

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра системного анализа и автоматического управления

**РАЗРАБОТКА ИНТЕРАКТИВНОЙ ЦИФРОВОЙ ПЛАТФОРМЫ ДЛЯ
РАЗВИТИЯ НАВЫКОВ СИСТЕМНОГО АДМИНИСТРИРОВАНИЯ В
LINUX-СИСТЕМАХ С ИСПОЛЬЗОВАНИЕМ МЕТОДОЛОГИИ DEVOPS**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы

направления 09.03.04 — Программная инженерия

факультета компьютерных наук и информационных технологий

Панфилова Вадима Андреевича

Научный руководитель

доцент, к. т. н.

И. Н. Фомин

Заведующий кафедрой

к. ф.-м. н., доцент

И. Е. Тананко

Саратов 2026

ВВЕДЕНИЕ

Растут требования к подготовке специалистов в области системного администрирования вычислительных сетей и программной инженерии, в том числе в условиях формирования суверенного интернета Российской Федерации. Профессиональный стандарт «Системный администратор информационно-коммуникационных систем» и ФГОС по направлению 09.03.04 «Программная инженерия» предполагают формирование практических навыков работы с командной строкой Linux, сервисами операционной системы и инструментами сопровождения инфраструктуры.

Подготовка таких специалистов связана с задачами обеспечения устойчивости и управляемости сетевой инфраструктуры. Для вузов при этом остаются актуальными проблемы организации практики: разрозненные скрипты, ручная проверка отчётов, зависимость от конкретной аудитории и проприетарных тренажёров. Интерактивные цифровые платформы с контейнерной изоляцией, веб-терминалом и автоматической проверкой позволяют приблизить учебный процесс к профессиональной деятельности и объективно фиксировать результаты выполнения заданий.

В качестве объекта исследования рассматривается процесс формирования практических навыков системного администрирования Linux-систем и освоения методологии DevOps у студентов направления подготовки «Программная инженерия». Предметом исследования является интерактивная цифровая платформа как средство практико-ориентированного обучения.

Цель бакалаврской работы — разработка интерактивной цифровой платформы для развития навыков системного администрирования в Linux-системах с использованием методологии DevOps.

Поставленная цель определила **следующие задачи**:

1. изучить понятие, историю развития и характеристики интерактивных цифровых образовательных платформ;
2. провести анализ публикаций и существующих исследований по теме;
3. определить задачи и функции платформ для формирования навыков администрирования Linux и практик DevOps;
4. выполнить сравнительный анализ существующих платформ;

5. сформулировать требования к архитектуре и функционалу разрабатываемой системы;
6. разработать и реализовать MVP интерактивной цифровой платформы BashForces;
7. провести тестирование и оценить возможность использования платформы в учебном процессе.

Методологические основы разработки интерактивных цифровых платформ и виртуальных лабораторий представлены в работах Т. В. Никулиной, К. Ю. Гревцова, О. Е. Антипова, М. А. Белова, М. В. Казарина, А. В. Красова; вопросы микросервисной архитектуры, технологического стека и автоматической проверки лабораторных работ рассмотрены в публикациях С. И. Мухаметгалиева, А. С. Петрова, а также в документации Docker и FastAPI.

Теоретическая значимость работы заключается в систематизации требований к интерактивной платформе для подготовки системных администраторов Linux и в обосновании архитектурного подхода с разделением прикладного и лабораторного контуров.

Практическая значимость бакалаврской работы состоит в создании открытого MVP платформы BashForces. Преподаватель получает единый контур для группы: назначение лабораторий, контроль прогресса и журнал попыток проверок. Студент работает в браузере без ручной настройки окружения на персональном компьютере. Платформа поддерживает сценарии Docker, systemd, сетевых утилит и элементов DevOps (CI/CD, Ansible, мониторинг) через декларативные манифесты лабораторных работ. Решение может использоваться в СГУ и других вузах при преподавании дисциплин по операционным системам и программной инженерии.

Бакалаврская работа состоит из введения, четырёх разделов, заключения, списка использованных источников и десяти приложений. Общий объём работы — 68 страниц, из них 66 страниц — основное содержание, включая 8 рисунков и 11 таблиц, цифровой носитель в качестве приложения, список использованных источников информации — 24 наименования.

В работе применялись методы анализа предметной области и сравнительного обзора, функционального моделирования (IDEF0), проектирования программных систем (микросервисная декомпозиция, ER-модели), прототипирования MVP и экспериментальной проверки (функциональное, интеграционное,

нагрузочное тестирование, экспертная апробация в учебной группе).

