



> Конспект > 2 урок > Практика: создаём простой DAG

> Оглавление

- > [Оглавление](#)
- > [Клонирование репозитория](#)
- > [Создание DAG-а](#)
- > [Отправка изменений в репозиторий](#)
- > [DAG на веб-сервере Airflow](#)

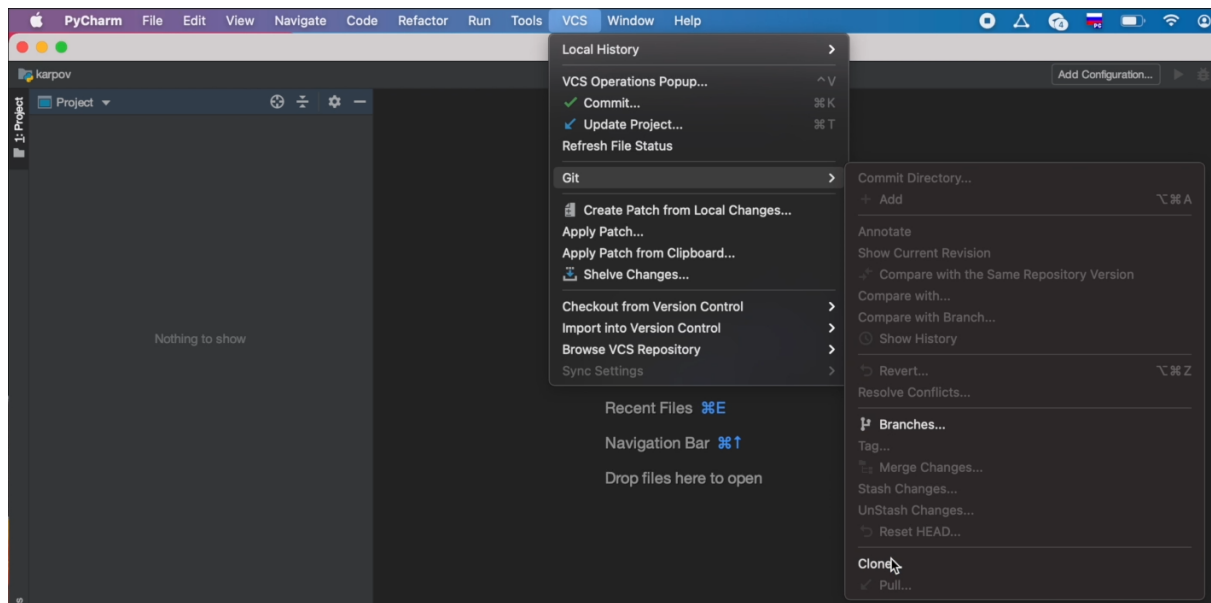
> Клонирование репозитория

Зайдём в наш репозиторий и скопируем ссылку для клонирования (Clone with HTTPS).

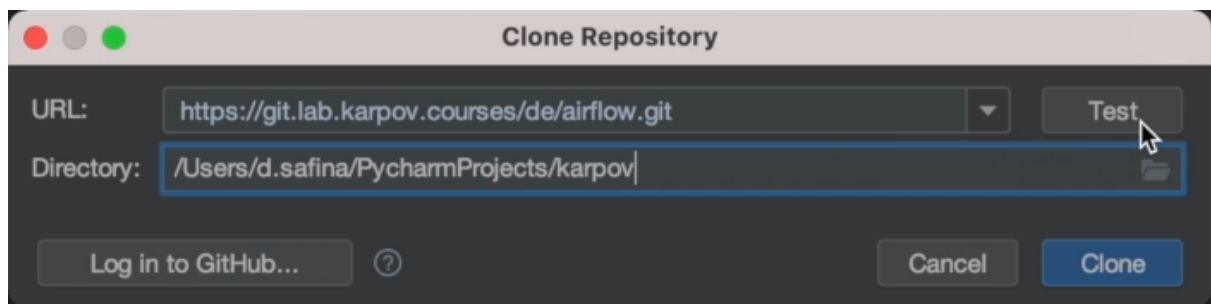
The screenshot shows the GitLab web interface for a repository named 'airflow'. The sidebar on the left contains navigation links: Project information, Repository, Issues (0), Merge requests (12), CI/CD, Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, Analytics, Wiki, and Snippets. The main content area displays the repository details, including the project ID (132), star and fork counts (0 each), and commit statistics (1,631 Commits, 37 Branches, 0 Tags, 2.4 MB Files, 4.4 MB Storage). A commit history table is visible, showing the latest commit 'Delete ds_test.py' by Дина Сафина. A 'Clone with HTTPS' button is highlighted, showing the URL 'https://git.lab.karpov.courses/'.

Name	Last commit
days	Delete ds_test.py
plugins	Homework 3 was started.
.airflowignore	+ .airflowignore 1 month ago
.gitignore	feature: add .gitignore 3 weeks ago

В IDE PyCharm клонируем репозиторий: **VCS -> Git -> Clone**.



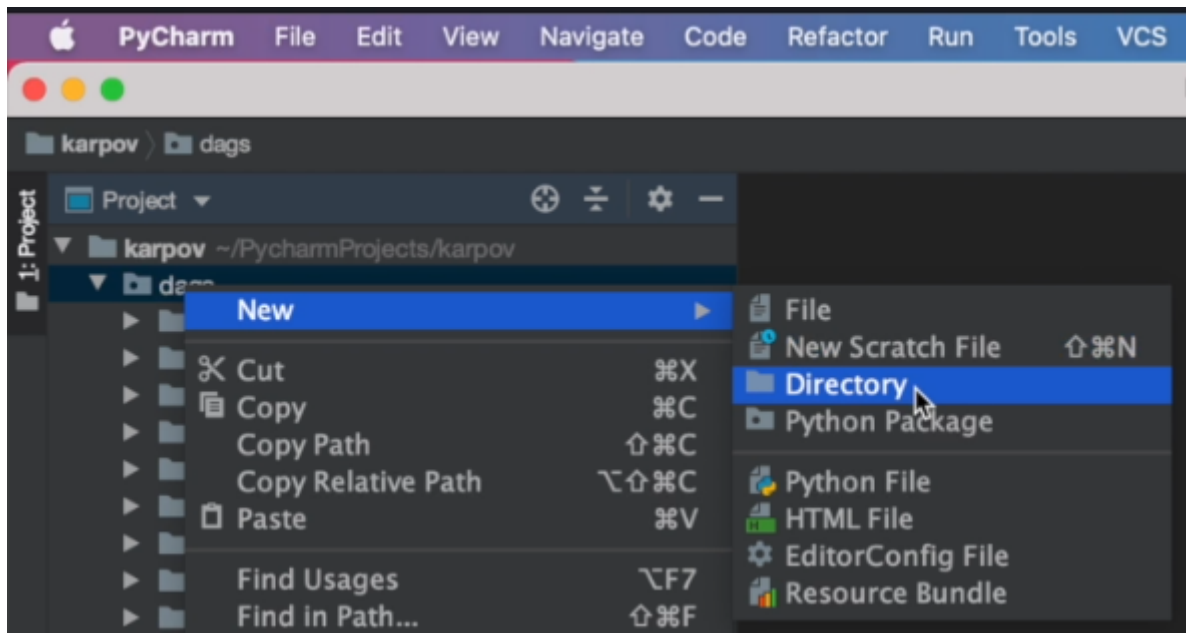
Вставляем ссылку, скопированную ранее, а также указываем путь, где будет храниться репозиторий.



