

SDP DataLab

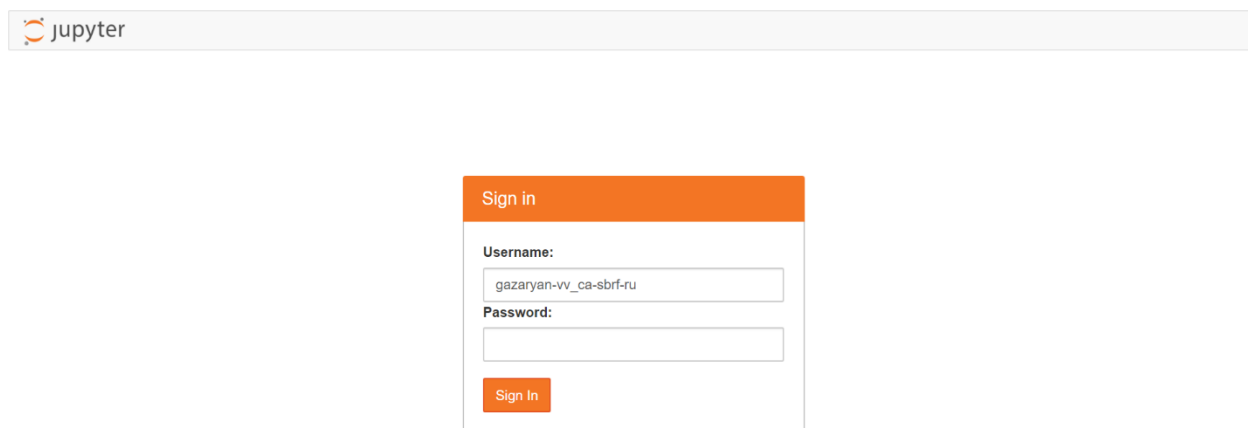
Руководство пользователя

Содержание

1. ВХОД В ПОДСИСТЕМУ	3
2. ВЫБОР ПРИЛОЖЕНИЯ	3
3. РАБОЧЕЕ МЕСТО.....	5

1. Вход в подсистему

Вход в систему осуществляется с помощью учётной записи Фабрики Данных (ФД), при этом пользователь должен пройти процедуру аутентификации. (Рисунок 1.1)



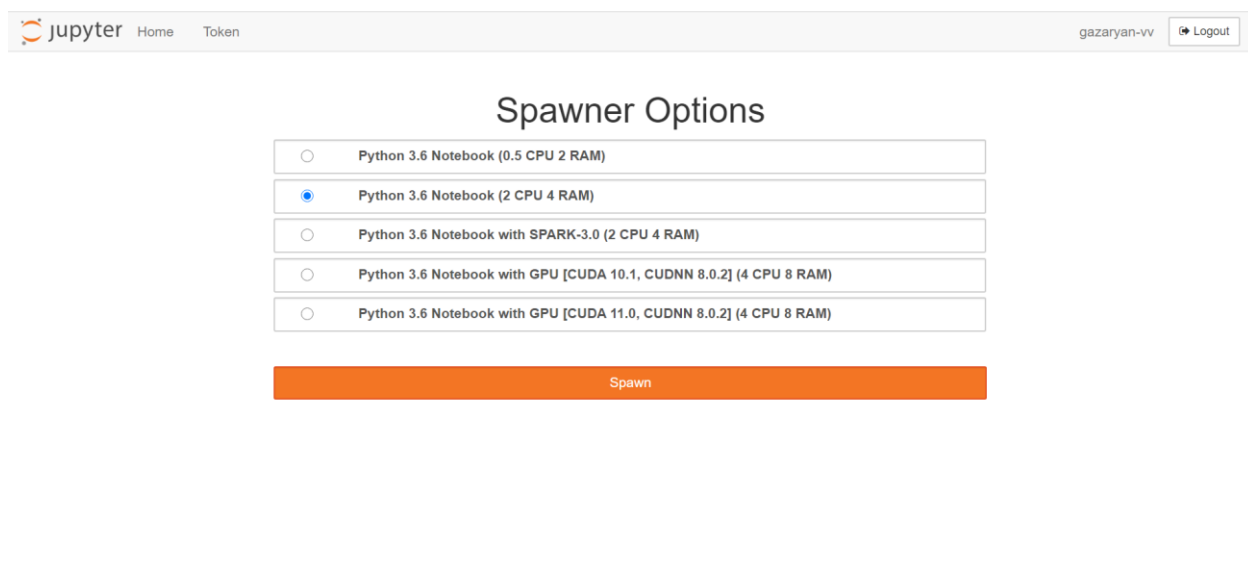
The image shows a Jupyter login interface. At the top left, there is a Jupyter logo and the text 'jupyter'. Below this is a 'Sign in' form with an orange header. The form contains two input fields: 'Username:' with the value 'gazaryan-vv_ca-sbrf-ru' and 'Password:'. Below the fields is an orange 'Sign In' button.

