

Курс DevOps

Инструкция по сдаче лабораторных работ

1 апреля 2026 г.

1 Цель документа

Настоящий документ описывает общие правила, требования и процесс выполнения и сдачи лабораторных работ по курсу DevOps. Соблюдение этих правил обязательно для участников курса.

2 Требования к обучающемуся для выполнения лабораторных работ

Для успешного выполнения лабораторных работ обучающемуся необходимо иметь теоретические и практические навыки в следующих областях:

- базовые навыки работы в UNIX-подобных операционных системах: понимание структуры файловой системы, процессов и прав доступа;
- навыки работы в командной строке: выполнение команд, работа с файлами и каталогами;
- основы компьютерных сетей: понимание принципов взаимодействия узлов в сети, назначение IP-адресов, принципы работы протокола HTTP;
- базовое понимание архитектуры клиент-серверных приложений.

3 Требования к техническому обеспечению

Для выполнения лабораторных работ обучающемуся необходимо наличие:

- устройство под управлением операционной системы семейства UNIX либо возможность использования виртуальной машины под управлением данной операционной системы;
- стабильное подключение к сети Интернет;
- учетная запись в github.com.

4 Организационные требования курса

Перед началом выполнения лабораторных работ обучающемуся необходимо выполнить следующие организационные действия:

1. выбрать приложение, используемое для выполнения лабораторных работ;
2. выполнить создание копии (fork) исходного репозитория в личное пространство GitHub;
3. создать pull request в исходный репозиторий для автоматических проверок;
4. предоставить права доступа сервисному боту в созданном репозитории.

4.1 Выбор приложения для лабораторной работы

Для выполнения лабораторных работ в данном курсе обучающемуся предлагается выбрать одно из приложений, предоставленных для работы. Ознакомиться с приложениями можно по следующим ссылкам:

- **Статический сайт:**

- **C# приложение:** [danlla/Csharp-example](#)

- **Python веб приложение:** [prafdin/catty-reminders-app](#)

В случае использования собственного приложения, которое не входит в предложенный список, необходимо согласовать его с преподавателем. При этом учитывайте, что:

- автоматические проверки для собственного приложения применяться не будут;
- при сдаче лабораторных работ необходимо продемонстрировать работоспособность приложения преподавателю.

4.2 Создание копии исходного репозитория

Перед началом выполнения лабораторных работ необходимо создать копию (**fork**) исходного репозитория [prafdin/devops-course-labs](#).


При создании копии назовите репозиторий в соответствии с выбранным приложением. При создании копии обратите внимание на чекбокс **Copy the main branch only** - он должен быть отключен.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner *

 prafdin

Repository name *

Csharp-example

✔ Csharp-example is available.


By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description

0 / 350 characters

Copy the `main` branch only

Contribute back to SA3000UDP/for-r by adding your own branch. [Learn more](#).

 You are creating a fork in your personal account.

Create fork

4.3 Регистрация в репозитории для автоматических проверок

Для сдачи лабораторных работ с применением автоматических проверок необходимо создать pull request в исходный репозиторий [prafdin/devops-course-labs](#).

В pull request необходимо добавить новый блок данных в файл params.json в список users. В новый блок необходимо внести следующую информацию:

- login: логин вашего аккаунта в GitHub
- id: транслитерация вашей фамилии. Например, ivanov.
- app: название выбранного приложения. Например, catty-reminders-app