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Интерактивные цифровые платформы как инструмент развития практических навыков в области системного администрирования и DevOps» посвящён теоретическим основам темы.

В подразделе 1.1 определено понятие интерактивной цифровой платформы, описана история развития виртуальных лабораторий и роль контейнеризации Docker в образовании. Перечислены ключевые характеристики современных платформ:

- доступ через веб-интерфейс без установки ПО на рабочее место студента;
- изолированные окружения (контейнеры или виртуальные машины);
- интерактивный веб-терминал для работы с командной строкой Linux;
- автоматическая проверка выполненных заданий;
- отображение учебного прогресса;
- поддержка DevOps-практик (IaC, CI/CD, мониторинг).

Подраздел 1.2 содержит обзор публикаций по виртуальным лабораториям и контейнерным учебным средам (в том числе работы М. А. Белова, А. В. Красова и др.). Анализ показывает, что в научной литературе по-прежнему слабо представлены открытые комплексные решения для Linux и DevOps, адаптированные под российские вузы. Подраздел 1.3 систематизирует образовательные задачи и функции платформ: интерактивный терминал, изолированные окружения, автоматическая проверка, учёт прогресса, инструменты преподавателя (группы, мониторинг попыток). В подразделе 1.4 выполнен сравнительный анализ KillerCoda, Rotoro.Cloud, KodeKloud, LabEx и Instruqt; отмечены ограничения коммерческих платформ по открытости, стоимости и глубине сценариев Linux/DevOps.

Подраздел 1.5 формулирует цель, техническую постановку (MVP с изоляцией, терминалом, проверками, учётом прогресса), объект и предмет исследования. Отдельно обоснован выбор Docker как базового механизма изоляции: воспроизводимость окружения, переносимость между узлами и соответствие индустриальной практике DevOps. Для преподавателя зафиксированы требования к мониторингу группы и журналу попыток, для студента — к прозрачности

критериев автоматической проверки и к сохранению прогресса между занятиями. Сформулированы границы MVP: декларативное описание лабораторий в манифестах `lab.yaml`, разделение контуров приложения и лабораторного Docker-контура, поддержка сценариев системного администрирования и базовых DevOps-практик без претензии на полную замену коммерческих тренажёров. **Вывод по разделу:** для российских вузов целесообразна открытая адаптируемая платформа, дополняющая существующие коммерческие решения.

Второй раздел «Проектирование интерактивной цифровой платформы» описывает проектные решения платформы BashForces.

В подразделе 2.1 построена IDEF0-модель процесса разработки (входы, управление, механизмы, выходы). Подраздел 2.2 фиксирует функциональные и нефункциональные требования. К ключевым функциональным требованиям отнесены: аутентификация и роли, каталог курсов и лабораторий, запуск и остановка сессий, веб-терминал, автоматические проверки, фиксация прогресса, инструменты преподавателя. Нефункциональные требования включают безопасную изоляцию, масштабируемость учебной нагрузки, воспроизводимость развёртывания и сопровождаемость конфигурации. Подраздел 2.3 обосновывает микросервисную архитектуру и стек FastAPI, React, PostgreSQL, Redis, Nginx, Docker.

На рисунке 1 представлена общая архитектура платформы.

