

Zautomatyzuj swoją firmę z Pythonem : praktyczne rozwiązania dla firmowej sieci / Bassem Aly. – Gliwice, copyright © 2019

Spis treści

O autorze	9
O recenzencie	10
Przedmowa	11
Rozdział 1. Przygotowanie środowiska pracy	17
Wprowadzenie do języka Python	17
Wersje języka Python	18
Czy to oznacza, że nie mogę napisać programu, który będzie działał jednocześnie dla Pythona w wersji 2 i 3?	19
Instalacja języka Python	20
Instalacja PyCharm IDE	22
Konfiguracja projektu za pomocą PyCharma	25
Właściwości środowiska PyCharm	29
Debugowanie kodu	29
Refaktoryzacja kodu	30
Instalacja pakietów za pośrednictwem GUI	32
Podsumowanie	34
Rozdział 2. Biblioteki stosowane do automatyzacji zadań	35
Pakiety Pythona	35
Ścieżki wyszukiwania pakietów	36
Podstawowe biblioteki Pythona	37
Biblioteki sieciowe	37
Biblioteki do obsługi systemu i chmury	39
Dostęp do kodu źródłowego modułu	40
Wizualizacja kodu Pythona	42
Podsumowanie	45
Rozdział 3. Konfigurowanie sieciowego środowiska laboratoryjnego	47
Wymagania techniczne	47
Kiedy i jak zautomatyzować zadania w sieci?	48
Dlaczego potrzebujemy automatyzacji?	48
Screen scraping czy API — czego używać w automatyzacji?	48
Dlaczego warto wykorzystać Python do automatyzacji zadań sieciowych?	49
Przyszłość automatyzacji zadań sieciowych	50

Konfiguracja laboratorium	51
Instalacja EVE-NG	51
Instalacja na VMware Workstation	52
Instalacja poprzez VMware ESXi	54
Instalacja poprzez Red Hat KVM	55
Dostęp do EVE-NG	56
Instalacja pakietu EVE-NG dla klienta	60
Ładowanie obrazów do EVE-NG	61
Budowanie topologii sieci	61
Dodanie nowych węzłów	62
Łączenie węzłów	63
Podsumowanie	65

Rozdział 4. Zarządzanie urządzeniami sieciowymi za pomocą języka Python

Wymagania techniczne	68
Python i SSH	68
Moduł Paramiko	68
Moduł Netmiko	71
Wykorzystanie protokołu Telnet za pomocą Pythona	77
Zmiana konfiguracji poprzez telnetlib	80
Praca z sieciami z wykorzystaniem biblioteki netaddr	82
Instalowanie modułu netaddr	82
Metody modułu netaddr	83
Przykładowe przypadki użycia	85
Konfiguracja kopii zapasowej urządzenia	85
Utworzenie własnego terminala dostępowego	88
Odczyt danych z arkusza Excela	90
Więcej przykładów	92
Podsumowanie	93

Rozdział 5. Pobieranie użytecznych informacji z urządzeń sieciowych

Wymagania techniczne	96
Zasada działania parserów	96
Wprowadzenie do wyrażeń regularnych	96
Tworzenie wyrażeń regularnych za pomocą Pythona	98
Audyt konfiguracji za pomocą biblioteki CiscoConfParse	104
Biblioteka CiscoConfParse	104
Wspierani producenci	105
Instalacja biblioteki CiscoConfParse	105
Praca z biblioteką CiscoConfParse	106
Wizualizacja danych za pomocą biblioteki Matplotlib	108
Instalacja biblioteki Matplotlib	109
Ćwiczenia z biblioteką Matplotlib	109

Wizualizacja danych protokołu SNMP za pomocą biblioteki Matplotlib	112
Podsumowanie	113
Rozdział 6. Tworzenie konfiguracji przy użyciu języków Python i Jinja2	115
Co to jest YAML?	115
Formatowanie plików YAML	116
Tworzenie konfiguracji przy użyciu Jinja2	119
Odczyt szablonów z pliku	126
Używanie pętli i operacji warunkowych w Jinja2	127
Podsumowanie	135
Rozdział 7. Równoległe wykonywanie skryptu w języku Python	137
W jaki sposób system operacyjny wykonuje kod Pythona?	137
Biblioteka Pythona do przetwarzania wieloprotocowego	139
Pierwsze kroki z przetwarzaniem wieloprotocowym	140
Komunikacja wewnątrzprocesowa	143
Podsumowanie	144
Rozdział 8. Przygotowanie środowiska laboratoryjnego	145
Jak uzyskać obraz systemu operacyjnego?	145
Pobieranie dystrybucji CentOS	146
Pobieranie dystrybucji Ubuntu	146
Tworzenie maszyny do automatyzacji za pomocą hipernadzorcy	147
Tworzenie maszyny Linuxowej za pomocą VMware ESXi	147
Tworzenie maszyny Linuxowej za pomocą KVM	152
Pierwsze kroki z Cobblerem	156
Jak działa Cobbler?	156
Instalacja Cobblera na serwerze automatyzacji	157
Udostępnianie serwerów za pośrednictwem Cobblera	160
Podsumowanie	165
Rozdział 9. Moduł subprocess	167
Klasa Popen()	167
Odczyt z stdin, stdout i stderr	170
Funkcja call()	173
Podsumowanie	174
Rozdział 10. Uruchamianie zadań związanych z administracją systemu za pomocą biblioteki Fabric	175
Wymagania techniczne	176
Co to jest Fabric?	176
Instalacja	177
Operacje biblioteki Fabric	178
Uruchomienie pierwszego pliku Fabric	180