Рисунок 1.1 – вход в систему

2. Выбор приложения

Сразу после прохождения процесса аутентификации пользователям предоставляется расширяемый набор приложений с кратким описанием содержимого. (Рисунок 2.1).

(Список будет актуализироваться по мере выхода релизов).



The image shows the Jupyter 'Spawner Options' page. At the top left, there is a Jupyter logo and the text 'jupyter Home Token'. At the top right, there is a user name 'gazaryan-vv' and a 'Logout' button. The main content is titled 'Spawner Options' and lists five options with radio buttons:

- Python 3.6 Notebook (0.5 CPU 2 RAM)
- Python 3.6 Notebook (2 CPU 4 RAM)
- Python 3.6 Notebook with SPARK-3.0 (2 CPU 4 RAM)
- Python 3.6 Notebook with GPU [CUDA 10.1, CUDNN 8.0.2] (4 CPU 8 RAM)
- Python 3.6 Notebook with GPU [CUDA 11.0, CUDNN 8.0.2] (4 CPU 8 RAM)

At the bottom, there is an orange 'Spawn' button.

Рисунок 2.1 – список доступных приложений

В любой вариации (Таблица 2.1) доступен следующий предустановленный функционал:

- RHEL 8
- Python 3.6
- Java 8
- Krb5 workstation
- Hadoop client (интеграция с разделяемым хранилищем)

Таблица 2.1

Приложение	Описание
Python 3.6 Notebook	<ul style="list-style-type: none"> • Jupyter Notebook на базе python 3.6 (без предустановленных модулей)
Python 3.6 Notebook with SPARK	<ul style="list-style-type: none"> • Jupyter Notebook на базе python 3.6 (без предустановленных модулей) • Apache Spark 3.0
Python 3.6 Notebook with GPU	<ul style="list-style-type: none"> • Jupyter Notebook на базе python 3.6 (с предустановленными базовыми модулями: tensorflow, torch, keras, numpy, scipy, matplotlib, sklearn и т.д.) • Возможность работы с NVIDIA/GPU.

После выбора одного из приложений – осуществляется создание и запуск указанного контейнера. (Рисунок 2.2)