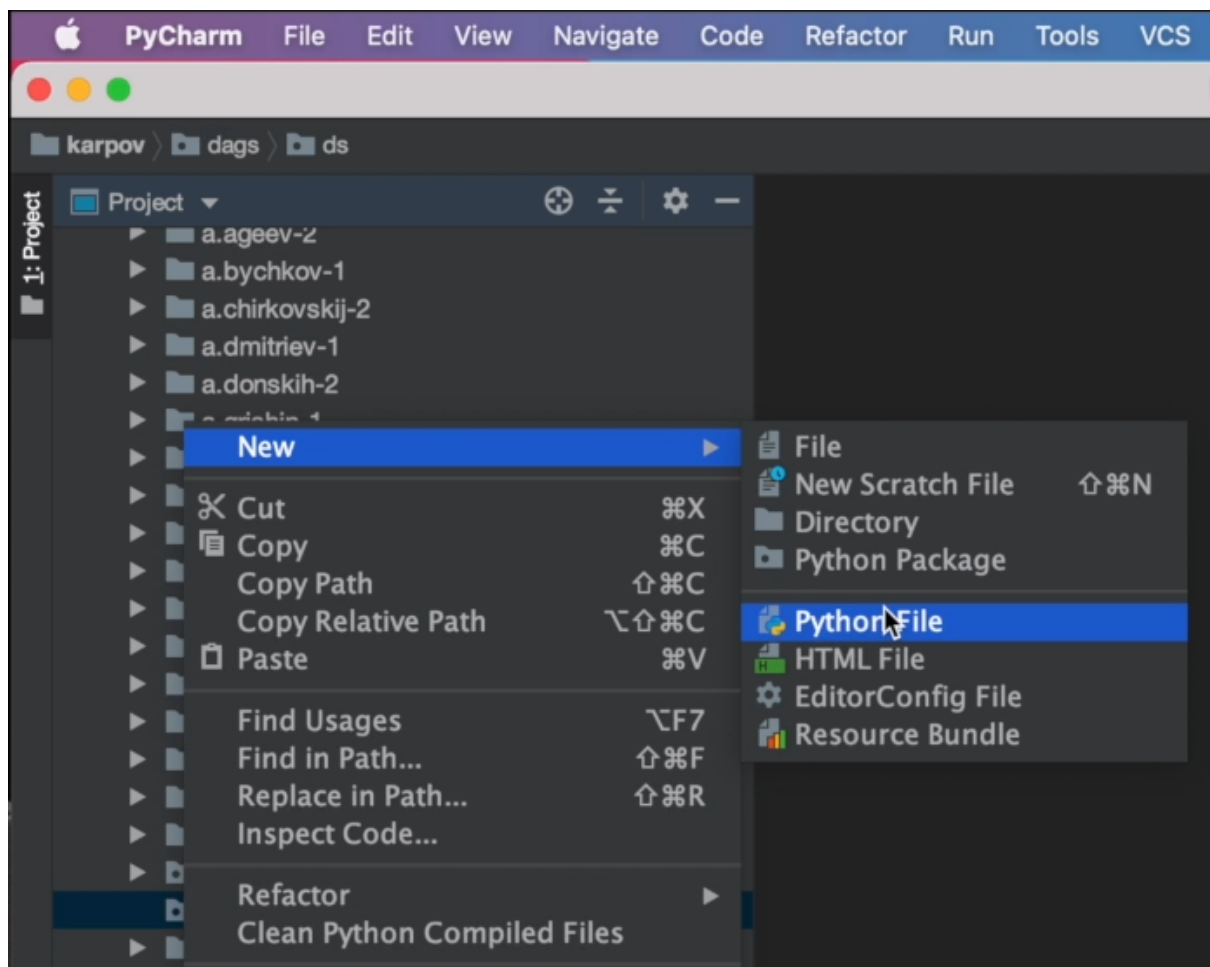
Для клонирования будет запрошен пароль, если он ещё не сохранён.

> Создание DAG-а

В PyCharm видим скопированный репозиторий, создадим папку (в нашем случае с названием **ds**), в которой далее создадим DAG.



В этой папке создаём файл python (в нашем случае `ds_test.py`).



Содержание `ds_test.py`, который будет выводить дату запуска DAG Instance и сообщение «Hello World!»:

```
"""
Тестовый даг
"""
from airflow import DAG
from airflow.utils.dates import days_ago
import logging

from airflow.operators.dummy_operator import DummyOperator
from airflow.operators.bash import BashOperator
from airflow.operators.python_operator import PythonOperator

DEFAULT_ARGS = {
    'start_date': days_ago(2),
    'owner': 'ds',
    'poke_interval': 600
}

with DAG("ds_test",
        schedule_interval='@daily',
        default_args=DEFAULT_ARGS,
        max_active_runs=1,
        tags=['ds']) as dag:

    dummy = DummyOperator(task_id="dummy")

    echo_ds = BashOperator(
        task_id='echo_ds',
        bash_command='echo {{ ds }}',
        dag=dag
    )

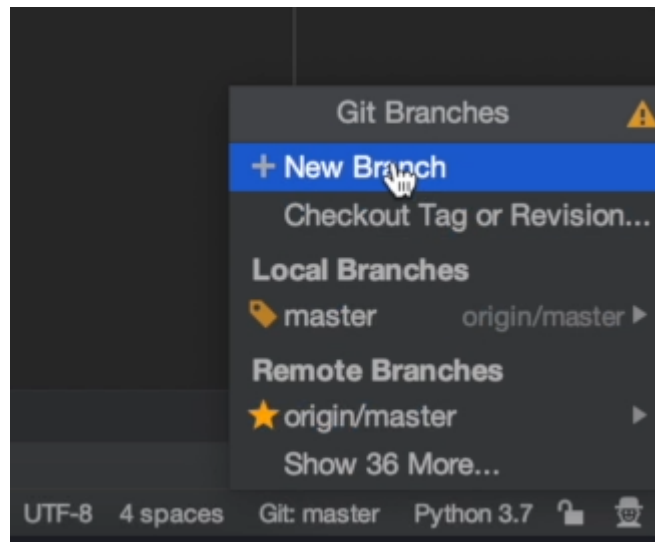
    def hello_world_func():
        logging.info("Hello World!")

    hello_world = PythonOperator(
        task_id='hello_world',
        python_callable=hello_world_func,
        dag=dag
    )

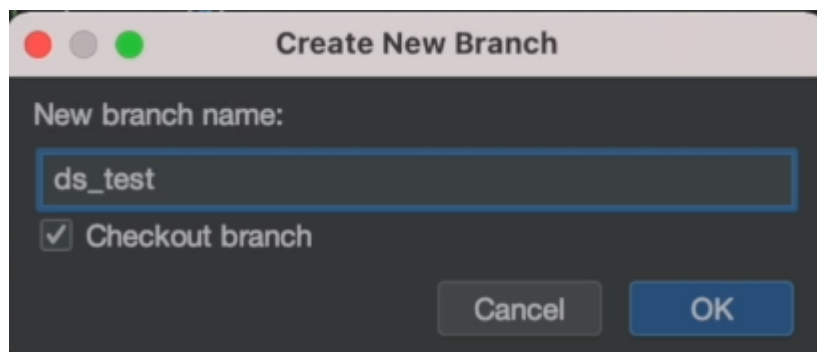
    dummy >> [echo_ds, hello_world]
```