Więcej na temat narzędzia fab	183
Sprawdzanie stanu systemu za pomocą narzędzia Fabric	184
Inne przydatne właściwości modułu Fabric	188
Role	188
Menadżery kontekstu	189
Podsumowanie	191
Rozdział 11. Generowanie raportów i monitorowanie systemu	193
Zbieranie danych w systemie Linux	193
Wysyłanie e-mailem wygenerowanych danych	198
Wykorzystanie modułów obsługujących datę i czas	200
Regularne uruchamianie skryptu	202
Zarządzanie użytkownikami za pomocą Ansible	203
Linux	203
Microsoft Windows	204
Podsumowanie	205
Rozdział 12. Współpraca z bazą danych	207
Instalacja MySQL na serwerze automatyzacji	207
Zabezpieczanie zainstalowanej aplikacji	208
Weryfikacja instalacji bazy danych	209
Dostęp do bazy danych MySQL z poziomu języka Python	210
Wysyłanie zapytań do bazy danych	212
Wstawianie rekordów do bazy	213
Podsumowanie	216
Rozdział 13. Administracja systemem za pomocą Ansible	217
Terminologia Ansible	218
Instalacja Ansible w systemie Linux	219
Systemy RHEL i CentOS	219
Ubuntu	219
Korzystanie z Ansible w trybie ad hoc	220
Jak działa Ansible?	223
Tworzenie pierwszego playbooka	224
Warunki, uchwyt i pętle Ansible	226
Tworzenie warunków	226
Tworzenie pętli w Ansible	229
Uruchamianie zadań za pomocą uchwytów	230
Praca z faktami Ansible	231
Praca z szablonami Ansible	232
Podsumowanie	234
Rozdział 14. Tworzenie maszyn wirtualnych VMware i zarządzanie nimi	235
Konfigurowanie środowiska laboratoryjnego	235

Tworzenie pliku VMX za pomocą Jinja2	238
Budowa szablonu VMX	238
Obsługa danych z Excela	241
Generowanie plików VMX	243
Pythonowe klienty VMware	250
Instalacja PyVmomi	251
Pierwsze kroki z PyVmomi	252
Zmiana stanu maszyny wirtualnej	256
Więcej przykładów	257
Zarządzanie instancjami za pomocą playbooków Ansible	257
Podsumowanie	260
Rozdział 15. Współpraca z API OpenStack	261
Działanie usług sieciowych RESTful	262
Konfigurowanie środowiska pracy	263
Instalacja pakietu rdo-OpenStack	264
Tworzenie pliku odpowiedzi	264
Edycja pliku odpowiedzi	265
Uruchomienie packstat	265
Dostęp do GUI OpenStacka	265
Wysyłanie żądań do OpenStacka	266
Tworzenie instancji za pomocą języka Python	269
Tworzenie obrazu	269
Ustawianie konfiguracji serwera (ustawianie flavorów)	271
Tworzenie sieci i podsieci	272
Uruchamianie instancji	274
Zarządzanie instancjami OpenStacka za pomocą Ansible	275
Instalacja biblioteki Shade oraz Ansible	276
Tworzenie playbooka Ansible	276
Podsumowanie	279
Rozdział 16. Automatyzacja usług AWS za pomocą Boto3	281
Moduły Pythona do obsługi AWS	281
Instalacja Boto3	282
Zarządzanie instancjami AWS	284
Usuwanie instancji	285
Automatyzowanie usług AWS S3	286
Tworzenie kubełków	286
Ładowanie pliku do kubełka	287
Usuwanie kubełka	287
Podsumowanie	288
Rozdział 17. Framework Scapy	289
Zasada działania frameworku Scapy	289
Instalacja frameworku Scapy	290

Systemy Unixowe	290
Wsparcie dla systemów Windows i macOS	291
Tworzenie pakietów za pomocą frameworku Scapy	291
Przechwytywanie i modyfikowanie pakietów	296
Wstrzykiwanie danych do pakietów	297
Podsluchiwanie pakietów	299
Zapisywanie pakietów do pliku pcap	301
Podsumowanie	301

Rozdział 18. Budujemy skaner sieciowy za pomocą języka

Python	303
Zasada działania skanera sieciowego	303
Budujemy skaner sieciowy za pomocą języka Python	304
Ulepszanie kodu	305
Skanowanie usług	307
Współdzielenie kodu za pomocą GitHuba	310
Tworzenie konta na GitHubie	311
Tworzenie i ładowanie kodu	311
Podsumowanie	316

Skorowidz	317
------------------	------------