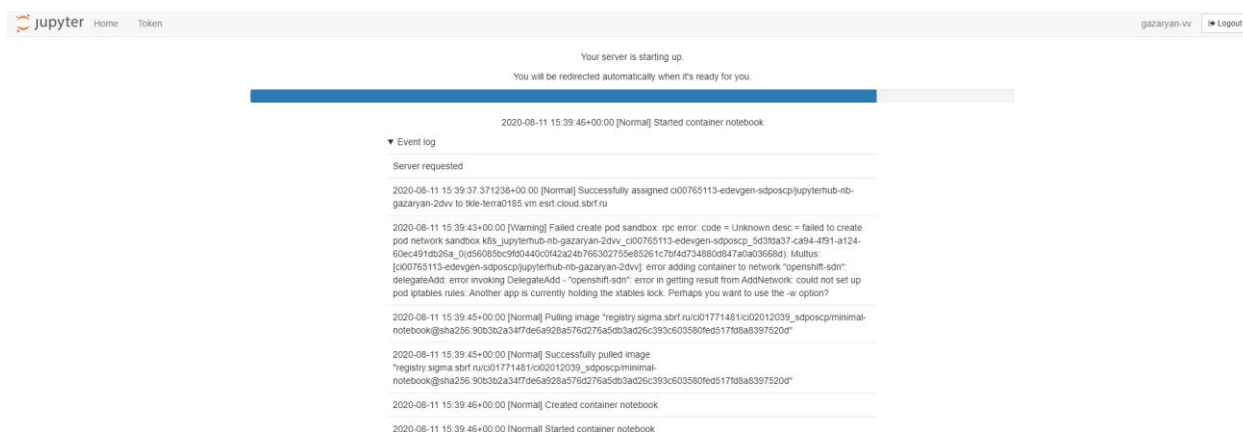


Рисунок 2.2 – создание и запуск контейнера

3. Рабочее место

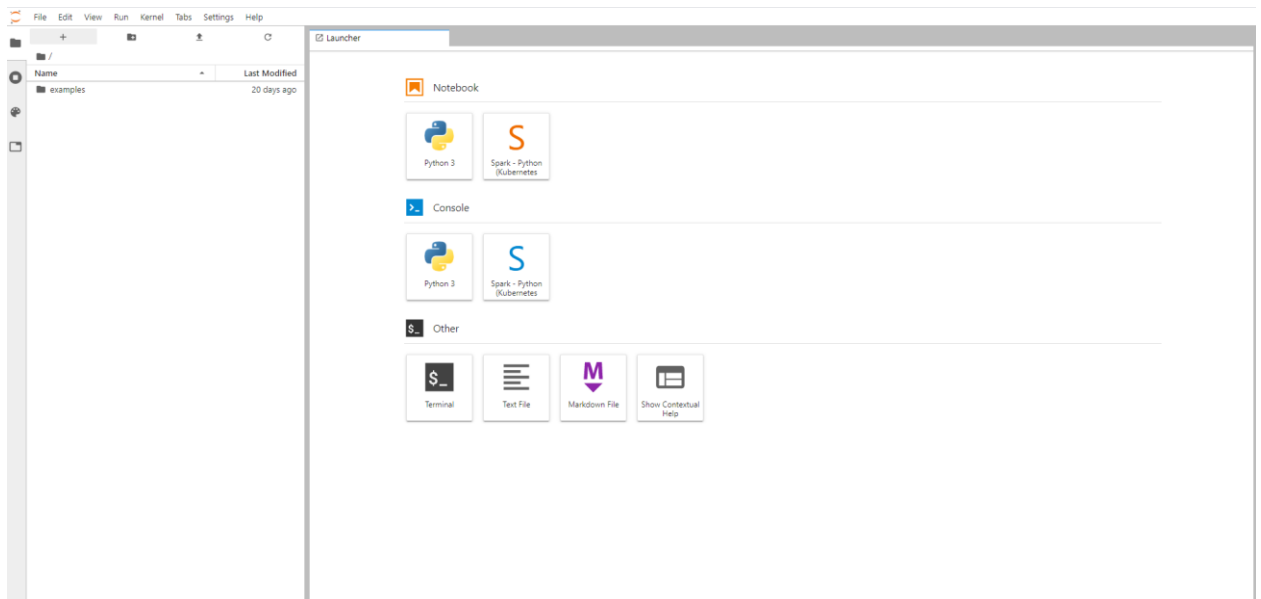


Рисунок 3.1 – рабочее место

Рабочее место пользователей – адаптивное расширение для Jupyter Notebook – JupyterLab (Рисунок 3.1). Для начала работы в секции Notebook выберите один из предустановленных kernels (Python 3, Spark – Python и т.д.)