> Отправка изменений в репозиторий

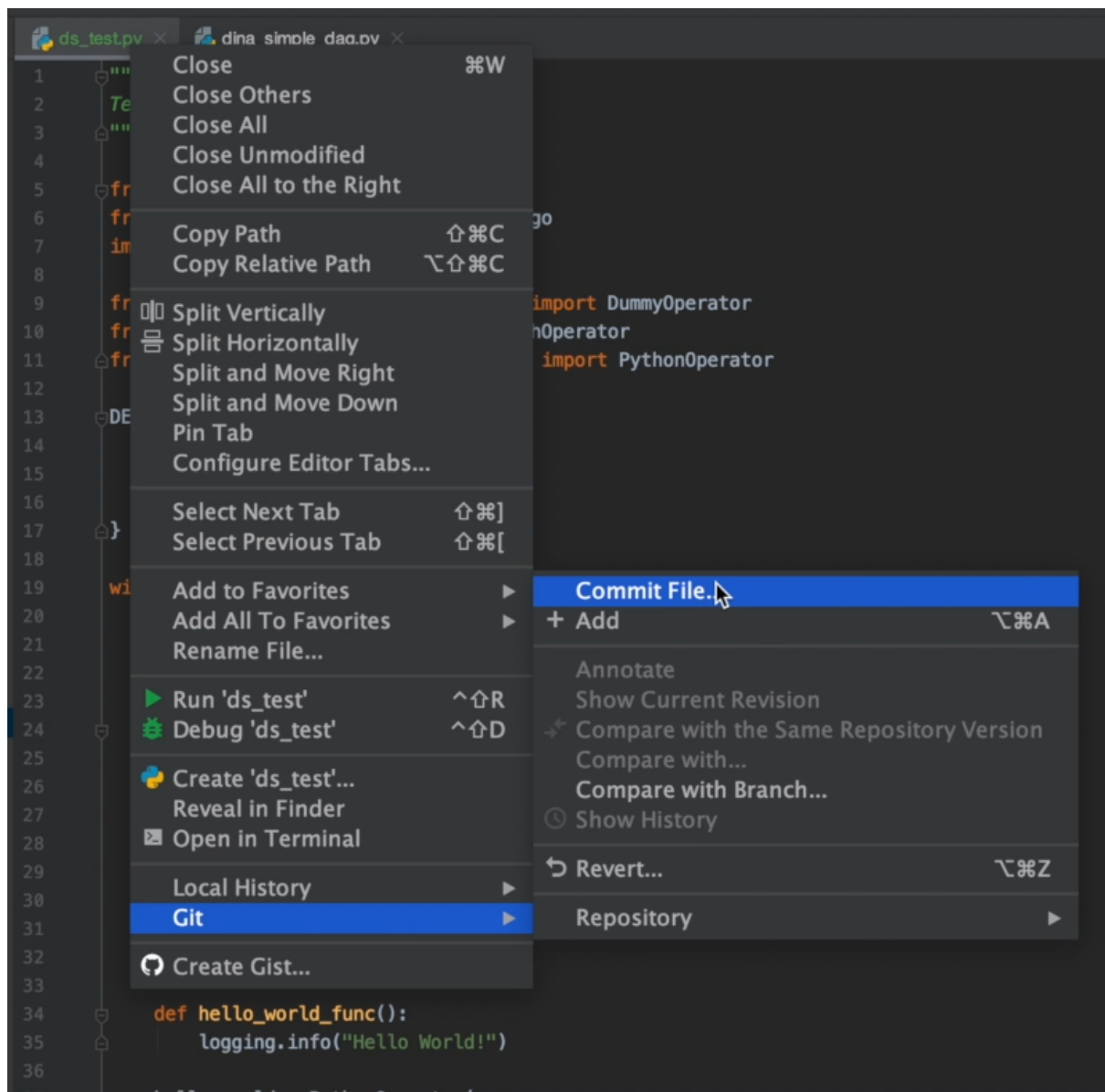
Заменим ветку, в которой мы работаем. Сейчас мы работали в ветке `master`, нам нужно создать новую. Для этого в правом нижнем углу PyCharm выбираем `Git: master -> NewBranch`.



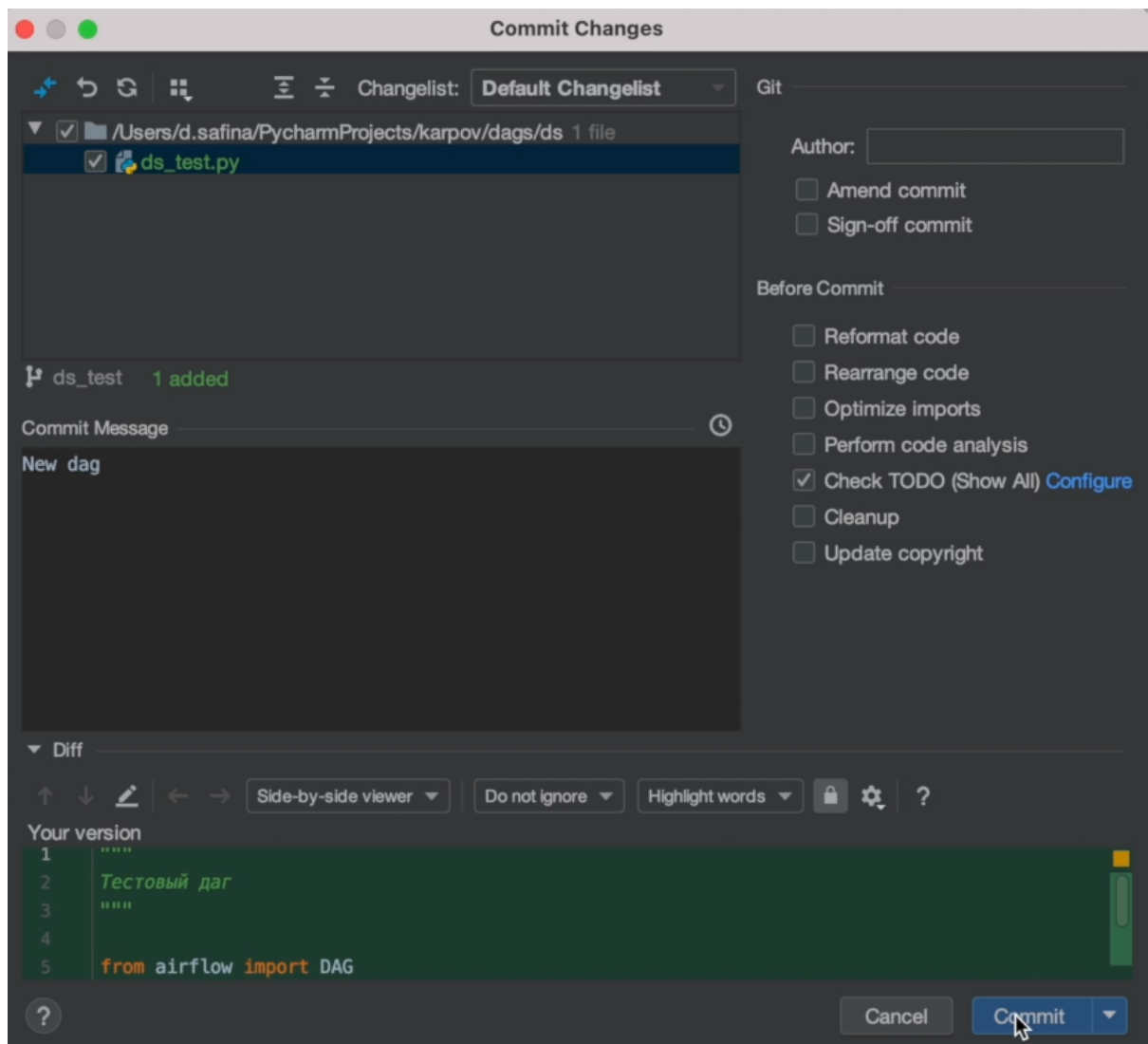
Создаём ветку `ds_test`.



