

Python для сетевых инженеров

Третье издание



Автоматизация сети,
программирование и DevOps

Эрик Чоу



 ПИТЕР®

Санкт-Петербург • Москва • Минск

2023

Краткое содержание

| | |
|--|-----|
| Предисловие | 16 |
| Введение | 18 |
| Об авторе | 20 |
| О научном редакторе | 21 |
| Вступление | 22 |
| Глава 1. Обзор TCP/IP и Python | 27 |
| Глава 2. Низкоуровневое взаимодействие с сетевыми устройствами | 61 |
| Глава 3. API и IDN-сети | 100 |
| Глава 4. Основы Ansible | 142 |
| Глава 5. Ansible: следующий уровень | 177 |
| Глава 6. Сетевая безопасность с использованием Python | 211 |
| Глава 7. Сетевой мониторинг с использованием Python: часть 1 | 240 |
| Глава 8. Сетевой мониторинг с использованием Python: часть 2 | 270 |
| Глава 9. Создание сетевых веб-сервисов с помощью Python | 305 |
| Глава 10. Облачные сетевые технологии AWS | 338 |
| Глава 11. Облачные сетевые технологии Azure | 373 |
| Глава 12. Анализ сетевых данных с помощью Elastic Stack | 412 |
| Глава 13. Работа с Git | 446 |
| Глава 14. Непрерывная интеграция с помощью Jenkins | 474 |
| Глава 15. TDD для сетей | 499 |

Оглавление

| | |
|---|-----------|
| Предисловие | 16 |
| Введение | 18 |
| Об авторе | 20 |
| О научном редакторе | 21 |
| Вступление | 22 |
| Кому подойдет эта книга | 23 |
| Какие темы здесь освещаются | 23 |
| Как извлечь максимум из этой книги | 25 |
| Загрузка файлов с примерами кода | 25 |
| Загрузка полноцветных иллюстраций | 26 |
| Условные обозначения | 26 |
| От издательства | 26 |
| Глава 1. Обзор TCP/IP и Python | 27 |
| Краткий обзор интернета | 29 |
| Серверы, хосты и сетевые компоненты | 30 |
| Появление дата-центров | 31 |
| Модель OSI | 35 |
| Клиент-серверная модель | 37 |
| Наборы сетевых протоколов | 38 |
| Протокол управления передачей (TCP) | 39 |
| Протокол пользовательских датаграмм (UDP) | 40 |
| Межсетевой протокол (IP) | 41 |
| Обзор языка Python | 43 |
| Версии Python | 45 |
| Операционные системы | 46 |

| | |
|---|-----------|
| Выполнение программы на Python | 47 |
| Встроенные в Python типы данных | 48 |
| Операторы в Python | 54 |
| Средства управления потоком выполнения в Python | 55 |
| Функции в Python | 56 |
| Классы в Python | 57 |
| Модули и пакеты в Python | 58 |
| Резюме | 60 |
| Глава 2. Низкоуровневое взаимодействие с сетевыми устройствами | 61 |
| Трудности работы с CLI | 62 |
| Создание виртуальной лаборатории | 64 |
| Физические устройства | 64 |
| Виртуальные устройства | 65 |
| Cisco VIRL | 66 |
| Cisco DevNet и dCloud | 71 |
| GNS3 | 73 |
| Библиотека Python Pexrest | 75 |
| Виртуальная среда Python | 75 |
| Установка Pexrest | 76 |
| Краткий обзор Pexrest | 76 |
| Наша первая программа на основе Pexrest | 81 |
| Другие возможности Pexrest | 82 |
| Pexrest и SSH | 84 |
| Итоговая программа на основе Pexrest | 85 |
| Библиотека Python Paramiko | 86 |
| Установка Paramiko | 86 |
| Краткий обзор Paramiko | 87 |
| Наша первая программа, написанная с использованием Paramiko | 90 |
| Другие возможности Paramiko | 91 |
| Итоговая программа на основе Paramiko | 93 |
| Библиотека Netmiko | 94 |
| Фреймворк Nornir | 96 |
| Недостатки Pexrest и Paramiko по сравнению с другими инструментами | 98 |
| Резюме | 99 |