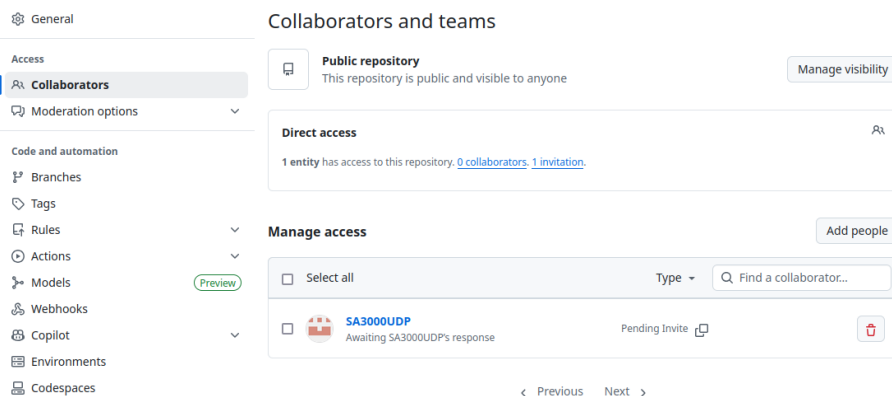
Название pull request должно удовлетворять шаблону 63NN Регистрация Иванов И.И.

После создания pull request убедитесь что все проверки пройдены. Если проверки не пройдены - внесите изменения и переотстройте pull request чтобы проверка запустилась заново.

4.4 Предоставление доступа боту

Для работы автоматических проверок необходимо предоставить доступ боту в созданный вами репозиторий.

Для этого перейдите в настройки репозитория, Access, Collaborators и отправьте приглашение аккаунту [SA3000UDP](#). После подтверждения приглашения бот будет иметь полный доступ к вашему репозиторию.



После завершения курса вы можете отозвать доступ.

5 Выполнение лабораторной работы

Для выполнения лабораторной работы необходимо соблюдать следующую последовательность действий:

1. При выполнении первой лабораторной работы создайте новую ветку lab1 от существующей ветки, соответствующей названию выбранного приложения.

2. Для каждой последующей лабораторной работы создавайте новую ветку от ветки предыдущей лабораторной работы. Например, ветка lab2 создается от ветки lab1 и т.д.
3. Все файлы, необходимые для выполнения лабораторной работы, добавляйте в соответствующую рабочую ветку.

6 Процесс сдачи

Для сдачи лабораторной работы Вам необходимо создать pull request в репозиторий [prafdin/devops-course-labs](#) в ветку по названию выбранного приложения.

Перед созданием pull request убедитесь, что требуемый сценарий лабораторной работы может быть выполнен: виртуальная машина включена, все необходимые приложения запущены.

Название pull request должно удовлетворять шаблону БЗNN Лаб.1 Иванов И.И.

После создания pull request в вашем репозитории автоматически запустится набор [проверок](#), которые необходимо пройти.

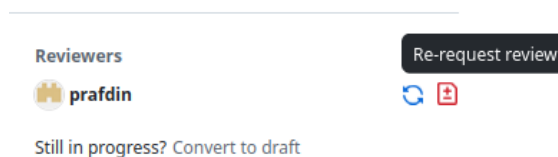
В случае необходимости внесения изменений для успешного прохождения проверок:

- внесите изменения в рабочую ветку;
- оставьте комментарий /rerun-all в pull request для повторного запуска проверок.

Если проверки упали по независящим от вас причинам, допускается их перезапуск без внесения изменений. В случае, если вы считаете, что проверки упали неправомерно и все требования лабораторной работы выполнены, необходимо обратиться к преподавателю.

После успешного завершения всех проверок ожидайте проверку от преподавателя.

После внесения исправлений по замечаниям преподавателя необходимо запросить повторное ревью в pull request:

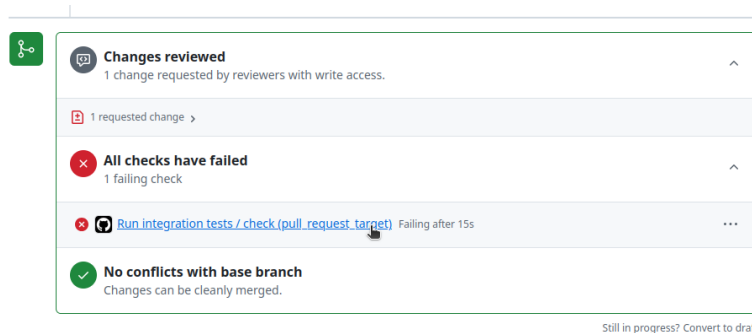


В рамках одной лабораторной работы допускается не более трёх запросов на повторное ревью. При превышении указанного лимита при сдаче добавляются дополнительные вопросы в количестве, равном превышению.

