automatyczne dostarczanie infrastruktury z wykorzystaniem systemu Terraform	259
Dostarczanie komory S3	259
Dostarczanie certyfikatu SSL z usługi AWS ACM	262
Dostarczanie dystrybucji Amazon CloudFront	263
Dostarczanie rekordu DNS Route 53	265
Kopiowanie statycznych plików do usługi S3	267
Usuwanie wszystkich zasobów AWS dostarczonych przez Terraform	267
Zautomatyzowane dostarczanie infrastruktury za pomocą systemu Pulumi	267
Tworzenie nowego projektu Pythona Pulumi dla usług AWS	268
Tworzenie wartości konfiguracyjnych dla stosu staging	272
Dostarczanie certyfikatu SSL ACM	273
Dostarczanie strefy Route 53 i rekordów DNS	273
Dostarczanie dystrybucji CloudFront	275
Dostarczanie rekordu DNS Route 53 dla adresu URL witryny	277
Tworzenie i wdrażanie nowego stosu	277
Ćwiczenia	279

11. Technologie kontenerowe: Docker i Docker Compose	281
Czym jest kontener Dockera?	282
Tworzenie, budowanie, uruchamianie i usuwanie obrazów i kontenerów Dockera	282
Publikowanie obrazów Dockera w Rejestrze Dockera	285
Uruchamianie kontenera Dockera z tego samego obrazu na innym hoście	287
Uruchamianie wielu kontenerów Dockera za pomocą systemu Docker Compose	289
Przenoszenie usług docker-compose do nowego hosta i systemu operacyjnego	299
Ćwiczenia	302
12. Orkiestracja kontenerów: Kubernetes	305
Przegląd pojęć związanych z systemem Kubernetes	306
Korzystanie z systemu Kompose do tworzenia manifestów Kubernetesa na podstawie pliku docker-compose.yaml	307
Instalacja manifestów Kubernetesa w lokalnym klastrze Kubernetesa z wykorzystaniem minikube	308
Uruchomienie klastra GKE Kubernetes w GCP za pomocą Pulumi	322
Instalacja przykładowej aplikacji Flask do GKE	324
Instalacja wykresów Helm Prometheus i Grafana	330
Niszczanie klastra GKE	334
Ćwiczenia	335
13. Technologie bezserwerowe	337
Wdrażanie tej samej funkcji Pythona do chmur dostawców z Wielkiej Trójki	339
Instalacja frameworka Serverless	340
Wdrażanie funkcji Pythona w usłudze AWS Lambda	340
Wdrażanie funkcji Pythona na platformie Google Cloud Functions	342
Wdrażanie funkcji Pythona w usłudze AWS Lambda	348
Wdrażanie funkcji Pythona do platform FaaS działających w trybie self-hosted	351
Wdrażanie funkcji Pythona do usługi OpenFaaS	352
Konfigurowanie tabeli DynamoDB, funkcji Lambda i metod API Gateway za pomocą AWS CDK	358
Ćwiczenia	376
14. MLOps i inżynieria uczenia maszynowego	377
Czym jest uczenie maszynowe?	377
Nadzorowane uczenie maszynowe	377
Modelowanie	380
Ekosystem uczenia maszynowego w Pythonie	382
Uczenie głębokie z wykorzystaniem frameworka PyTorch	383
Platformy uczenia maszynowego w chmurze	386

Model dojrzałości uczenia maszynowego	387
Najważniejsza terminologia uczenia maszynowego	388
Poziom 1. Formułowanie, identyfikowanie zakresu i definiowanie problemu	389
Poziom 2. Ciągłe dostawy danych	389
Poziom 3. Ciągłe dostawy oczyszczonych danych	391
Poziom 4. Ciągłe dostawy eksploracyjnych analiz danych	392
Poziom 5. Ciągłe dostarczania tradycyjnych narzędzi ML i AutoML	393
Poziom 6. Operacyjna pętla sprzężenia zwrotnego narzędzi ML	393
Model Sklearn Flask z wykorzystaniem systemów Kubernetes i Docker	394
Sklearn Flask z wykorzystaniem Kubernetesa i Dockera	397
EDA	398
Modelowanie	399
Dostrajanie skalowanego algorytmu GBM	399
Dopasowywanie modelu	400
Ocena	401
adhoc_predict	401
Przepływ pracy JSON	402
Skalowanie danych wejściowych	402
adhoc_predict z modułu Pickle	403
Skalowanie danych wejściowych	404
Ćwiczenia	404
Zadanie związane ze studium przypadku	405
Pytania i zadania kontrolne	405
15. Inżynieria danych	407
Small data	408
Obsługa plików typu small data	408
Zapis do pliku	409
Odczyt z pliku	409
Potok generatora używany w celu czytania i przetwarzania wierszy	409
Korzystanie z formatu YAML	410
Big Data	411
Narzędzia Big Data, komponenty i platformy	413
Źródła danych	413
Systemy plików	414
Przechowywanie danych	415
Pobieranie strumieni danych w czasie rzeczywistym	416
Studium przypadku: budowanie własnego potoku danych	417
Inżynieria danych w trybie bezserwerowym	418
Korzystanie z usługi AWS Lambda z wykorzystaniem zdarzeń CloudWatch	418
Logowanie z wykorzystaniem usług Amazon CloudWatch i AWS Lambda	419

Wykorzystanie usługi AWS Lambda w celu zapełniania kolejki w usłudze Amazon Simple Queue Service	420
Konfiguracja mechanizmu wyzwalającego zdarzenie CloudWatch	421
Tworzenie funkcji Lambda sterowanych zdarzeniami	422
Odczyt zdarzeń Amazon SQS z funkcji AWS Lambda	422
Wnioski	429
Ćwiczenia	429
Zadanie związane ze studium przypadku	430
16. Historie wojenne DevOps i wywiady	431
Studio filmowe nie może produkować filmów	432
Studio gier nie może opublikować gry	434
Uruchomienie skryptów Pythona zajmuje 60 sekund	435
Gaszenie pożarów za pomocą pamięci podręcznej i inteligentnej instrumentacji	437
Automatyzacja zabierze Ci pracę!	437
Antywzorce DevOps	439
Brak zautomatyzowanego serwera budowania	439
„Latanie po omacku”	439
Trudności w koordynacji jako stan ciągły	439
Brak pracy zespołowej	441
Wywiady	445
Glenn Solomon	445
Andrew Nguyen	446
Gabriella Roman	448
Rigoberto Roche	449
Jonathan LaCour	450
Ville Tuulos	452
Joseph Reis	454
Teijo Holzer	455
Matt Harrison	457
Michael Foord	458
Zalecenia	461
Ćwiczenia	462
Wyzwania	462
Projekt końcowy	462