| | |
|---|------------|
| Глава 3. API и IDN-сети | 100 |
| Инфраструктура как код | 101 |
| Сети, ориентированные на намерения | 102 |
| Консольный вывод и структурированные результаты API-запроса | 103 |
| Моделирование данных для IaC | 106 |
| YANG и NETCONF | 108 |
| API и платформа ACI от Cisco | 108 |
| Cisco NX-API | 109 |
| Модель Cisco YANG | 115 |
| Cisco ACI и APIC-EM | 116 |
| Контроллер Cisco Meraki | 119 |
| API на языке Python для Juniper Networks | 120 |
| Juniper и NETCONF | 121 |
| Juniper PyEZ для разработчиков | 125 |
| API на языке Python для устройств Arista | 130 |
| Работа с eAPI от Arista | 130 |
| Библиотека Arista Pyeapi | 135 |
| Пример работы с VyOS | 140 |
| Другие библиотеки | 141 |
| Резюме | 141 |
| Глава 4. Основы Ansible | 142 |
| Ansible: более декларативный фреймворк | 143 |
| Короткий пример с Ansible | 146 |
| Установка управляющего узла | 146 |
| Установка разных версий Ansible из исходного кода | 147 |
| Подготовка лаборатории | 148 |
| Ваш первый сценарий Ansible | 149 |
| Преимущества Ansible | 153 |
| Отсутствие агентов | 154 |
| Идемпотентность | 155 |
| Простота и расширяемость | 155 |
| Поддержка от производителей сетевого оборудования | 156 |
| Архитектура Ansible | 158 |
| YAML | 159 |
| Файлы реестров | 159 |

| | |
|--|------------|
| Переменные | 161 |
| Шаблоны Jinja2 | 165 |
| Сетевые модули Ansible | 165 |
| Локальные соединения и факты | 166 |
| Переменная provider | 166 |
| Пример Ansible с устройствами Cisco | 168 |
| Пример сценария для Ansible 2.8 | 171 |
| Пример Ansible с устройствами Juniper | 174 |
| Пример Ansible с устройствами Arista | 175 |
| Резюме | 176 |
| Глава 5. Ansible: следующий уровень | 177 |
| Подготовка лаборатории | 178 |
| Условные выражения в Ansible | 178 |
| Выражение when | 179 |
| Факты о сетевых устройствах в Ansible | 181 |
| Условные выражения в сетевых модулях | 184 |
| Циклы в Ansible | 185 |
| Стандартные циклы | 186 |
| Циклический перебор словарей | 188 |
| Шаблоны | 190 |
| Переменные в шаблонах Jinja2 | 192 |
| Циклы в Jinja2 | 193 |
| Условные выражения в Jinja2 | 193 |
| Переменные групп и хостов | 196 |
| Переменные группы | 196 |
| Переменные хоста | 197 |
| Ansible Vault | 198 |
| Подключение файлов и роли в Ansible | 200 |
| Инструкции include в Ansible | 201 |
| Роли Ansible | 202 |
| Написание собственного модуля | 206 |
| Ваш первый модуль | 206 |
| Ваш второй модуль | 208 |
| Резюме | 210 |

| | |
|--|-----|
| Глава 6. Сетевая безопасность с использованием Python | 211 |
| Подготовка лаборатории | 212 |
| Python Scapy | 216 |
| Установка Scapy | 216 |
| Интерактивные примеры | 218 |
| Захват пакетов с помощью Scapy | 220 |
| Сканирование TCP-портов | 221 |
| Коллекция пакетов для проверки связи | 225 |
| Распространенные атаки | 226 |
| Ресурсы о Scapy | 226 |
| Списки доступа | 227 |
| Реализация списков доступа с помощью Ansible | 228 |
| Списки доступа по MAC-адресам | 231 |
| Поиск в Syslog | 233 |
| Поиск с помощью модуля регулярных выражений | 234 |
| Другие инструменты | 236 |
| Приватные VLAN | 236 |
| UFW и Python | 237 |
| Дополнительный материал | 238 |
| Резюме | 239 |
| Глава 7. Сетевой мониторинг с использованием Python: часть 1 | 240 |
| Подготовка лаборатории | 241 |
| SNMP | 242 |
| Подготовка | 244 |
| PySNMP | 246 |
| Python для визуализации данных | 251 |
| Matplotlib | 252 |
| Pygal | 259 |
| Работа с Sastic в Python | 264 |
| Установка | 265 |
| Сценарий на Python в качестве источника данных | 267 |
| Резюме | 269 |
| Глава 8. Сетевой мониторинг с использованием Python: часть 2 | 270 |
| Graphviz | 271 |
| Подготовка лаборатории | 272 |

| | |
|--|------------|
| Установка | 274 |
| Примеры работы с Graphviz | 274 |
| Примеры с Graphviz и Python | 277 |
| Создание графа ближайших соседей с помощью LLDP | 278 |
| Потоковый мониторинг | 287 |
| Разбор NetFlow с помощью Python | 288 |
| Мониторинг трафика с помощью ntop | 293 |
| Расширение ntop с помощью Python | 296 |
| sFlow | 300 |
| Резюме | 304 |
| Глава 9. Создание сетевых веб-сервисов с помощью Python | 305 |
| Сравнение веб-фреймворков для Python | 307 |
| Flask и подготовка лаборатории | 309 |
| Введение в фреймворк Flask | 310 |
| Клиент HTTPie | 312 |
| Маршрутизация URL | 313 |
| URL-переменные | 314 |
| Генерация URL | 316 |
| Возвращение результата с помощью jsonify | 317 |
| API для сетевых ресурсов | 318 |
| Flask-SQLAlchemy | 318 |
| API для работы с содержимым сети | 320 |
| API для работы с устройствами | 323 |
| API для работы с отдельными устройствами | 325 |
| Динамические сетевые операции | 326 |
| Асинхронные операции | 328 |
| Аутентификация и авторизация | 331 |
| Выполнение Flask в контейнерах | 333 |
| Резюме | 337 |
| Глава 10. Облачные сетевые технологии AWS | 338 |
| Подготовка к работе с AWS | 339 |
| AWS CLI и Python SDK | 340 |
| Обзор сети AWS | 344 |
| Виртуальное частное облако | 351 |
| Таблицы и цели маршрутизации | 355 |

