

# Python для DevOps и NetOps

Создавайте инструменты эксплуатации, которым можно доверить повторяющиеся проверки инфраструктуры.

**120**

академических часов

**40 + 80**

занятия + самоподготовка

**8 + 1**

проектов модулей + итог

## Зачем проходить этот курс

- Объедините inventory, HTTP API и SSH в одном наборе команд.
- Сделаете массовые операции управляемыми: с лимитами, таймаутами и dry-run.
- Научитесь передавать утилиту команде вместе с тестами, контейнером и инструкцией.

## Кому подойдет

DevOps-, NetOps- и SRE-инженерам, системным администраторам и специалистам эксплуатации.

## Что нужно на старте

Нужны базовый Python, командная строка Linux, основы сетей, HTTP и SSH. Желательно знакомство с Docker и Git.

**Онлайн • Живые занятия • Группа до 8 человек**

# Понятный путь к результату

40 ак. ч. с преподавателем + 80 ак. ч. самостоятельной работы. Один академический час - 45 минут. Практика и проекты входят в указанный объём.

| Модуль                       | Занятия | Самост. |
|------------------------------|---------|---------|
| 01. Среда и CLI-инструмент   | 4       | 8       |
| 02. Сетевые факты и модели   | 5       | 10      |
| 03. Конфигурации и изменения | 5       | 10      |
| 04. HTTP API платформ        | 6       | 12      |
| 05. Драйверы и SSH           | 6       | 12      |
| 06. Параллельный опрос       | 6       | 12      |
| 07. Логи, метрики и отчёты   | 4       | 8       |
| 08. Упаковка и доставка      | 4       | 8       |

## Как устроена практика

На занятиях разбираются подход и ключевые решения. После каждого модуля вы завершаете мини-проект, проверяете его по критериям и оформляете результат. Итоговый проект объединяет эти работы; это один развиваемый кейс, а не дополнительная нагрузка сверх 120 часов.

## Инструменты курса

Python 3 • Typer • Pydantic • ipaddress • Jinja2 • httpx • Paramiko • asyncio • pytest • Docker • CI

## Ведущий преподаватель

Сергей Валерьевич Герасименко - технический директор и основатель IT-компаний «ТЕРАКОР» и «ТЕРАКОР Digital», ведущий преподаватель и практикующий эксперт.

# Изучаете. Применяете. Проверяете.

**МОДУЛЬ 01 / 4 ак. ч. занятий + 8 ак. ч. самостоятельно**

## Среда и CLI-инструмент

- Репозиторий, окружение и зависимости
- Typer, конфигурация и коды завершения
- Pydantic для inventory
- Docker Compose как учебный стенд

### ПРОЕКТ ПОСЛЕ МОДУЛЯ / Inventory CLI

Создать команды проверки и просмотра списка устройств.

**Готово, когда:** Некорректный inventory отклоняется до запуска; help описывает параметры и примеры.

**МОДУЛЬ 02 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно**

## Сетевые факты и модели

- ipaddress: адреса, подсети и пересечения
- Модели интерфейсов и маршрутов
- Парсинг текстового вывода и нормализация
- Проверка неполных и неожиданных данных

### ПРОЕКТ ПОСЛЕ МОДУЛЯ / Аудитор адресного плана

Нормализовать сведения об интерфейсах и найти пересекающиеся подсети.

**Готово, когда:** Ошибочные адреса отмечены; пересечения показаны; результат выгружается в JSON.

# Изучаете. Применяете. Проверяете.

**МОДУЛЬ 03 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно**

## Конфигурации и изменения

- YAML, JSON и TOML
- Jinja2 и обязательные переменные
- pathlib, версии файлов и diff
- Dry-run и отделение генерации от применения

### ПРОЕКТ ПОСЛЕ МОДУЛЯ / Генератор конфигураций

Сформировать конфиги по inventory и показать изменения относительно текущей версии.

**Готово, когда:** Пропущенная переменная вызывает ошибку; dry-run ничего не меняет; diff сохранён.

**МОДУЛЬ 04 / 6 ак. ч. занятий + 12 ак. ч. самостоятельно**

## HTTP API платформ

- httpx, пагинация, таймауты и ограниченные повторы
- Аутентификация и переменные окружения
- Идемпотентность операций и обработка 429/5xx
- Тестовые ответы и подмена внешнего API

### ПРОЕКТ ПОСЛЕ МОДУЛЯ / Клиент инфраструктурного API

Собрать ресурсы по страницам и подготовить план изменения тегов.

**Готово, когда:** Все страницы обработаны; ошибка API понятна; небезопасная операция не повторяется вслепую.

# Изучаете. Применяете. Проверяете.

**МОДУЛЬ 05 / 6 ак. ч. занятий + 12 ак. ч. самостоятельно**

## Драйверы и SSH

- Protocol и контракт устройства
- Factory и отдельные драйверы транспорта
- Paramiko, проверка host keys и таймауты
- Тестирование без доступа к реальному парку

### **ПРОЕКТ ПОСЛЕ МОДУЛЯ / Коллектор фактов и бэкапов**

Реализовать драйвер учебного Linux-узла и тестовый драйвер.