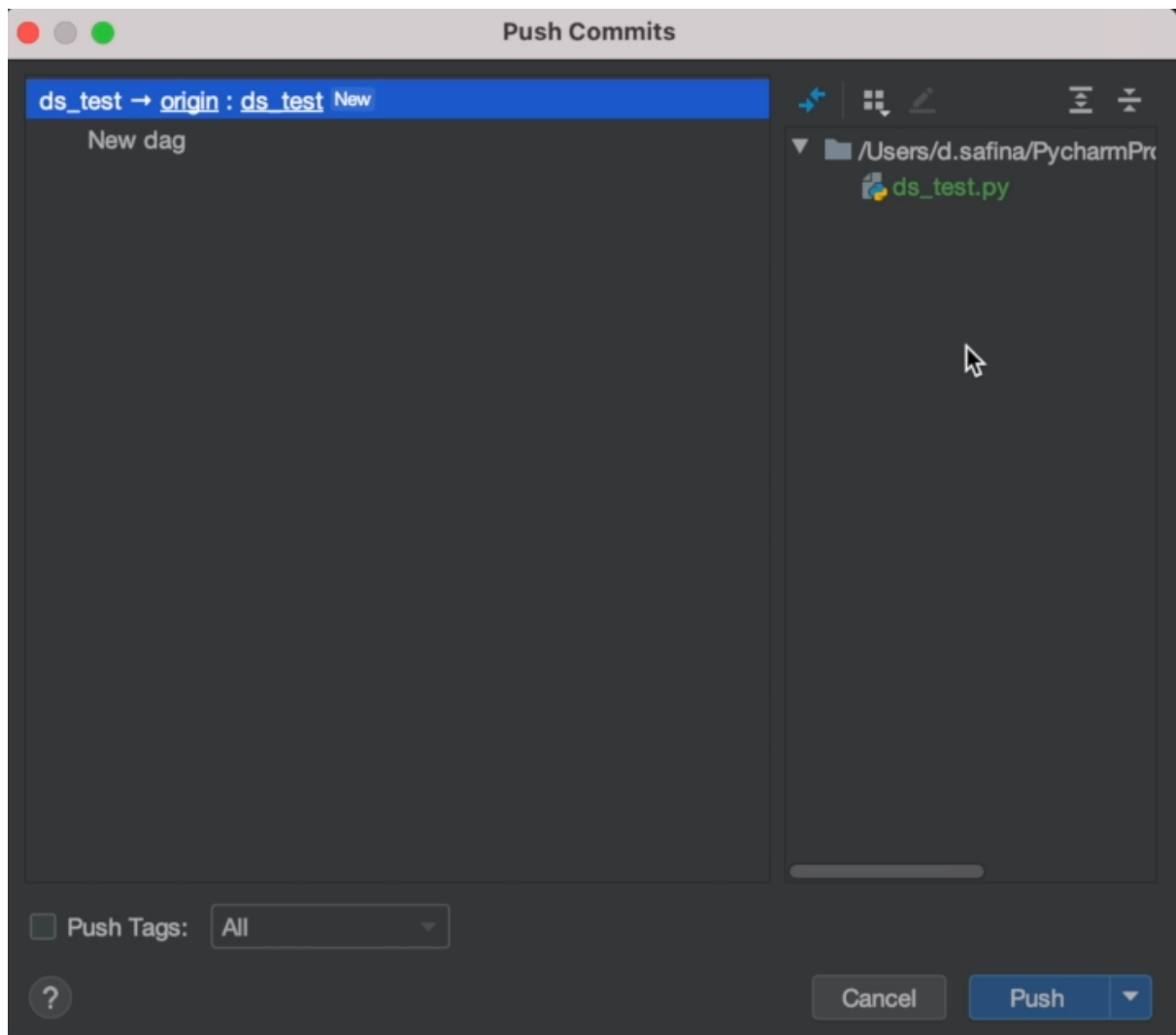
В созданной ветке commit-им изменения.



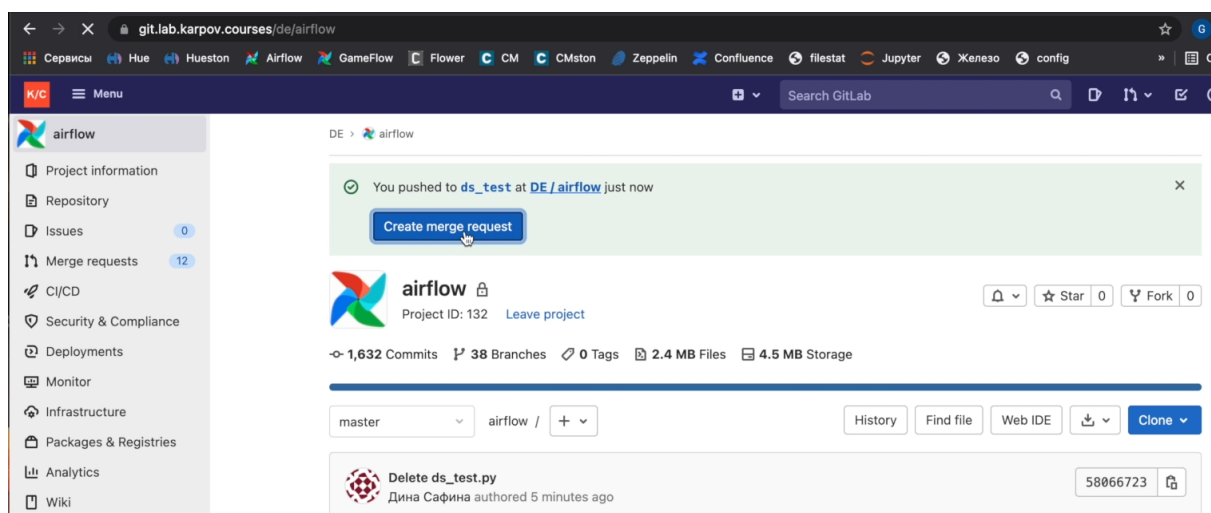
Заполняем Commit Message.



Push-им этот commit, вызывая соответствующее окно с помощью Ctrl+Shift+K.



Теперь заходим на наш git и видим, что можно создать merge request.



Создадим merge request.

Assignee [Assign to me](#)

Reviewer

Milestone


Labels

Merge options ☒ Delete source branch when merge request is accepted.
☐ Squash commits when merge request is accepted. [?](#)

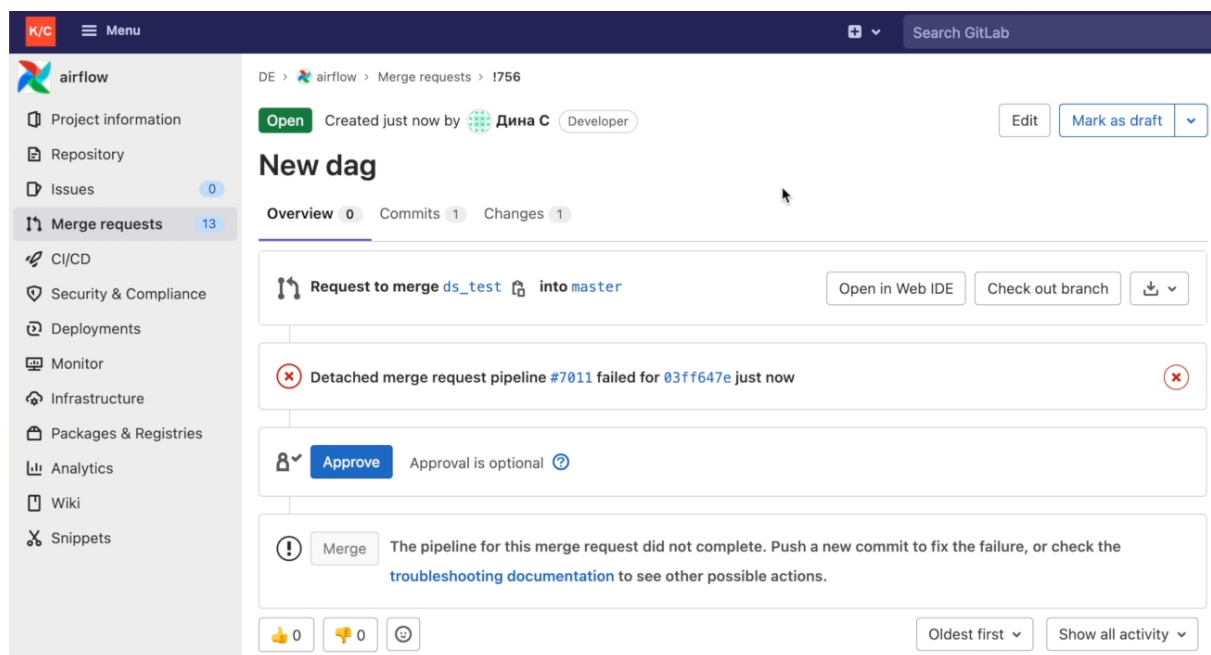
[Create merge request](#) [Cancel](#)

Commits 1 Changes 1

05 Dec, 2021 1 commit

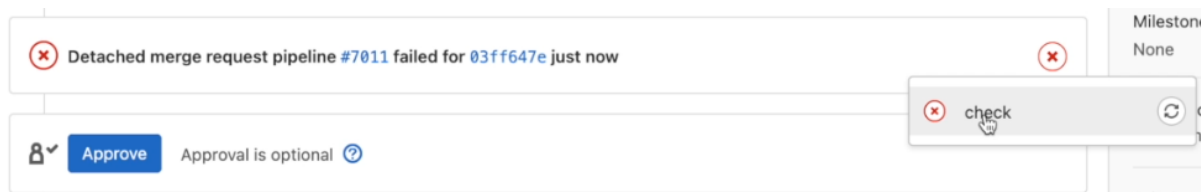
 **New dag**
Дина С authored just now

Видим, что проверка merge request-а не проходит.