После получения одобрения (approve) от преподавателя обучающийся допускается к посещению сдачи лабораторной работы. На сдаче необходимо ответить на два вопроса по выполненной лабораторной работе, а также прокомментировать выполненную работу и пояснить принятые решения.

6.1 Просмотр результатов проверки

Для детального просмотра результатов проверки перейдите в соответствующий pull request и нажмите на название проверки.



В открывшейся вкладке перейдите к логам выполнения основного Python скрипта и самостоятельно проанализируйте их.

6.2 Перезапуск автоматической проверки

Если при создании pull request вы не учли все требования лабораторной работы или системы сдачи, проверку можно запустить заново, добавив комментарий `/rerun-all` в pull request. При условии, что вы заранее зарегистрировались на курс, система автоматически перезапустит проверки.

В ряде случаев, перечисленных ниже, команда `/rerun-all` может не сработать из-за особенностей работы GitHub Actions. Тогда необходимо переоткрыть pull request: сначала закройте его, затем откройте снова. Это действие инициирует повторную проверку с актуальными данными.

Ситуации, когда требуется переоткрыть pull request:

1. переименование pull request;
2. изменение кода GitHub Actions в ветке main основного репозитория.

Список не является исчерпывающим. Если вы столкнулись с незадокументированным сценарием, при котором потребовалось переоткрывать pull request, сообщите об этом в рамках программы Bug Bounty.

```
720 ##[debug]Loading env
721 ▶ Run python3 -m scripts.check_webhooks_devops_assignment \
740 ##[debug]/usr/bin/bash --noprofile --norc -e -o pipefail /home/runner/work/_temp/ccb42d7a-49d6-4b53-a82f-144fd07c5a13.sh
741 Checking assignment for repository: git@github.com:sa3000udp/check-assignment-tests.git
742 [check_app_is_alive] Start test
743 --- Running Test: GET request to http://app_sa3000udp_course.prafdin.ru ---
744 Test FAILED: Received status code 404.
745 [check_app_is_alive] Test failed
746 [check_event_update_site] Start test
747 [check_event_update_site] Test failed with exception:
748 Meta tag with 'deployref' name not found on page
749
750 2 out of 2 checks FAILED.
751 Error: Process completed with exit code 1.
752 ##[debug]Finished: run
```

Для анализа ошибок обратитесь к разделу [Требования автоматической проверки](#), а также к исходному коду проверок в репозитории [prafdin/check-assignment](#).

Если после изучения логов и исходного кода анализ ошибки затруднен, обратитесь к преподавателю.

7 Требования автоматической проверки

7.1 Требования к работе GitHub Webhooks

Код автоматической проверки: [scripts/check_webhooks_devops_assignment.py](#)

7.1.1 Приложение запущено и доступно через FRP

Условие проверки: приложение должно быть запущено и доступно по адресу:¹ [app.ID.course.prafdin.ru](#).

Исходный код проверки: [check_app_is_alive](#).

7.1.2 Приложение обновляется при наступлении события в репозитории

Условие проверки: значение *deployref*, извлечённое с главной страницы до наступления события, не должно совпадать со значением, полученным после наступления события.

Исходный код проверки: [check_event_update_site](#).

7.1.3 Значение *deployref* соответствует ожидаемому

Условие проверки: значение *deployref*, извлечённое с главной страницы после наступления события, соответствует хешу коммита, вызвавшего данное событие.

Исходный код проверки: [check_deploy_ref_matches_commit](#).

7.2 Требования к работе GitHub Actions

Код автоматической проверки: [scripts/check_github_actions_assignment.py](#).

Список проверок, повторно используемых из прошлого задания:

- [Приложение запущено и доступно через FRP](#)
- [Значение *deployref* соответствует ожидаемому](#)

7.2.1 Событие *push* запускает *workflow*

Условие проверки: для коммита, созданного автоматической проверкой, должен запуститься *workflow* и он должен завершиться успехом.