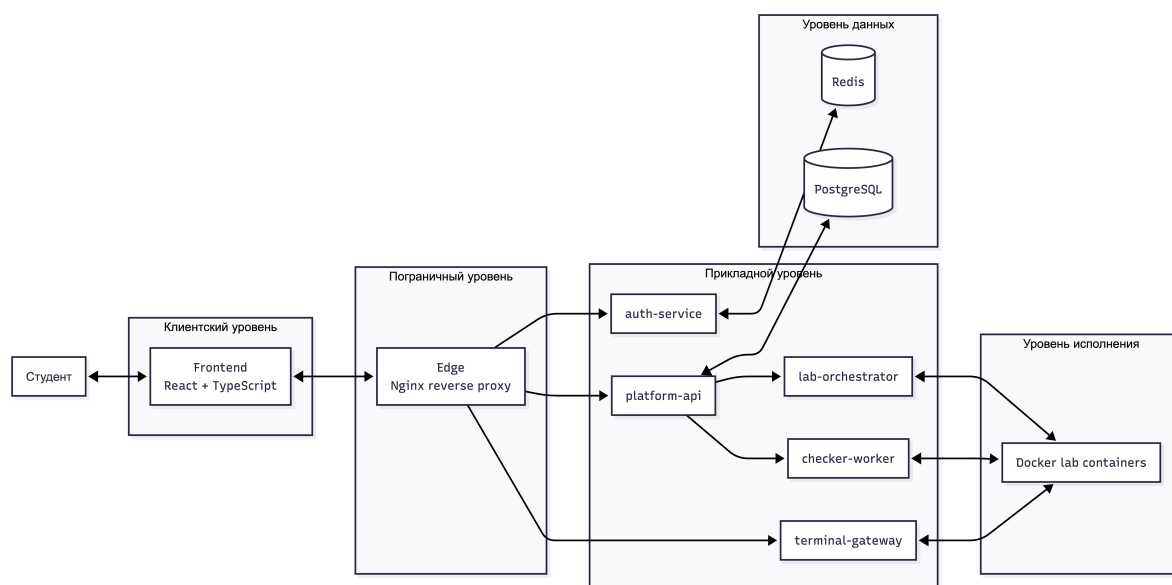


Рисунок 1 – Общая архитектура платформы BashForces

Подраздел 2.4 описывает общую архитектуру и потоки данных. Пользова-

тельский интерфейс взаимодействует с сервисом аутентификации и сервисом `platform-api`; данные о курсах, сессиях и прогрессе хранятся в PostgreSQL, а кратковременные ограничения и вспомогательные структуры — в Redis.

После аутентификации клиент обращается к API платформы; при запуске лаборатории создаётся сессия в базе и контейнер в `lab-orchestrator`; терминал работает по WebSocket через Nginx; проверку шага выполняет `checker-worker` в среде студента, результат фиксируется в базе. Такой порядок обеспечивает согласованность данных на всём пути от входа в систему до фиксации результата проверки.

Подраздел 2.5 раскрывает функциональную структуру платформы по ролям. Для студента определены сценарии входа в каталог, запуска и завершения лабораторной сессии, работы в терминале и просмотра статуса проверок. Для преподавателя — назначение лабораторий группе, мониторинг активных сессий и анализ журнала попыток. Для администратора — управление учётными записями, конфигурацией развёртывания и параметрами изоляции.

Подраздел 2.6 формализует модели данных: сущности пользователя, курса, лаборатории, учебной сессии, шага, попытки проверки и записи прогресса; зафиксированы связи «курс — лаборатория», «сессия — контейнер», «шаг — проверка», что обеспечивает трассируемость результатов обучения. Подраздел 2.7 проектирует изоляцию: `DinD per session`, TTL сессий, очистка ресурсов, схема `splitdeploy` (узлы `prodvuz` и `labvm`) для разделения привилегированного контура Docker на отдельном узле. Отдельно рассмотрены угрозы межсессионного влияния и меры снижения риска (ограничение привилегий, сетевые политики, контроль времени жизни контейнера). **Вывод:** получена согласованная архитектура с явными границами доверия и предсказуемым жизненным циклом сессии.

На рисунке 2 показано взаимодействие компонентов при прохождении лабораторной работы.