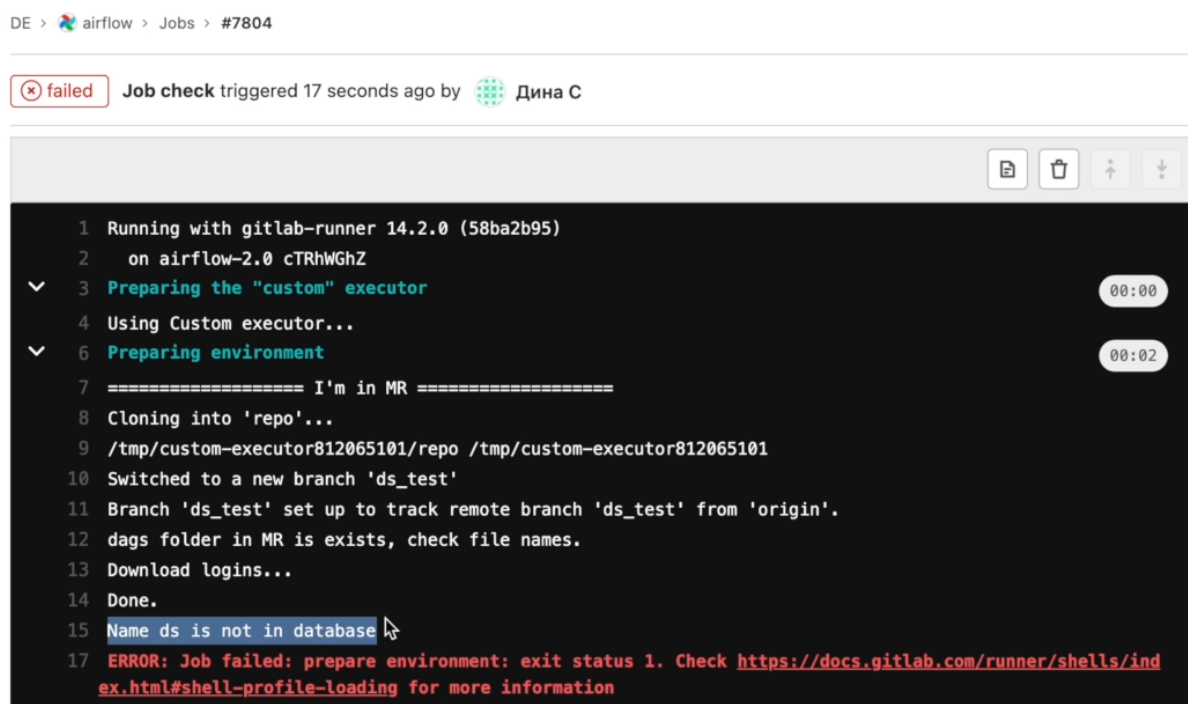


The screenshot shows the GitLab merge request interface for a project named 'airflow'. The left sidebar contains a navigation menu with options like Project information, Repository, Issues, Merge requests (selected), CI/CD, Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, Analytics, Wiki, and Snippets. The main content area shows the merge request details for 'New dag', created by 'Дина С' (Developer). It includes tabs for Overview, Commits, and Changes. A red error message states: 'Detached merge request pipeline #7011 failed for 03ff647e just now'. Below this, there is an 'Approve' button and a note that 'Approval is optional'. At the bottom, a message indicates that the pipeline for this merge request did not complete and suggests pushing a new commit or checking the troubleshooting documentation. The interface also shows buttons for 'Open in Web IDE', 'Check out branch', and 'Mark as draft'.

Заходим в test: failed.

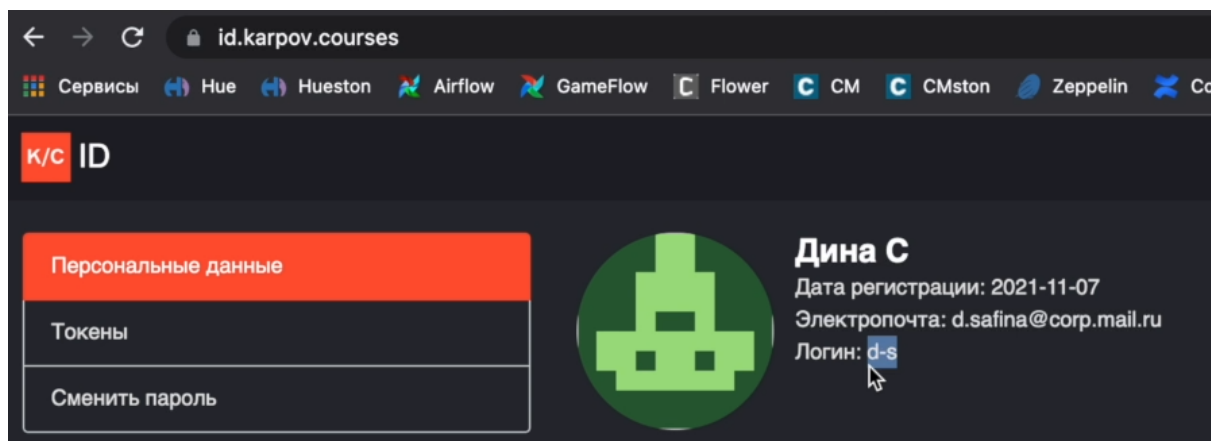


Видим причину: «Name ds is not in database».

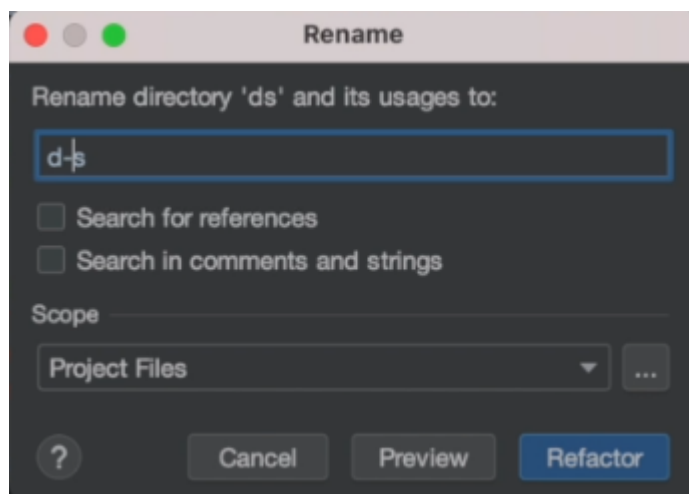


Чекер проверял соответствие названия папки и логина.

Корректное название логина можно уточнить на id.karpov.courses.