**Готово, когда:** Неизвестный host key требует явного доверия; секреты скрыты; ошибки транспорта отделены.

**МОДУЛЬ 06 / 6 ак. ч. занятий + 12 ак. ч. самостоятельно**

## Параллельный опрос

- asyncio, TaskGroup и Semaphore
- ThreadPoolExecutor для блокирующих вызовов
- Отмена, лимиты, таймауты и backpressure
- Частичные результаты и измерение времени

### **ПРОЕКТ ПОСЛЕ МОДУЛЯ / Массовый health-check**

Опросить набор HTTP и SSH-узлов с ограничением числа одновременных операций.

**Готово, когда:** Недоступный узел не срывает весь опрос; лимит соблюден; итог содержит ошибки и задержки.

# Изучаете. Применяете. Проверяете.

**МОДУЛЬ 07 / 4 ак. ч. занятий + 8 ак. ч. самостоятельно**

## Логи, метрики и отчёты

- Структурированные логи и идентификатор запуска
- Latency, доступность и группировка отказов
- Маскирование чувствительных полей
- Markdown/HTML-отчёт для дежурной смены

### ПРОЕКТ ПОСЛЕ МОДУЛЯ / Отчёт состояния инфраструктуры

Сформировать сводку проблемных узлов с причинами и временем ответа.

**Готово, когда:** У отчёта есть время запуска и охват; частичный сбой явно виден; секретов в выводе нет.

**МОДУЛЬ 08 / 4 ак. ч. занятий + 8 ак. ч. самостоятельно**

## Упаковка и доставка

- CLI entry point и версии пакета
- Dockerfile и конфигурация вне образа
- CI: lint, test, build
- Документация, smoke test и защита

### ПРОЕКТ ПОСЛЕ МОДУЛЯ / NetOps Automation Kit

Объединить команды, тесты и отчёт в переносимый инструмент.

**Готово, когда:** Запуск из чистого окружения воспроизводим; CI выполняет тесты; инструкция содержит dry-run.

# Ваш главный кейс в портфолио

## NetOps Automation Kit

Набор CLI-команд загружает inventory, собирает факты с учебных узлов, сравнивает конфигурации и формирует отчёт о состоянии инфраструктуры. Изменения сначала показываются в dry-run.

### Критерии готовности

- Inventory валидируется до начала сетевых операций.
- Опрос ограничивает параллелизм и не зависает на недоступном узле.
- Отчёт сохраняет частичные результаты и причины ошибок.
- Секреты не записаны в репозиторий и скрыты в логах.
- Есть тесты, Docker-образ и инструкция запуска в CI и локально.

### Что останется после обучения

Репозиторий с исходным кодом, учебные данные или конфигурация стенда, автоматические проверки, инструкция запуска и описание решений. Это материал для демонстрации навыков на работе и техническом собеседовании.

### Как представить результат

Покажите задачу, запустите основной сценарий, продемонстрируйте обработку ошибки и объясните один архитектурный выбор. Отдельно зафиксируйте, что можно улучшить следующим шагом.

Итоговый кейс собирается из проектов модулей. Его завершение и подготовка защиты включены в часы последнего модуля.

## Что вы сможете делать

- Валидировать inventory и нормализовать сетевые факты.
- Генерировать конфигурации и показывать различия до применения.
- Опросить несколько узлов с контролем параллелизма и частичных отказов.
- Выпускать воспроизводимый CLI-инструмент через CI.

## Практическое обучение в ТЕРАКОР

Живой онлайн-формат позволяет задавать вопросы и разбирать решения. Программа сочетает занятия с преподавателем и самостоятельную практику. После успешного прохождения - сертификат ТЕРАКОР и оформленный проект.

### Перед записью

Обсудите исходный уровень, рабочую задачу и время для практики. Для уверенного результата важно проходить модули последовательно и завершать проекты. Курс развивает конкретные навыки; профессиональный уровень зависит также от предыдущего опыта и объёма самостоятельной работы.

### Начните с консультации по программе

Подберём стартовый уровень и уточним формат обучения.

Открыть страницу курса → | [info@terakor.ru](mailto:info@terakor.ru)

+7 (960) 352-38-05 • [terakor.ru](https://terakor.ru)

Сроки старта, стоимость и состав выбранного тарифа уточняются при записи.

Программа оптимизирована на основе страницы курса ТЕРАКОР, изученной 29.09.2026. Это предлагаемая редакция для обновления курса и его описания.

Источник: [terakor.ru/catalog/python/python-devops/](https://terakor.ru/catalog/python/python-devops/)