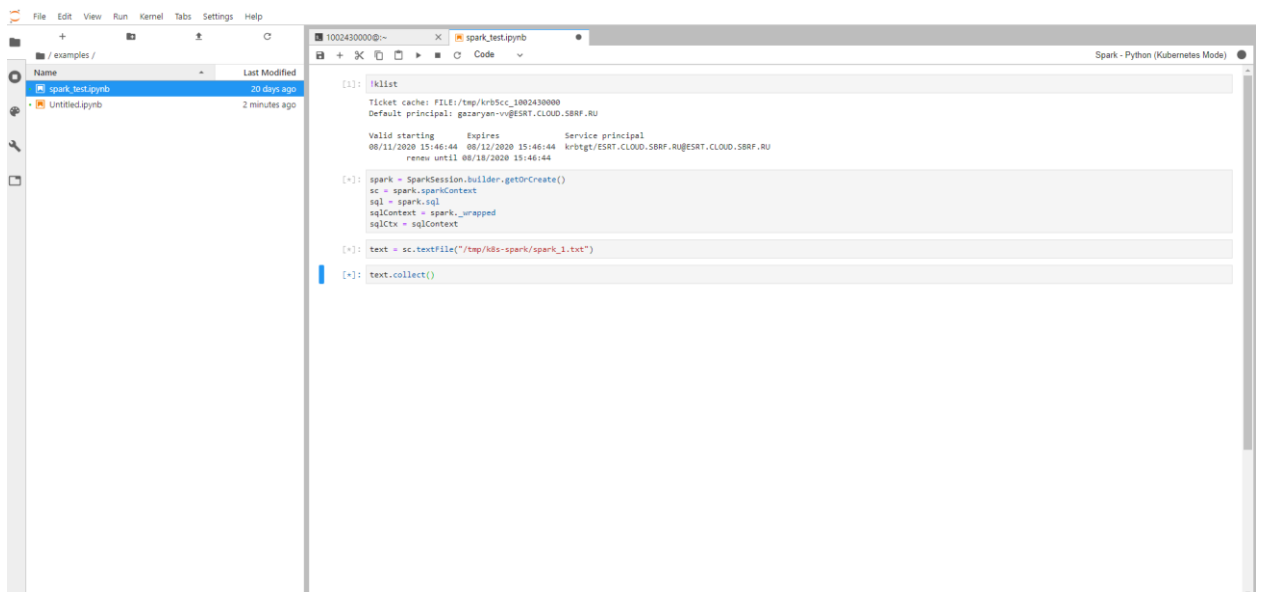


Рисунок 3.2 – Spark пример

Установка дополнительных python модулей осуществляется пользователями лично.

```
!pip install --no-cache-dir --upgrade tensorflow==2.2.0
```

В некоторых приложениях пользователям доступны предустановленные примеры ноутбуков, которые можно найти в папке `examples` в проводнике в левой части. (Например, «gru-test» Рисунок 3.3)

```
In [1]: import numpy
print('numpy:', numpy.__version__)

import scipy
print('scipy:', scipy.__version__)

import matplotlib
print('matplotlib:', matplotlib.__version__)

import IPython
print('iPython:', IPython.__version__)

import sklearn
print('scikit-learn:', sklearn.__version__)

import keras
print('keras: ', keras.__version__)

import theano
print('Theano: ', theano.__version__)

import tensorflow as tf
print('Tensorflow: ', tf.__version__)

import torch
print('Torch: ', torch.__version__)

numpy: 1.18.5
scipy: 1.4.1
matplotlib: 3.3.0
iPython: 7.16.1
scikit-learn: 0.23.2
keras: 2.4.3

WARNING (theano.tensor.blas): Using NumPy C-API based implementation for BLAS functions.

Theano: 1.0.5
Tensorflow: 2.2.0
Torch: 1.4.0
Keras: 2.4.3

In [2]: # tf.test.is_gpu_available(
#       cuda_only=False, min_cuda_compute_capability=None
# )
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))

Num GPUs Available: 1

In [3]: torch.cuda.current_device()
torch.cuda.device(0)
# torch.cuda.device_count()
torch.cuda.get_device_name(0)
# torch.cuda.is_available()

Out[3]: 'Tesla P100-PCIE-16GB'
```

Рисунок 3.3 – GPU пример

Остановка работы, перезапуск и выход из системы

После полного завершения работы - приложение стоит остановить, выбрав в навигационном меню *File – Hub Control Panel* (Рисунок 4.1)

В открывшейся панели управления нажать кнопку *Stop My Server* (Рисунок 4.2)