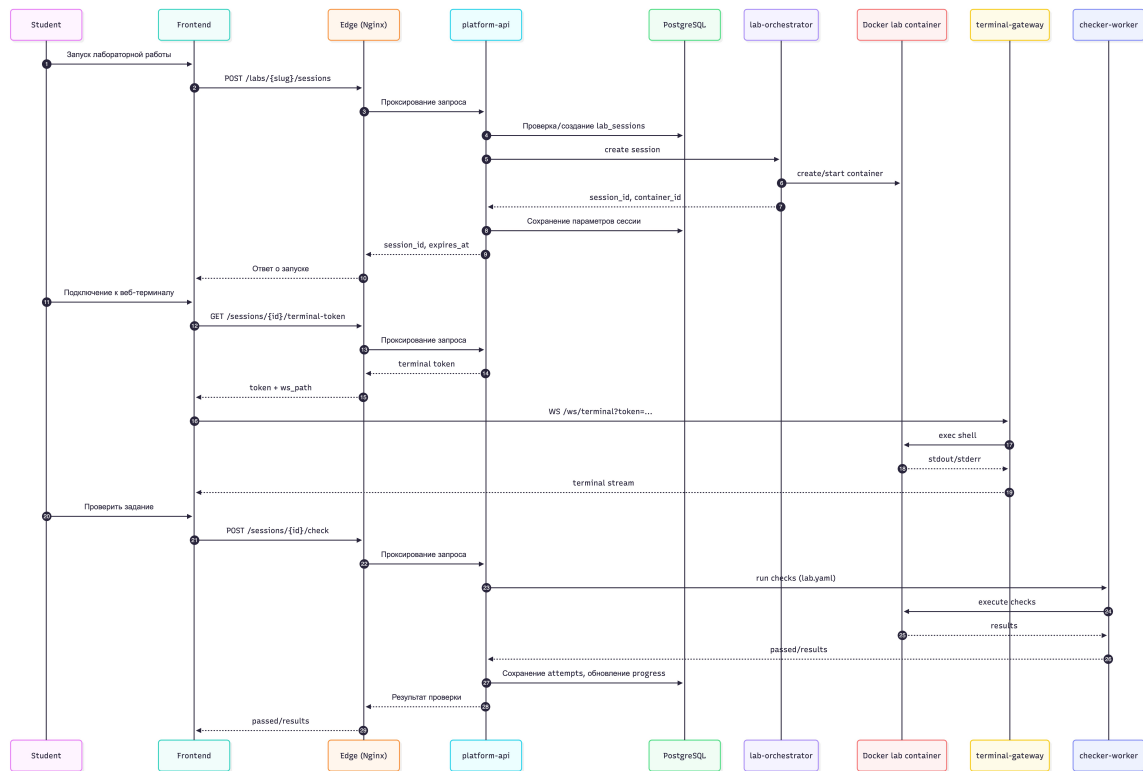


Рисунок 2 – Взаимодействие компонентов платформы в учебном сценарии

Схема взаимодействия отражает типовой сценарий занятия. Студент выбирает лабораторную работу; сервис platform-api создаёт учебную сессию, lab-orchestrator поднимает изолированное окружение, terminal-gateway обеспечивает доступ к командной оболочке.

По завершении шага checker-worker сравнивает фактическое состояние с критериями из манифеста и передаёт результат в API платформы для отображения прогресса в интерфейсе.

Третий раздел «Реализация и тестирование интерактивной цифровой платформы» раскрывает реализацию MVP.

Состав реализованных сервисов MVP:

- auth-service — учётные записи, выдача JWT;
- platform-api — каталог, сессии, прогресс, API для frontend;
- lab-orchestrator — жизненный цикл контейнеров и DinD;
- terminal-gateway — WebSocket-доступ к оболочке;
- checker-worker — выполнение проверок по lab.yaml;

— frontend (React) — учебный интерфейс и терминал.

Формат `lab.yaml` задаёт последовательность шагов лабораторной работы, ожидаемый результат, подсказки и набор проверок для каждого шага. Такой подход позволяет преподавателю расширять каталог занятий без пересборки прикладных сервисов: достаточно добавить манифест и образ лаборатории в репозиторий платформы. Проверки выполняются внутри изолированного контейнера студента, что исключает влияние соседних сессий на результат оценивания.

Подраздел 3.1 описывает backend: структура сервисов, JWT-аутентификация, роли (студент, преподаватель, администратор), защищённые межсервисные вызовы с внутренним токеном при создании лабораторной сессии. Реализованы REST-эндпоинты каталога, управления сессиями, фиксации прогресса и служебные операции очистки устаревших контейнеров.

Подраздел 3.2 — frontend на React и TypeScript: HTTP-клиент с refresh токенов, отображение шагов лаборатории, канал WebSocket для терминала, обработка состояний ошибок сети и истечения сессии.

Подраздел 3.3 — подсистема лабораторных окружений: провайдер Docker, оркестратор сессий, DinD per session, шлюз терминала с проверкой terminal-token, модуль проверок с разбором `lab.yaml` и типами проверок (код выхода команды, содержимое файла, скрипт). Каталог лабораторий поставляется вместе с платформой (в том числе `docker-intro`, `kind-intro`, сценарии по `systemd` и сетевым утилитам). Предусмотрены режимы развёртывания: локальный контур для разработки и отладки, а также эксплуатационная схема `split-deploy` с выделенным узлом лабораторного Docker-контура. Настроен pipeline CI/CD для сборки образов, прогона проверок и доставки конфигурации на учебный стенд.

Подраздел 3.4 фиксирует результаты модульного и интеграционного тестирования реализованных компонентов до перехода к комплексной апробации в разделе 4. **Вывод:** реализован и проверен сквозной учебный сценарий в едином веб-интерфейсе.

