

Установка TempletSDK в сессию пользователя TLJH с использованием nbgitpuller

Пусть задача делится на подзадачи:

- 1) Установка VirtualBox и создание виртуальной машины с ОС Linux Ubuntu. ISO можно скачать по ссылке <https://ubuntu.com/download/desktop?ref=nf>, на жестком диске необходимо выделить не меньше 30 ГБ, оперативной памяти достаточно 4 ГБ.
- 2) Установка на виртуальную машину Pip3 и JupyterHub. Python3 поставляется с Ubuntu по умолчанию. Это можно проверить.

**python3 --version**

```
user@UbuntuLinux:~$ python3 --version
Python 3.10.12
```

Для установки pip вводим следующую команду **sudo apt install python3-pip**

```
admin@UbuntuLinux:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc
  libjs-underscore libpython3-dev libpython3.10-dev python3-dev
  python3-distutils python3-setuptools python3-wheel python3.10-dev zlib1g-dev
Suggested packages:
  apache2 | lighttpd | httpd python-setuptools-doc
The following NEW packages will be installed:
  javascript-common libexpat1-dev libjs-jquery libjs-sphinxdoc
  libjs-underscore libpython3-dev libpython3.10-dev python3-dev
  python3-distutils python3-pip python3-setuptools python3-wheel
  python3.10-dev zlib1g-dev
0 upgraded, 14 newly installed, 0 to remove and 5 not upgraded.
Need to get 8 013 kB of archives.
After this operation, 34,2 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Вводим Y, ждем окончание загрузки

Удостоверимся, что pip установлен на компьютере **python3 --version**

```
admin@UbuntuLinux:~$ pip --version
pip 22.0.2 from /usr/lib/python3/dist-packages/pip (python 3.10)
admin@UbuntuLinux:~$
```

После обновляем pip до последней версии **sudo pip3 install --upgrade pip**

```
admin@UbuntuLinux:~$ sudo pip3 install --upgrade pip
Requirement already satisfied: pip in /usr/lib/python3/dist-packages (22.0.2)
Collecting pip
  Downloading pip-23.3.1-py3-none-any.whl (2.1 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 3.8 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.2
    Not uninstalling pip at /usr/lib/python3/dist-packages, outside environment /usr
    Can't uninstall 'pip'. No files were found to uninstall.
Successfully installed pip-23.3.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
admin@UbuntuLinux:~$ pip --version
pip 23.3.1 from /usr/local/lib/python3.10/dist-packages/pip (python 3.10)
admin@UbuntuLinux:~$
```

Устанавливаем JupyterLab `pip install jupyterlab`

Устанавливаем JupyterHub

```
sudo apt-get install npm nodejs
```

```
sudo python3 -m pip install jupyterhub
```

```
sudo npm install -g configurable-http-proxy
```

Далее необходимо создать файл конфигурации `sudo jupyterhub --generate-config`

Затем используйте следующую команду для редактирования файла конфигурации: `sudo nano jupyterhub_config.py`

Необходимо вставить следующую информацию в файл:

```
c.JupyterHub.port = 8888
```

```
c.Spawner.default_url = '/lab'
```

```
c.Authenticator.admin_users = {'admin'} # список администраторов
```

```
c.Authenticator.allowed_users = {'user1', 'user2', 'user3'} # список пользователей
```

Пример:

```
admin@UbuntuLinux: ~
GNU nano 6.2          jupyterhub config.py
# Configuration file for jupyterhub.

c = get_config() #noqa

c.JupyterHub.port=8888
c.Spawner.default_url='/lab'
c.Authenticator.admin_users={'admin'}
c.Authenticator.allowed_users={'user1','user2','user3'}
#-----
# Application(SingletonConfigurable) configuration
#-----
## This is an application.