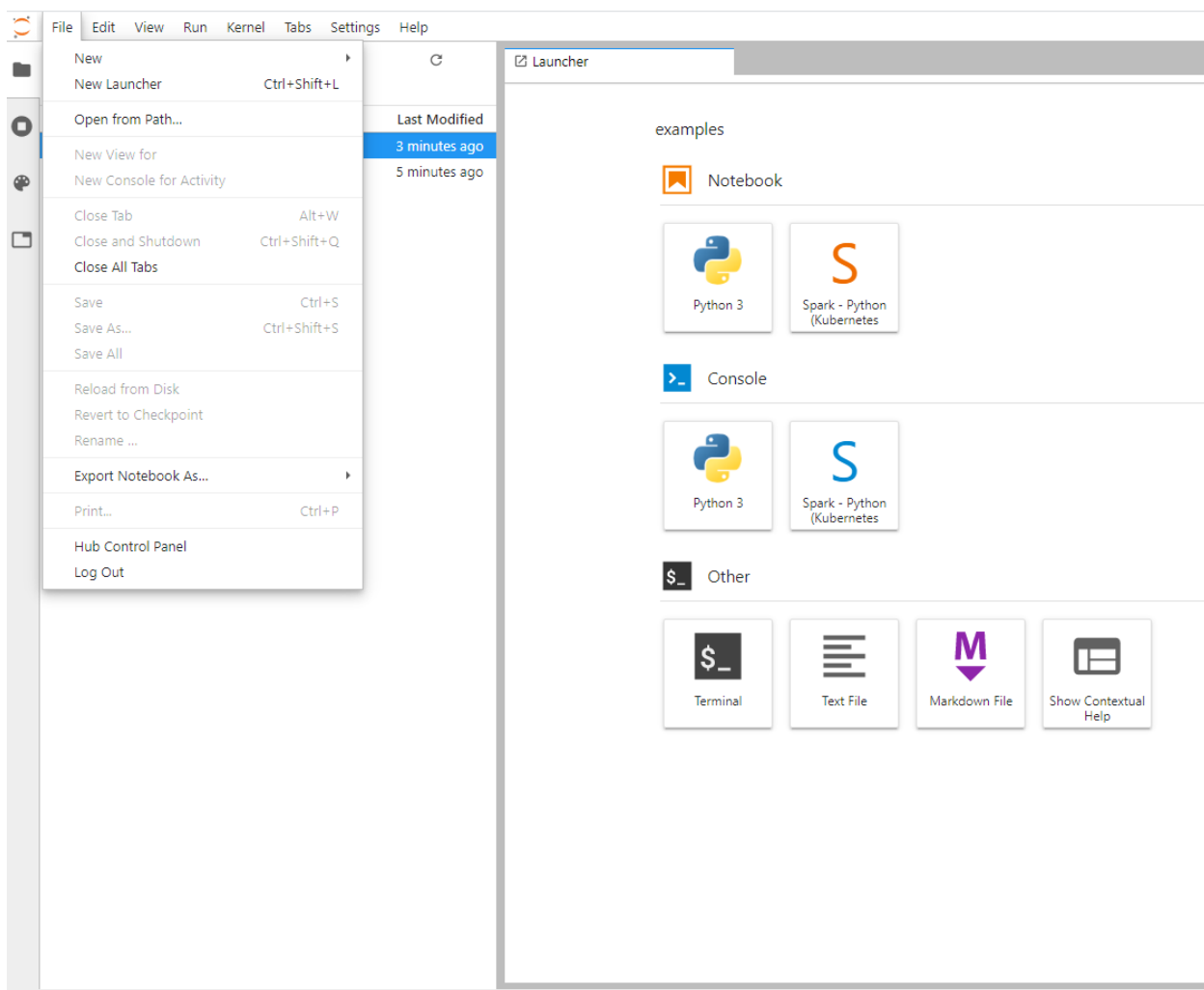


Рисунок 4.1 – навигационное меню

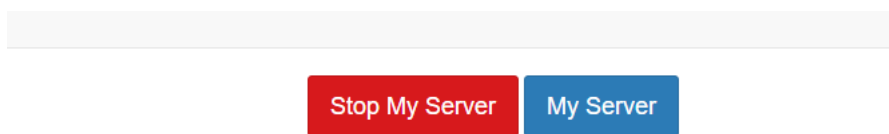


Рисунок 4.2 – Панель управления

Для запуска остановленного приложения нажмите *Start My Server* в панели управления.

Для выхода из системы в навигационном меню нажмите кнопку «*Log Out*».