Меняем название папки на корректное: **d-s**, а также owner-а и tags.



```
"""
Тестовый даг
"""

from airflow import DAG
from airflow.utils.dates import days_ago
import logging

from airflow.operators.dummy_operator import DummyOperator
from airflow.operators.bash import BashOperator
from airflow.operators.python_operator import PythonOperator

DEFAULT_ARGS = {
    'start_date': days_ago(2),
    'owner': 'd-s',
    'poke_interval': 600
}

with DAG("ds_test"
        schedule_interval='@daily',
        default_args=DEFAULT_ARGS,
        max_active_runs=1,
        tags=['d-s']
    ) as dag:

    dummy = DummyOperator(task_id="dummy")

    echo_ds = BashOperator(
        task_id='echo_ds',
        bash_command='echo {{ ds }}',
        dag=dag
    )

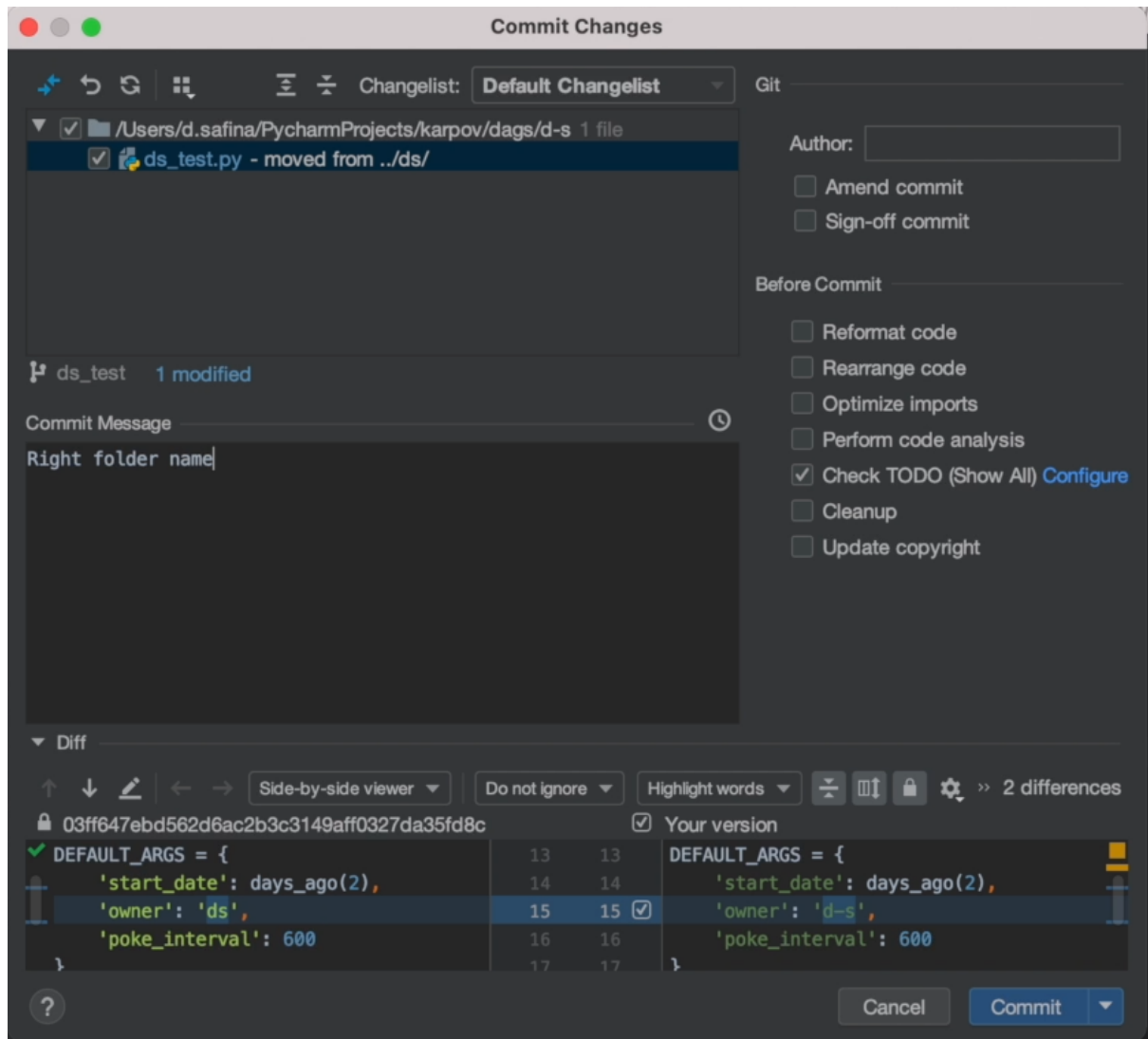
    def hello_world_func():
        logging.info("Hello World!")

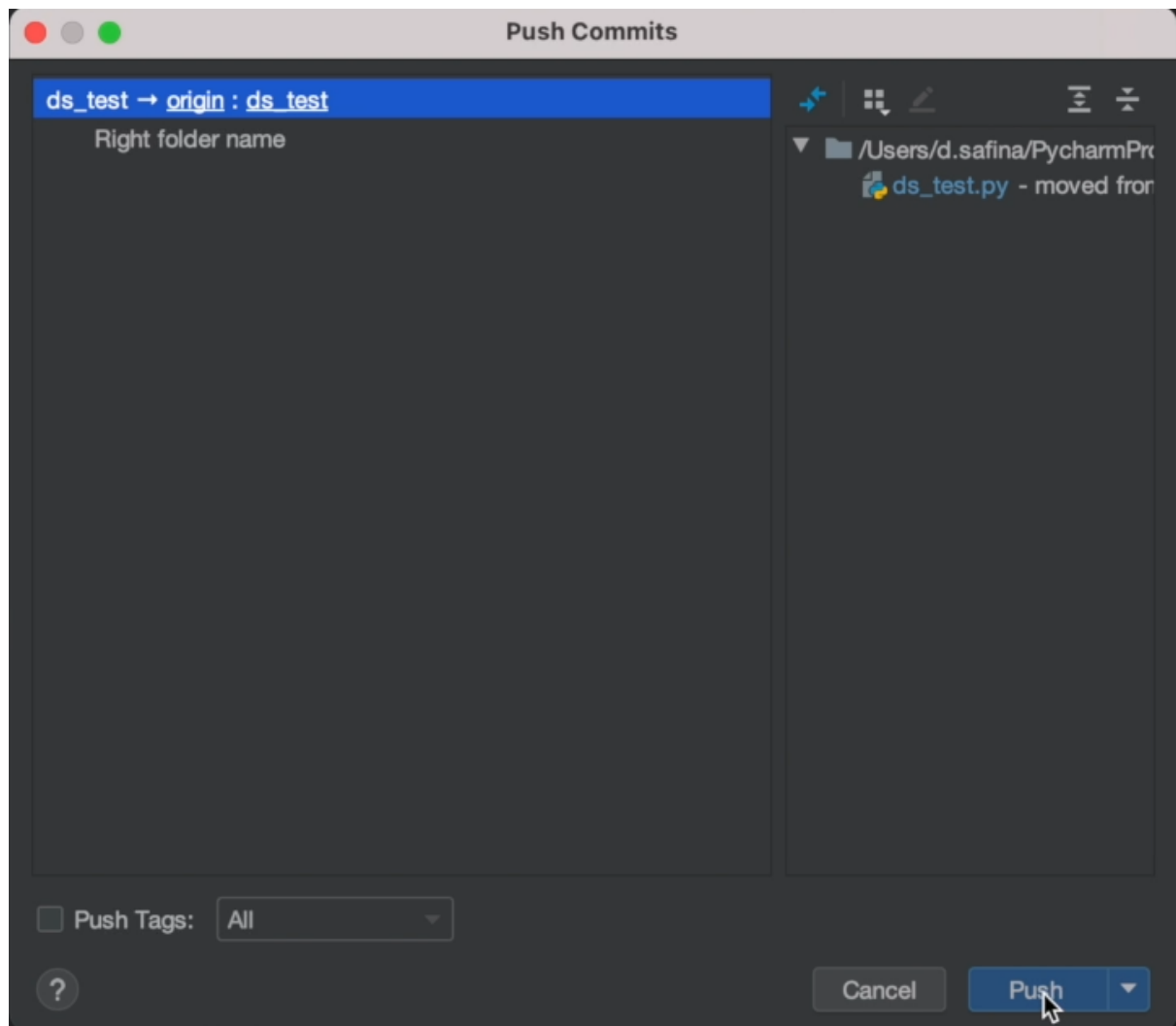
    hello_world = PythonOperator(
        task_id='hello_world',
```

```
python_callable=hello_world_func,
dag=dag
)

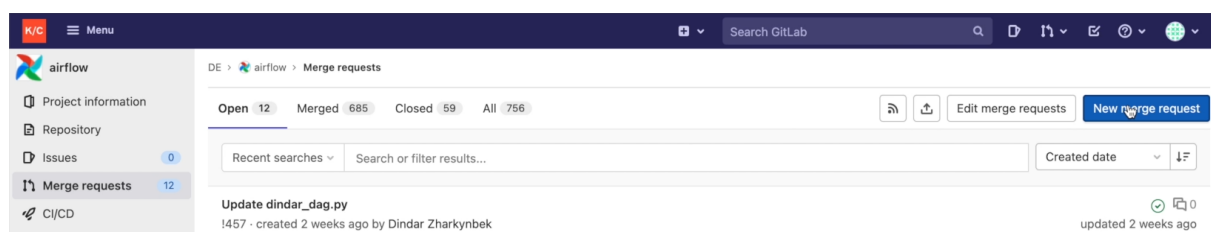
dummy >> [echo_ds, hello_world]
```