Исходный код проверки: [push_and_check_workflow](#).

¹Здесь и далее по тексту вместо ID необходимо использовать ID, которое было указано при регистрации на курс

7.2.2 В workflow запускаются тесты

Условие проверки: для коммита, созданного автоматической проверкой, должны запускаться юнит-тесты, результаты которых прикладываются в workflow артефакт с названием *test_result*. Тесты должны завершиться без ошибок.

Исходный код проверки: [check_tests_passed](#).

7.2.3 В репозитории есть требуемые workflow файлы

Условие проверки: в репозитории должны находиться оба файла: *./github/workflows/ci.yaml* и *./github/workflows/deploy.yaml*.

Исходный код проверки: [check_required_workflow_files](#).

7.2.4 Release запускает CD workflow

Условие проверки: релиз, созданный автоматической проверкой, должен запустить CD workflow и обновить сайт: значение *deployref*, извлечённое с главной страницы до наступления события, не должно совпадать со значением, полученным после наступления события.

Исходный код проверки: [check_release_updates_site](#).

7.3 Требования к работе Docker

Код автоматической проверки: [scripts/check_docker_assignment.py](#).

Список проверок, повторно используемых из прошлого задания:

- Приложение запущено и доступно через FRP
- Значение *deployref* соответствует ожидаемому
- Событие *push* запускает workflow
- В workflow запускаются тесты
- В репозитории есть требуемые workflow файлы
- Release запускает CD workflow

7.3.1 CI workflow собирает Image

Условие проверки: после завершения CI workflow в репозитории создается [Package](#) с тегом имеджа *"ghcr.io/image_name:tag"*, где *image_name* строится по правилу *логин_пользователя/название_приложения*, а *tag* - это hash коммита, созданного автоматической проверкой.

Исходный код проверки: [check_docker_image_exists](#).

7.4 Требования к работе Docker Compose

Код автоматической проверки: [scripts/check_compose_assignment.py](#).
Список проверок, повторно используемых из прошлого задания:

- Приложение запущено и доступно через FRP
- Значение `deployref` соответствует ожидаемому
- Событие `push` запускает `workflow`
- В `workflow` запускаются тесты
- В репозитории есть требуемые `workflow` файлы
- `Release` запускает `CD workflow`

8 Bug Bounty

В рамках курса действует программа *Bug Bounty*, направленная на выявление недочётов и ошибок в автоматических проверках лабораторных работ.

Если обучающийся обнаружил некорректную работу проверок (ошибки логики, ложные срабатывания, некорректные условия проверки и т.п.), он может заявить о найденной проблеме и получить дополнительные преференции.

8.1 Что считается недочётом проверки

К недочётам автоматических проверок относятся:

- ложные отрицательные срабатывания (проверка не проходит при корректно выполненной лабораторной работе);
- ложные положительные срабатывания (проверка проходит при некорректной реализации);
- некорректные или противоречивые требования проверки;
- ошибки в логике сценариев проверки;

8.2 Неточности в руководстве и методических материалах

В рамках программы также приветствуются вопросы, связанные с лабораторной работой. Если заданный вопрос помогает выявить неточность, неоднозначность формулировок или запутанность руководства либо методических материалов, такая ситуация может быть признана проблемой документации.

8.3 Порядок сообщения о найденной проблеме

Для того чтобы сообщить о найденной проблеме или задать вопрос, создайте Issue в исходном репозитории [prafdin/devops-course-labs](#) с описанием ошибки.

8.4 Преференции

В случае подтверждения обнаруженной ошибки обучающемуся могут быть начислены бонусные баллы в диапазоне от 0.5 до 1.5 в зависимости от критичности ошибки. За вопрос, позволивший выявить неточность документации, могут быть начислены бонусные баллы до 0.5.

Накопленные бонусные баллы можно использовать для закрытия лабораторных работ. Любая лабораторная работа может быть закрыта при списании 3 баллов.