Четвёртый раздел «Тестирование и анализ результатов» посвящён верификации и апробации.

В подразделе 4.1 сформулированы цели и задачи тестирования, виды проверок (функциональное, интеграционное, нагрузочное, usability) и критерии успешности на основе трассировки «требование — сценарий — результат».

Для каждого ключевого требования из раздела 2 зафиксированы проверяемые сценарии запуска лаборатории, работы терминала, выполнения проверки шага и отображения прогресса в интерфейсе преподавателя.

Подраздел 4.2 описывает организационно-техническую модель занятия для аудиторного, дистанционного и смешанного форматов: роли студента, преподавателя и администратора; этапы подготовки, выполнения и пост-анализа; регламент 90 минут; инструменты контроля группы и журнал попыток.

В подразделе 4.3 зафиксированы выявленные проблемы (таймауты DinD, гонки при очистке сессий, обработка ошибок внешних зависимостей) и способы устранения. Для таймаутов уточнены параметры ожидания готовности вложенного Docker-демона; для гонок при очистке — идемпотентные операции завершения сессии и повторные попытки освобождения ресурсов.

Описаны результаты пакетных прогонов (серии 5/10/15 сессий) с фиксацией успешности старта и времени запуска, а также нагрузочного испытания оркестратора лабораторий (до 460 одновременных DinD-сессий при заданных параметрах памяти). Сопоставление метрик показало устойчивый старт при типовой нагрузке учебной группы и необходимость планирования ресурсов узла при массовых занятиях.

Подраздел 4.4 оценивает техническую и педагогическую эффективность: подтверждён сквозной контур обучения, сокращено время организационных операций, ускорена обратная связь по проверкам. Сравнение с коммерческими платформами из раздела 1 подтверждает преимущество BashForces по адаптивности под рабочую программу вуза, контролируемости развёртывания и отсутствию лицензионной зависимости. По экспертной апробации группы из 15 студентов 5 курса 93 % участников отметили повышение вовлечённости и скорости освоения материала; преподаватель отметил снижение доли ручной проверки отчётов и возможность оперативно видеть типовые ошибки по журналу попыток. **Вывод:** платформа эксплуатационно пригодна для учебного процесса вуза и может тиражироваться без изменения базовой архитектуры.

В приложениях приведены фрагменты исходного кода ключевых модулей (`docker_provider.py`, `orchestrator.py`), конфигурации `split-deploy`, примеры `lab.yaml`, схемы развёртывания и фрагменты pipeline CI/CD.

Они дополняют основной текст и подтверждают воспроизводимость реализованных решений.

ЗАКЛЮЧЕНИЕ

Выпускная квалификационная работа была посвящена разработке интерактивной цифровой платформы BashForces для формирования практических навыков системного администрирования Linux и освоения DevOps-подхода в учебном процессе.

В ходе выполнения работы достигнута поставленная цель: разработан и проверен на работоспособность MVP интерактивной цифровой платформы с веб-интерфейсом, изолированными лабораторными окружениями, автоматической проверкой заданий и инструментами сопровождения занятий для преподавателя.

Были решены задачи, сформулированные во введении:

1. изучены понятие, история развития и ключевые характеристики интерактивных цифровых образовательных платформ;
2. выполнен анализ публикаций и существующих исследований по теме;
3. определены задачи и функции платформ, ориентированных на развитие навыков администрирования Linux и практик DevOps;
4. проведён сравнительный анализ существующих интерактивных платформ;
5. сформулированы требования к архитектуре и функционалу разрабатываемой системы;
6. разработан и реализован MVP платформы BashForces с микросервисной архитектурой, веб-терминалом, автоматической проверкой и подсистемой изоляции лабораторных окружений;
7. проведено тестирование платформы и подтверждена возможность её использования в учебном процессе.

Особое внимание уделено механизму DinD per session, схеме отдельного развёртывания (splitdeploy, узлы prodvuz и labvm) и декларативному формату lab.yaml, что обеспечило разделение контуров доверия и расширяемость каталога лабораторных работ.

Результаты испытаний и апробации подтвердили практическую применимость платформы для проведения лабораторных занятий по Linux и DevOps-дисциплинам. Материалы работы могут использоваться при подготовке методических указаний и организации практики на кафедре.

Перспективными направлениями развития являются вынесение платформы в SaaS-контур, расширение библиотеки лабораторных работ и интеграция виртуальных машин на сессию для заданий, где контейнерной изоляции не хватает для полного воспроизведения учебного сценария.