Снова commit-им и push-им изменения.

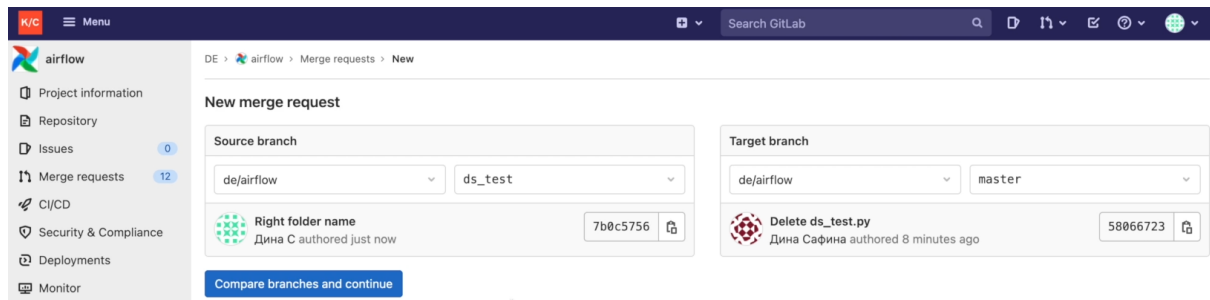




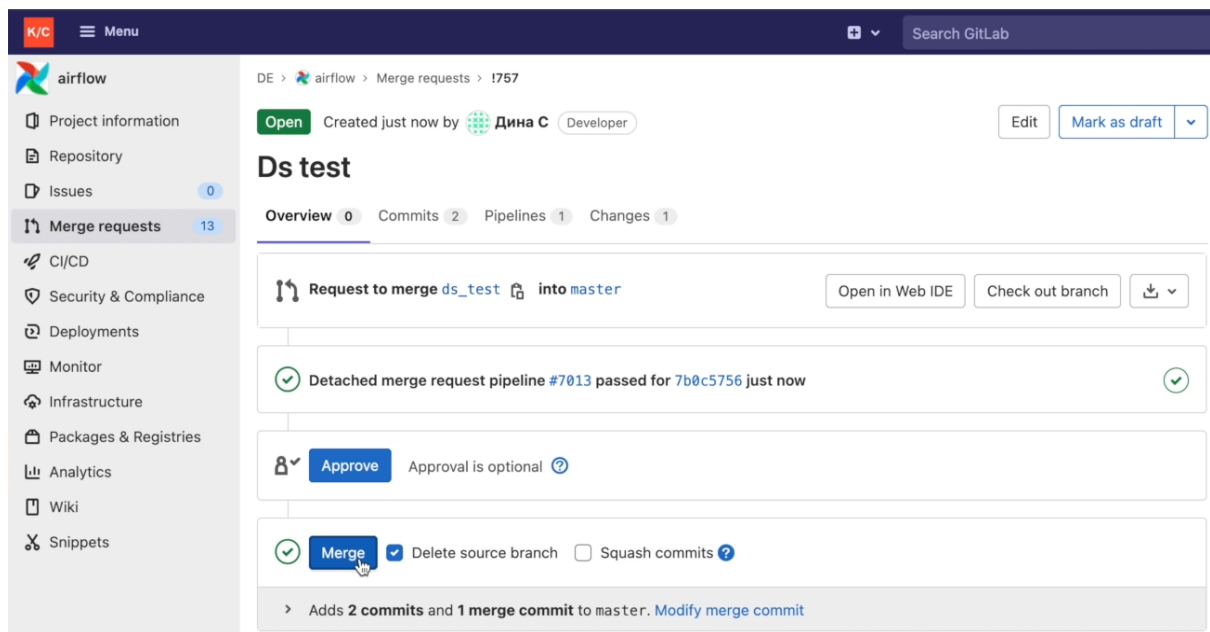
Заходим на git и самостоятельно создаём merge request, предварительно закроем предыдущий merge request (для этого необходимо зайти в него и нажать «[Close merge request](#)»).



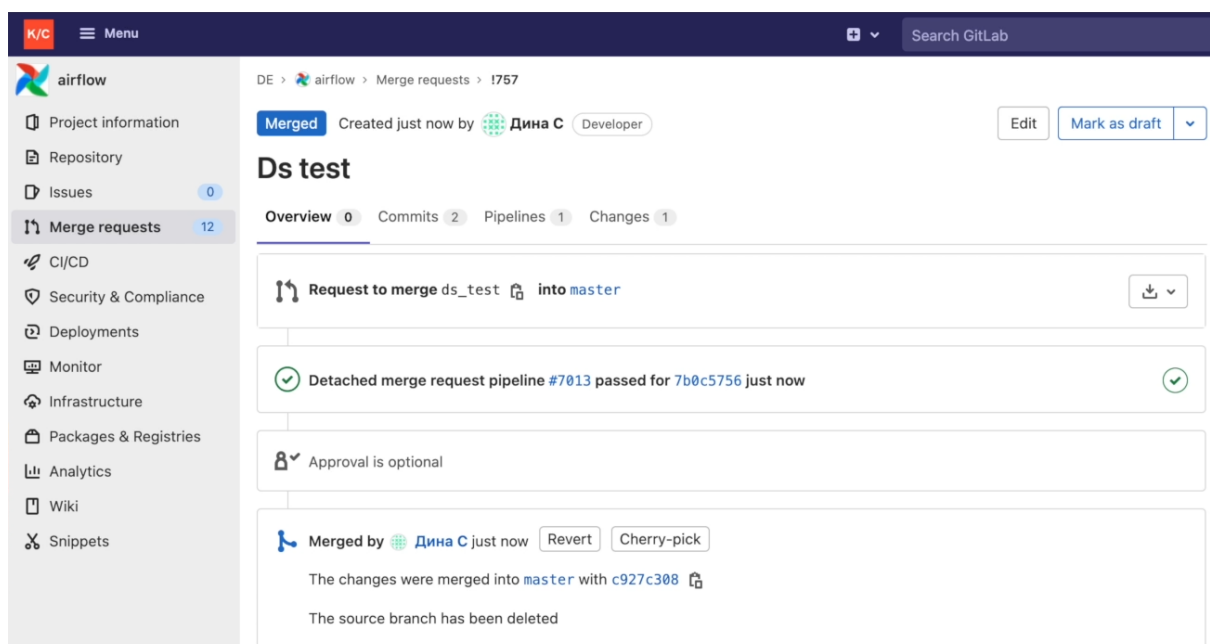
Выберем нашу ветку `ds_test` и создадим merge request ([Create merge request](#)).