## The date format used by logging formatters for %(asctime)s
# Default: '%Y-%m-%d %H:%M:%S'
# c.Application.log_datefmt = '%Y-%m-%d %H:%M:%S'

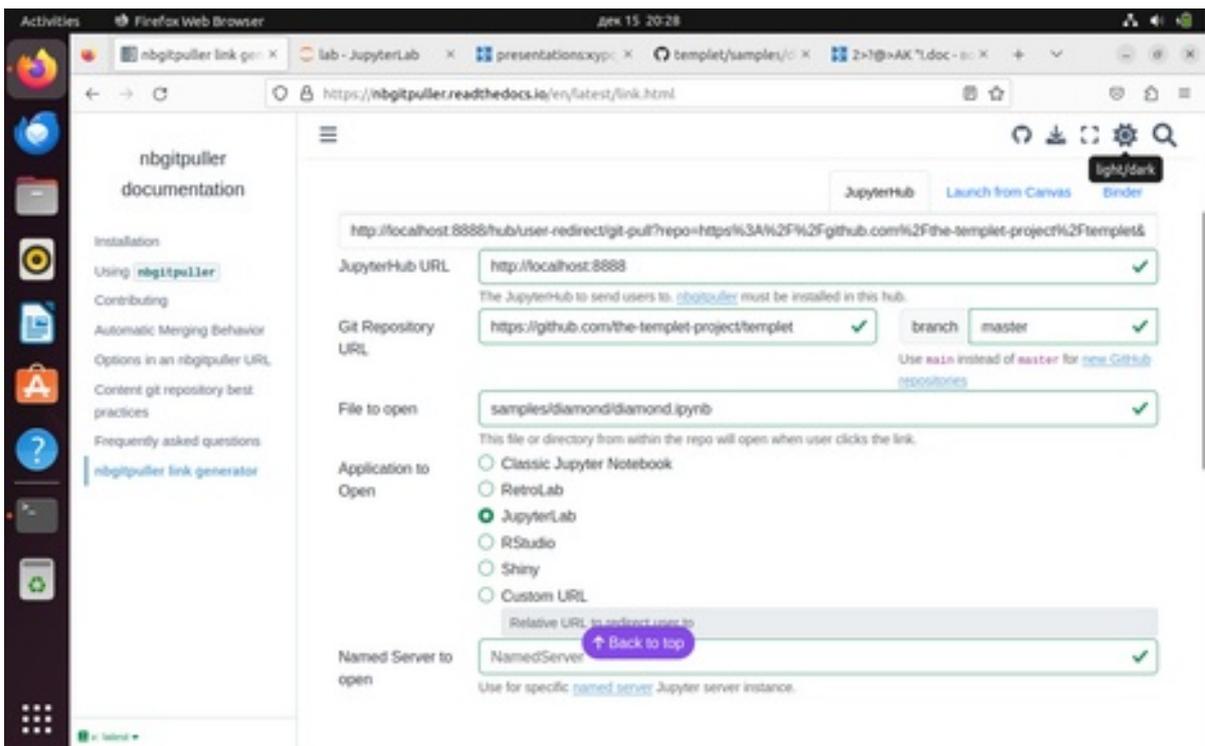
## The Logging format template
# Default: '[%(name)s]%(highlevel)s %(message)s'
# c.Application.log_format = '%(name)s]%(highlevel)s %(message)s'
[ Wrote 1533 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste      ^J Justify   ^_ Go To Line
```

После нажимаем ctrl+o, enter, ctrl+x.

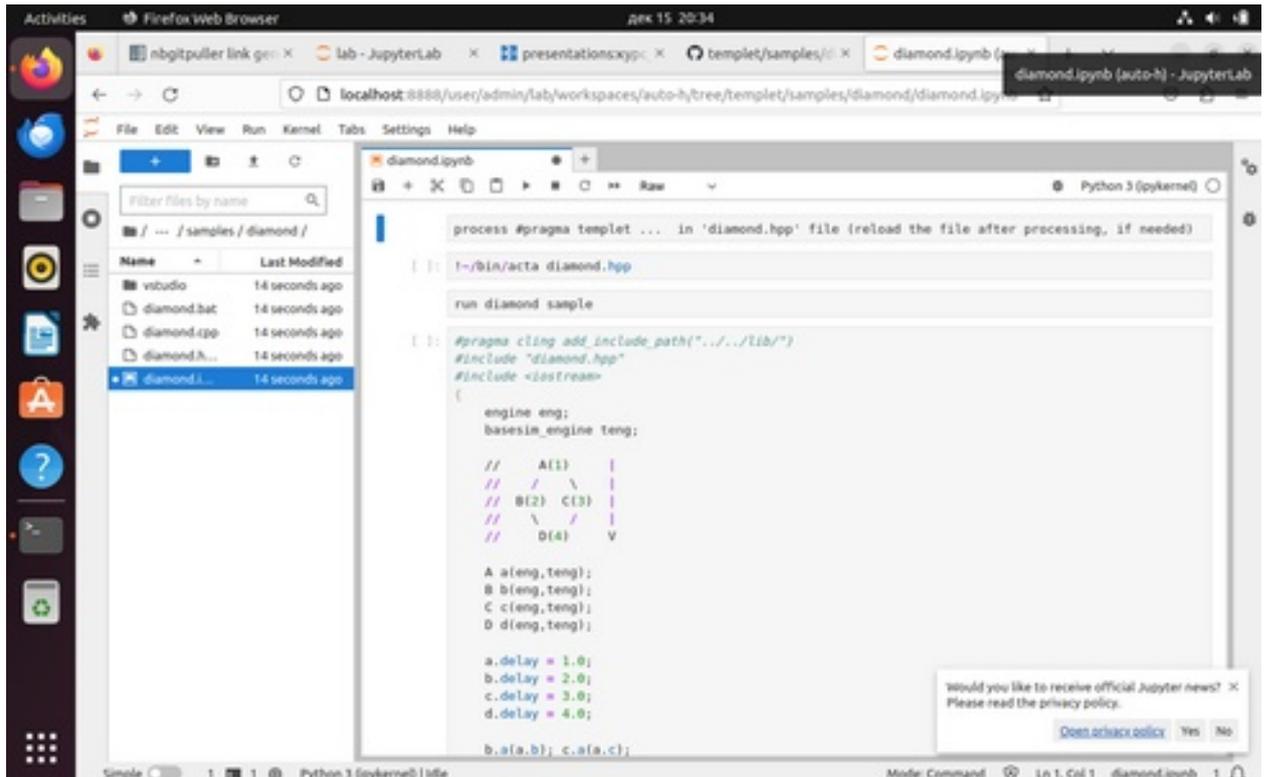
- 3) Необходимо установить библиотеку nbgitpuller `pip install nbgitpuller`
- 4) Проверим работу nbgitpuller. Запустим jupyterhub командой `sudo jupyterhub`

После этого найдем репозиторий nbgitpuller на GitHub и перейдем на страницу с генератором ссылок, ссылку на страницу можно найти в файле README.md

Вот пример данных для генерации ссылки



Переходим по сгенерированной ссылке.



Видим, что вытягивается репозиторий и открывается файл, путь которого указан в поле "File to open".

- 5) Теперь необходимо установить ядро C++. Для этого необходимо установить conda, установить conda можно следуя инструкции на <https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html>

Если после установки возникает ошибка

