МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной первым проректором ФГАОУ ВО
РУТ (МИИТ) Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Автоматизация CI/CD и управление инфраструктурой в Linux

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Цифровая инженерия транспортных процессов

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 937226
Подписал: руководитель образовательной программы Проневич Ольга Борисовна
Дата: 01.09.2026

1. Общие сведения о дисциплине (модуле).

Целью освоения учебной дисциплины является формирование у обучающихся системных знаний и практических навыков автоматизации процессов разработки, тестирования и развертывания программных решений, а также управления Linux-инфраструктурой для обеспечения надежного функционирования систем обработки данных в транспортной отрасли.

Дисциплина предназначена для формирования и закрепления у обучающихся компетенций, необходимых для решения следующих профессиональных задач:

- Автоматизация сборки, тестирования и развертывания Python-приложений с использованием CI/CD-инструментов.
- Администрирование и управление Linux-серверами с использованием скриптов и конфигурационных инструментов.
- Контейнеризация приложений с использованием Docker и оркестрация контейнеров.
- Настройка систем мониторинга и логирования для инфраструктурных и прикладных решений.
- Обеспечение безопасного хранения и управления секретами и учетными данными.
- Автоматизация миграции данных и управление конфигурациями в распределенных системах.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ППК-Р3 - Способен выполнять интеграцию программных модулей и компонентов и проверять работоспособность выпусков программного продукта.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- Методы и средства сборки модулей и компонентов компьютерного программного обеспечения: системы управления зависимостями (pip, poetry), инструменты сборки (setuptools, wheel), версионирование артефактов, управление окружениями (venv, conda), конфигурационные файлы сборки (pyproject.toml, setup.py).

- Интерфейсы взаимодействия с внешней средой: сетевые протоколы (HTTP/HTTPS, SSH, gRPC), файловые системы и стандартные потоки ввода/вывода, переменные окружения, конфигурационные файлы, API облачных провайдеров, механизмы межпроцессного взаимодействия и управления процессами в Linux.

Уметь:

- Применять методы и средства сборки модулей и компонентов для разработки процедур развертывания программного обеспечения: создавать Docker-образы, настраивать CI/CD-пайплайны (GitHub Actions, GitLab CI), автоматизировать деплой на различные среды (staging, production) с использованием скриптов и конфигурационных инструментов.

- Разрабатывать процедуры миграции и преобразования данных, включая написание скриптов для ETL-процессов, настройку репликации и синхронизации данных между системами, автоматизацию резервного копирования и восстановления, а также создавать программные интерфейсы для взаимодействия с внешними системами.

Владеть:

- Навыками автоматизации сборки, тестирования и развертывания программных продуктов с использованием CI/CD-инструментов, включая настройку триггеров, управление артефактами, организацию многоступенчатых пайплайнов и обеспечение безопасного хранения секретов.

- Навыками управления Linux-инфраструктурой с использованием Bash-скриптов, систем управления конфигурациями (Ansible), мониторинга и логирования, а также контейнеризации (Docker) для обеспечения воспроизводимости и масштабируемости сред выполнения.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 4 з.е. (144 академических часа(ов).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов

	Всего	Семестр №3
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 80 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Тема 1. Введение в DevOps и CI/CD. Рассматриваемые вопросы: - Роль DevOps и CI/CD в жизненном цикле разработки ПО. - Эволюция подходов к развертыванию: от ручного деплоя к автоматизации. - Принципы непрерывной интеграции, непрерывной доставки и непрерывного развертывания. - Обзор CI/CD-инструментов: GitHub Actions, GitLab CI, Jenkins, TeamCity. - Введение в Infrastructure as Code (IaC).
2	Тема 2. Основы Linux для инженера данных. Рассматриваемые вопросы: - Архитектура Linux: ядро, системные вызовы, файловая система. - Основные команды для работы с файлами, процессами, пользователями и правами доступа. - Управление пакетами (apt, yum, pip). - Работа с переменными окружения и конфигурационными файлами. - Логирование в Linux: syslog, journalctl.
3	Тема 3. Bash-скриптинг для автоматизации. Рассматриваемые вопросы: - Синтаксис Bash: переменные, циклы, условные операторы, функции. - Работа с файлами, строками и процессами из скриптов. - Обработка сигналов и управление ошибками. - Автоматизация рутинных задач с использованием cron и systemd. - Примеры скриптов для backup, мониторинга и деплоя.