Проверка прошла успешно.

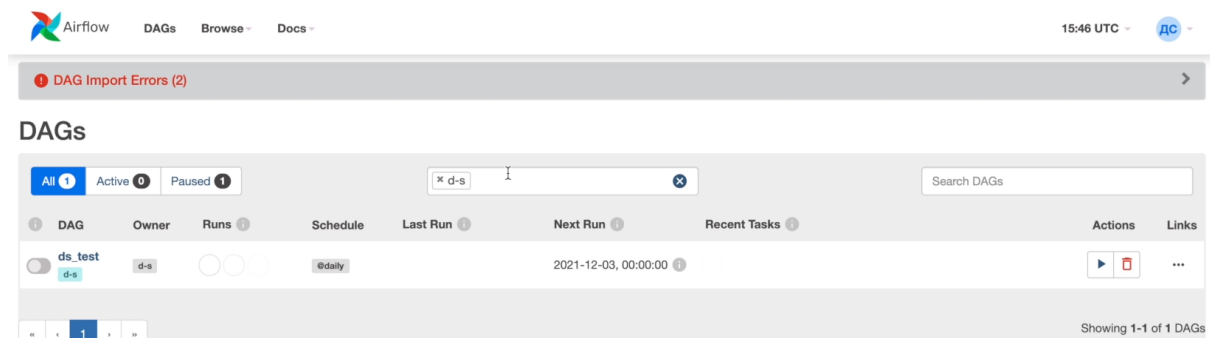


Теперь merge-им наш merge request в master.



> DAG на веб-сервере Airflow

Заходим на наш [Airflow](#) и по тегу **d-s** ищем нужный DAG.



Зайдём в этот DAG и запустим его.



Нажимаем на task **echo_ds** и в окне **Task Instance** нажимаем **Log**.

Task Instance: echo_ds
at: 2021-12-03, 00:00:00 UTC



Instance Details

Rendered

Log

All Instances

Filter Upstream

Download Log (by attempts):

1

Task Actions

Ignore All Deps

Ignore Task State

Ignore Task Deps

Run

Past

Future

Upstream

Downstream

Recursive

Failed

Clear

Past

Future

Upstream

Downstream

Mark Failed

Past

Future

Upstream

Downstream

Mark Success

Close

В логах видим дату выполнения DAG-а.

```
Airflow DAGs Browse Docs 15:46 UTC - DC
[2021-12-05, 15:46:32 UTC] {taskinstance.py:1035} INFO - Dependencies all met for <TaskInstance: ds_test.echo_ds scheduled__2021-12-03T00:00:00+00:00 [queued]>
[2021-12-05, 15:46:32 UTC] {taskinstance.py:1241} INFO -
[2021-12-05, 15:46:32 UTC] {taskinstance.py:1242} INFO - Starting attempt 1 of 1
[2021-12-05, 15:46:32 UTC] {taskinstance.py:1243} INFO -
[2021-12-05, 15:46:32 UTC] {taskinstance.py:1262} INFO - Executing <Task(BashOperator): echo_ds> on 2021-12-03 00:00:00+00:00
[2021-12-05, 15:46:32 UTC] {standard_task_runner.py:76} INFO - Running: ['airflow', 'tasks', 'run', 'ds_test', 'echo_ds', 'scheduled__2021-12-03T00:00:00+00:00', '--job-id', '1550']
[2021-12-05, 15:46:32 UTC] {standard_task_runner.py:77} INFO - Job 15500: Subtask echo_ds
[2021-12-05, 15:46:32 UTC] {standard_task_runner.py:52} INFO - Started process 9920 to run task
[2021-12-05, 15:46:32 UTC] {logging_mixin.py:109} INFO - Running <TaskInstance: ds_test.echo_ds scheduled__2021-12-03T00:00:00+00:00 [running]> on host f5b80008d3c8
[2021-12-05, 15:46:32 UTC] {taskinstance.py:1414} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_OWNER=ds
AIRFLOW_CTX_DAG_ID=ds_test
AIRFLOW_CTX_TASK_ID=echo_ds
AIRFLOW_CTX_EXECUTION_DATE=2021-12-03T00:00:00+00:00
AIRFLOW_CTX_DAG_RUN_ID=scheduled__2021-12-03T00:00:00+00:00
[2021-12-05, 15:46:33 UTC] {subprocess.py:62} INFO - Tmp dir root location:
/tmp
[2021-12-05, 15:46:33 UTC] {subprocess.py:74} INFO - Running command: ['bash', '-c', 'echo 2021-12-03']
[2021-12-05, 15:46:33 UTC] {subprocess.py:85} INFO - Output:
[2021-12-05, 15:46:33 UTC] {subprocess.py:89} INFO - 2021-12-03
[2021-12-05, 15:46:33 UTC] {subprocess.py:93} INFO - Command exited with return code 0
[2021-12-05, 15:46:33 UTC] {taskinstance.py:1280} INFO - Marking task as SUCCESS. dag_id=ds_test, task_id=echo_ds, execution_date=20211203T000000, start_date=20211205T154632, end_date=20211205T154633
[2021-12-05, 15:46:33 UTC] {local_task_job.py:154} INFO - Task exited with return code 0
[2021-12-05, 15:46:33 UTC] {local_task_job.py:264} INFO - 0 downstream tasks scheduled from follow-on schedule check
```


В логах таска hello_world видим строку "Hello World!".



```
*** Reading local file: /var/log/airflow/ds_test/hello_world/2021-12-04T00:00:00+00:00/1.log
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1035} INFO - Dependencies all met for <TaskInstance: ds_test.hello_world scheduled__2021-12-04T00:00:00+00:00 [queued]>
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1035} INFO - Dependencies all met for <TaskInstance: ds_test.hello_world scheduled__2021-12-04T00:00:00+00:00 [queued]>
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1241} INFO -
-----
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1242} INFO - Starting attempt 1 of 1
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1243} INFO -
-----
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1262} INFO - Executing <Task(PythonOperator): hello_world> on 2021-12-04 00:00:00+00:00
[2021-12-05, 15:46:37 UTC] {standard_task_runner.py:76} INFO - Running: ['airflow', 'tasks', 'run', 'ds_test', 'hello_world', 'scheduled__2021-12-04T00:00:00+00:00', '--job-id', '']
[2021-12-05, 15:46:37 UTC] {standard_task_runner.py:52} INFO - Started process 9924 to run task
[2021-12-05, 15:46:37 UTC] {standard_task_runner.py:77} INFO - Job 15507: Subtask hello_world
[2021-12-05, 15:46:37 UTC] {logging_mixin.py:109} INFO - Running <TaskInstance: ds_test.hello_world scheduled__2021-12-04T00:00:00+00:00 [running]> on host f5b80008d3c8
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1414} INFO - Exporting the following env vars:
AIRFLOW_CTX_DAG_OWNER=ds
AIRFLOW_CTX_DAG_ID=ds_test
AIRFLOW_CTX_TASK_ID=hello_world
AIRFLOW_CTX_EXECUTION_DATE=2021-12-04T00:00:00+00:00
AIRFLOW_CTX_DAG_RUN_ID=scheduled__2021-12-04T00:00:00+00:00
[2021-12-05, 15:46:37 UTC] {ds_test.py:35} INFO - Hello World!
[2021-12-05, 15:46:37 UTC] {python.py:152} INFO - Done. Returned value was: None
[2021-12-05, 15:46:37 UTC] {taskinstance.py:1280} INFO - Marking task as SUCCESS. dag_id=ds_test, task_id=hello_world, execution_date=20211204T000000, start_date=20211205T154637,
[2021-12-05, 15:46:37 UTC] {local_task_job.py:154} INFO - Task exited with return code 0
[2021-12-05, 15:46:37 UTC] {local_task_job.py:264} INFO - 0 downstream tasks scheduled from follow-on schedule check
```