| | |
|--|------------|
| Автоматизация с использованием CloudFormation | 357 |
| Группы безопасности и списки доступа к сети | 361 |
| Elastic IP | 363 |
| NAT-шлюзы | 364 |
| Direct Connect и VPN | 366 |
| VPN-шлюзы | 366 |
| Direct Connect | 367 |
| Сервисы для масштабирования сетей | 368 |
| Elastic Load Balancing | 369 |
| Сервис Route 53 DNS | 370 |
| Доставка содержимого с использованием CloudFront | 370 |
| Другие сетевые сервисы от AWS | 371 |
| Резюме | 371 |
| Глава 11. Облачные сетевые технологии Azure | 373 |
| Сравнение сетевых сервисов в Azure и AWS | 374 |
| Подготовка к работе с Azure | 375 |
| Администрирование Azure и API | 378 |
| Субъекты-службы в Azure | 381 |
| Сравнение Python и PowerShell | 383 |
| Глобальная инфраструктура Azure | 384 |
| Виртуальные сети Azure | 386 |
| Доступ к интернету | 389 |
| Создание сетевых ресурсов | 392 |
| Конечные точки сервисов для VNet | 394 |
| VNet-пиринг | 395 |
| Маршрутизация в виртуальных сетях | 397 |
| Сетевые группы безопасности | 402 |
| Azure VPN | 405 |
| Azure ExpressRoute | 408 |
| Сетевые балансировщики нагрузки в Azure | 409 |
| Другие сетевые сервисы Azure | 411 |
| Резюме | 411 |
| Глава 12. Анализ сетевых данных с помощью Elastic Stack | 412 |
| Что такое Elastic Stack | 413 |
| Топология лаборатории | 415 |

| | |
|--|------------|
| Elastic Stack как услуга | 420 |
| Первый полный пример | 421 |
| Elasticsearch и клиент на языке Python | 425 |
| Прием данных с помощью Logstash | 427 |
| Прием данных с использованием Beats | 430 |
| Поиск с помощью Elasticsearch | 435 |
| Визуализация данных с использованием Kibana | 440 |
| Резюме | 445 |
| Глава 13. Работа с Git | 446 |
| Git и разные аспекты управления контентом | 447 |
| Введение в Git | 448 |
| Преимущества Git | 449 |
| Терминология Git | 450 |
| Git и GitHub | 451 |
| Подготовка Git к работе | 451 |
| Gitignore | 452 |
| Примеры работы с Git | 454 |
| Ветви в Git | 458 |
| Пример работы с GitHub | 460 |
| Git и Python | 467 |
| GitPython | 467 |
| PyGitHub | 468 |
| Автоматизация резервного копирования конфигурационных файлов | 470 |
| Совместная работа с использованием Git | 472 |
| Резюме | 473 |
| Глава 14. Непрерывная интеграция с помощью Jenkins | 474 |
| Традиционный процесс управления изменениями | 475 |
| Введение в непрерывную интеграцию | 477 |
| Установка Jenkins | 478 |
| Пример с Jenkins | 481 |
| Первое задание для сценария на Python | 481 |
| Плагины Jenkins | 487 |
| Пример непрерывной интеграции в контексте сетевых технологий | 489 |
| Jenkins и Python | 496 |

| | |
|---|------------|
| Непрерывная интеграция в контексте администрирования сети | 497 |
| Резюме | 498 |
| Глава 15. TDD для сетей | 499 |
| Обзор разработки через тестирование | 500 |
| Разные виды тестов | 501 |
| Топология как код | 502 |
| Модуль unittest | 507 |
| Еще о тестировании в Python | 510 |
| Примеры с pytest | 511 |
| Написание тестов для сетей | 513 |
| Тестирование доступности | 514 |
| Тестирование задержек сети | 515 |
| Тестирование безопасности | 516 |
| Тестирование транзакций | 517 |
| Тестирование сетевой конфигурации | 517 |
| Тестирование сценариев Ansible | 518 |
| Интеграция pytest с Jenkins | 519 |
| Интеграция с Jenkins | 519 |
| pyATS и Genie | 524 |
| Резюме | 527 |