```
admin@UbuntuLinux:~$ conda
conda: command not found
```

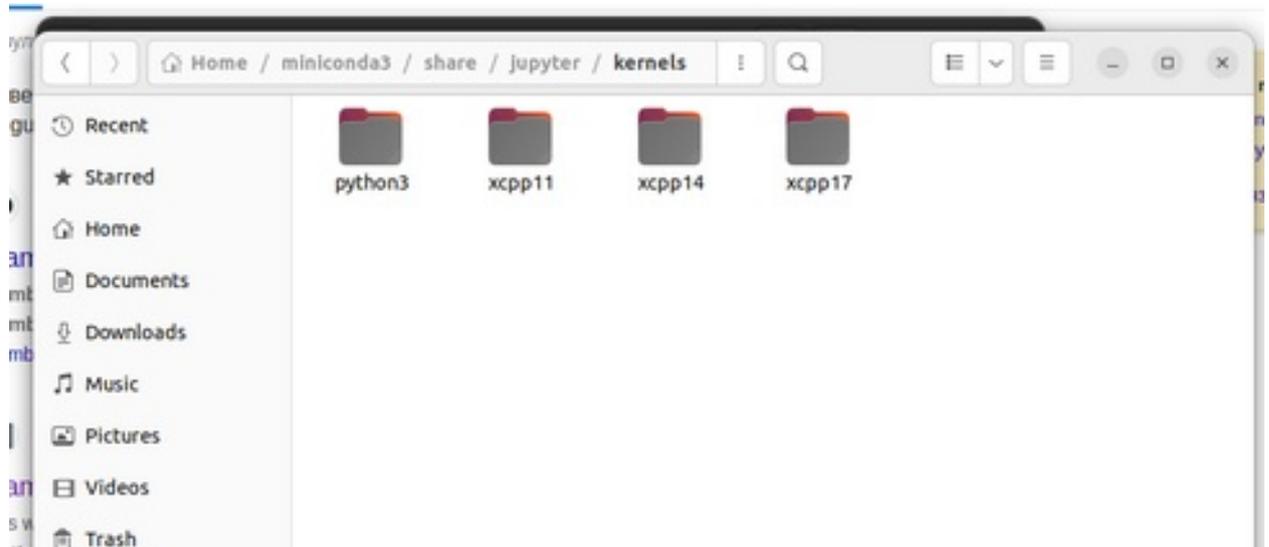
Необходимо установить conda в переменную среды PATH:

```
admin@UbuntuLinux:~$ $PATH
bash: /home/admin/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin: No such file or directory
admin@UbuntuLinux:~$ export PATH="/home/admin/miniconda3/condabin:$PATH"
admin@UbuntuLinux:~$ $PATH
bash: /home/admin/miniconda3/condabin:/home/admin/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin: No such file or directory
```

Устанавливаем xeus-cling `conda install xeus-cling -c conda-forge`

После этого мы столкнулись с проблемой, что jupyter не видел установленные ядра, но они находились в miniconda.

```
admin@UbuntuLinux:~$ jupyter kernelspec list
Available kernels:
  python3      /home/admin/.local/share/jupyter/kernels/python3
admin@UbuntuLinux:~$
```

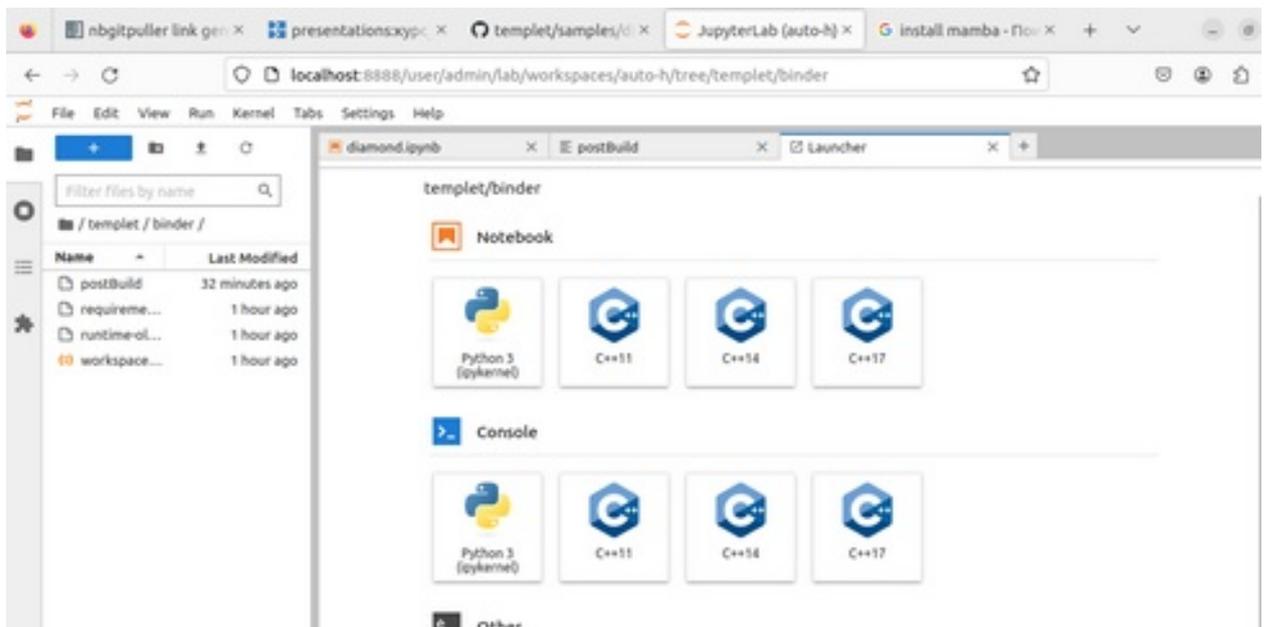


Все исправилось после того как мы добавили путь к bin miniconda в PATH