№ п/п	Тематика лекционных занятий / краткое содержание
4	Тема 4. Системы управления версиями: Git для командной работы. Рассматриваемые вопросы: - Модели ветвления: Git Flow, GitHub Flow, Trunk-based Development. - Работа с удаленными репозиториями и коллаборация (Pull Requests, Code Review). - Управление тегами и релизами. - Git hooks для автоматизации проверок перед коммитом. - Разрешение конфликтов и работа с историей коммитов.
5	Тема 5. Контейнеризация с Docker. Рассматриваемые вопросы: - Концепция контейнеризации: изоляция, легковесность, воспроизводимость. - Архитектура Docker: клиент-сервер, образы, контейнеры, Docker Registry. - Создание Dockerfile: инструкции FROM, RUN, COPY, ADD, WORKDIR, EXPOSE, CMD, ENTRYPOINT. - Управление образами: сборка, тегирование, публикация. - Работа с томами и сетями в Docker.
6	Тема 6. Docker Compose для многоконтейнерных приложений. Рассматриваемые вопросы: - Структура docker-compose.yml: сервисы, сети, тома. - Оркестрация нескольких сервисов (Python-приложение, PostgreSQL, Redis). - Управление переменными окружения в compose. - Профили и зависимости между сервисами. - Запуск и остановка композиции.
7	Тема 7. Основы GitHub Actions. Рассматриваемые вопросы: - Архитектура GitHub Actions: события, workflow, job, step, action. - Структура YAML-файла для workflow. - Встроенные и пользовательские действия (actions). - Триггеры: push, pull_request, schedule, workflow_dispatch. - Управление секретами и переменными окружения.
8	Тема 8. Сборка и тестирование Python-проектов в CI/CD. Рассматриваемые вопросы: - Настройка окружения для Python: установка зависимостей, кэширование. - Запуск тестов (pytest) и линтеров (ruff, flake8) в CI. - Проверка покрытия кода (coverage.py) и публикация отчетов. - Проверка типов (mypy) и форматирование (black). - Сборка артефактов и их сохранение.
9	Тема 9. Сборка Docker-образов в CI/CD. Рассматриваемые вопросы: - Сборка Docker-образов в CI-пайплайне. - Использование Docker layer caching для ускорения сборки. - Публикация образов в Docker Registry (Docker Hub, GitHub Container Registry). - Тегирование образов с использованием версий и SHA-коммитов. - Безопасная работа с Docker в CI (управление секретами).
10	Тема 10. Развертывание (Deployment) в CI/CD. Рассматриваемые вопросы: - Стратегии развертывания: rolling update, blue-green, canary, A/B-тестирование. - Развертывание на удаленные серверы через SSH. - Использование Docker для деплоя на сервер. - Автоматический откат (rollback) при неудачном деплое. - Практика: деплой на staging и production.

№ п/п	Тематика лекционных занятий / краткое содержание
11	Тема 11. Управление конфигурациями с Ansible. Рассматриваемые вопросы: - Введение в Ansible: архитектура, inventory, модули, плейбуки. - Написание плейбуков для настройки серверов. - Управление пакетами, пользователями, службами, файлами. - Использование переменных и шаблонов (Jinja2). - Применение Ansible для развертывания приложений.
12	Тема 12. Мониторинг и логирование инфраструктуры. Рассматриваемые вопросы: - Метрики и мониторинг: Prometheus, экспортеры, Grafana. - Основы работы с PromQL. - Логирование: стек ELK (Elasticsearch, Logstash, Kibana) и Loki. - Сбор метрик приложений и системных метрик. - Настройка алертинга (Alertmanager).
13	Тема 13. Безопасность в CI/CD: управление секретами. Рассматриваемые вопросы: - Принципы безопасного хранения секретов. - Использование GitHub Secrets, GitLab CI Variables. - Подходы к управлению секретами: HashiCorp Vault, AWS Secrets Manager. - Защита артефактов и сборок от компрометации. - Подпись контейнеров и проверка целостности.
14	Тема 14. Автоматизация миграции и преобразования данных. Рассматриваемые вопросы: - Инструменты миграции данных: Alembic, Flyway, Liquibase. - Автоматизация ETL-процессов в CI/CD. - Миграция схем баз данных без даунтайма. - Тестирование миграций в CI. - Восстановление данных и резервное копирование.
15	Тема 15. Инструменты оркестрации контейнеров: введение в Kubernetes. Рассматриваемые вопросы: - Архитектура Kubernetes: мастер-ноды, воркер-ноды, Pod, Service, Deployment. - Основные объекты Kubernetes. - Деплой приложений в Kubernetes. - Сравнение Docker Swarm и Kubernetes. - Использование kubectl.
16	Тема 16. Итоговый пайплайн: от коммита до продакшена. Рассматриваемые вопросы: - Интеграция всех этапов в единый CI/CD-пайплайн. - Автоматизация полного жизненного цикла data-проекта. - Организация обратной связи: уведомления, отчеты, дашборды. - Культура DevOps: совместная ответственность разработки и эксплуатации. - Подведение итогов курса.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Тема 1. Знакомство с Linux и Bash. Рассматриваемые вопросы: - Установка Linux в виртуальной машине / WSL. - Основные команды Linux: навигация, работа с файлами, права доступа. - Написание простых Bash-скриптов. - Настройка переменных окружения.
2	Тема 2. Продвинутый Bash-скриптинг. Рассматриваемые вопросы: - Работа с условиями и циклами. - Обработка параметров командной строки. - Автоматизация резервного копирования. - Создание systemd-сервиса для запуска приложения.
3	Тема 3. Работа с Git и Git Flow. Рассматриваемые вопросы: - Создание репозитория, работа с ветками и тегами. - Модели ветвления: практика. - Создание Pull Request и Code Review. - Настройка Git hooks.
4	Тема 4. Введение в Docker. Рассматриваемые вопросы: - Установка Docker. - Создание Dockerfile для Python-приложения. - Сборка образа и запуск контейнера. - Работа с томами и портами.
5	Тема 5. Docker Compose для проекта. Рассматриваемые вопросы: - Создание docker-compose.yml для приложения с БД. - Оркестрация Python-приложения, PostgreSQL и Redis. - Управление окружениями через .env. - Запуск и остановка проекта.
6	Тема 6. Настройка GitHub Actions: основы. Рассматриваемые вопросы: - Создание первого workflow на GitHub Actions. - Настройка триггеров (push, pull_request). - Запуск проверок в CI. - Использование GitHub Secrets.
7	Тема 7. CI для Python-проекта. Рассматриваемые вопросы: - Настройка workflow для Python-проекта. - Установка зависимостей, запуск тестов и линтеров. - Проверка покрытия кода тестами. - Сохранение артефактов сборки.
8	Тема 8. Сборка и публикация Docker-образов в CI. Рассматриваемые вопросы: - Интеграция Docker в GitHub Actions. - Сборка образа в CI. - Публикация в GitHub Container Registry. - Кэширование слоев Docker.
9	Тема 9. Развертывание через SSH в CI. Рассматриваемые вопросы:

№ п/п	Тематика практических занятий/краткое содержание
	- Настройка SSH-доступа к серверу. - Автоматический деплой на сервер через GitHub Actions. - Выполнение команд на удаленном сервере. - Управление секретами (SSH-ключи, пароли).
10	Тема 10. Введение в Ansible. Рассматриваемые вопросы: - Установка Ansible. - Написание inventory и плейбуков. - Настройка веб-сервера через Ansible. - Использование модулей apt, service, copy, template.
11	Тема 11. Мониторинг с Prometheus и Grafana. Рассматриваемые вопросы: - Развертывание Prometheus и Grafana через Docker Compose. - Настройка экспортеров (node_exporter). - Создание дашбордов в Grafana. - Настройка алертов.
12	Тема 12. Логирование с ELK / Loki. Рассматриваемые вопросы: - Настройка сбора логов приложения. - Использование стека ELK или Loki. - Просмотр и фильтрация логов. - Визуализация логов.
13	Тема 13. Управление секретами в CI/CD. Рассматриваемые вопросы: - Работа с GitHub Secrets. - Использование HashiCorp Vault (локально). - Безопасное хранение учетных данных для БД и внешних API. - Подпись и проверка Docker-образов.
14	Тема 14. Миграция данных с Alembic. Рассматриваемые вопросы: - Настройка Alembic для SQLAlchemy. - Создание и применение миграций. - Интеграция миграций в CI/CD. - Тестирование миграций.
15	Тема 15. Введение в Kubernetes. Рассматриваемые вопросы: - Развертывание локального кластера (minikube или kind). - Создание Deployment и Service. - Деплой приложения в Kubernetes. - Управление подами и их масштабирование.
16	Тема 16. Итоговый проект: полноценный CI/CD-пайплайн. Рассматриваемые вопросы: - Сборка полного пайплайна: тесты, линтеры, сборка Docker, деплой. - Настройка автоматического деплоя на сервер. - Защита и демонстрация итогового проекта.

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение официальной документации Docker, GitHub Actions, Ansible, Prometheus
2	Самостоятельное написание Bash-скриптов для автоматизации рутинных задач.
3	Разработка CI/CD-пайплайна для учебного Python-проекта.
4	Настройка мониторинга и логирования для собственного проекта.
5	Сравнение различных CI/CD-инструментов (GitHub Actions vs GitLab CI vs Jenkins).
6	Изучение материалов по Kubernetes и оркестрации контейнеров.
7	Подготовка к защите итогового проекта.
8	Подготовка к промежуточной аттестации.
9	Подготовка к текущему контролю.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Кочедыков, С. С. Практические задания для освоения программы практики для магистрантов: Практикум : учебное пособие / С. С. Кочедыков, Г. А. Потапов. — Москва : РТУ МИРЭА, 2026. — 118 с. — ISBN 978-5-7339-2968-2.	https://e.lanbook.com/book/523325
2	Баланов, А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов. — 2-е изд., стер. — Санкт-Петербург : Лань, 2025. — 312 с. — ISBN 978-5-507-52472-3	https://e.lanbook.com/book/45182

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

MS Office (Excel, Word)

Notepad++

Браузер Chrome

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Python 3.11/3.12 и JupyterLab / Jupyter Notebook
pandas, NumPy, SciPy, statsmodels и Pingouin
Matplotlib, Seaborn и Plotly
PostgreSQL и DBeaver Community Edition
Microsoft Excel и Power BI Desktop / LibreOffice Calc
Yandex DataLens, Metabase или Apache Superset
Git и GitHub / GitLab

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).

Компьютер преподавателя с доступом к сети Интернет
Компьютеры студентов с установленным Python-окружением и табличным процессором
Доступ к учебной базе событийных данных и экспериментальным выборкам
Доступ к PostgreSQL и BI-системе для визуализации результатов
Доступ к облачной виртуальной машине или Docker-контейнерам для воспроизводимого анализа
Проектор и экран для проектора
Маркерная доска

9. Форма промежуточной аттестации:

Экзамен в 3 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

руководитель образовательной программы О.Б. Проневич

Согласовано:

Директор Д.В. Паринов

Руководитель образовательной программы О.Б. Проневич

Председатель учебно-методической комиссии Д.В. Паринов