```
admin@UbuntuLinux:~$ export PATH="/home/admin/miniconda3/bin":$PATH
admin@UbuntuLinux:~$ $PATH
bash: /home/admin/miniconda3/bin:/home/admin/miniconda3/condabin:/home/admin/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin: No such file or directory
admin@UbuntuLinux:~$ jupyter kernelspec list
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
Available kernels:
python3      /home/admin/.local/share/jupyter/kernels/python3
xcpp11      /home/admin/miniconda3/share/jupyter/kernels/xcpp11
xcpp14      /home/admin/miniconda3/share/jupyter/kernels/xcpp14
xcpp17      /home/admin/miniconda3/share/jupyter/kernels/xcpp17
admin@UbuntuLinux:~$
```

В итоге единственное решение которое помогло, переместить ядра в .local//share/jupyter/kernels

```
admin@UbuntuLinux:~$ jupyter kernelspec list
[ListKernelSpecs] WARNING | Config option 'kernel_spec_manager_class' not recognized by 'ListKernelSpecs'.
0.00s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.
Available kernels:
python3      /home/admin/.local/share/jupyter/kernels/python3
xcpp11      /home/admin/.local/share/jupyter/kernels/xcpp11
xcpp14      /home/admin/.local/share/jupyter/kernels/xcpp14
xcpp17      /home/admin/.local/share/jupyter/kernels/xcpp17
admin@UbuntuLinux:~$
```



Пример сборки TemplеtSDK x3

Заходим в любой файл .ipynb, вызываем postBuild. Перед этим необходимо настроить файл postBuild относительно текущего каталога!

Например, если я запускаю файл из файла каталога 2-ого уровня вложенности пути будут выглядеть так:

```
README.md x diamond.ipynb x postBuild x +
1 #!/bin/bash
2 ls
3 #mamba install -y -c conda-forge xeus-cling
4 #mamba create -y --name everest python=2.7
5
6 cd ../../cgen
7 g++ skel.cpp -o skel
8 g++ cgen.cpp lexer.cpp parse.cpp -o cgen
9 g++ acta.cpp -o acta
10
11 mv skel ../../bin/skel
12 mv cgen ../../bin/cgen
13 mv acta ../../bin/acta
14
15 chmod 755 ../../bin/skel
16 chmod 755 ../../bin/cgen
17 chmod 755 ../../bin/acta
18 chmod 755 ../../bin/everest.sh
19
```

Я буду запускать из файла diamond.ipynb

```
[1]: !chmod 755 ../../../../binder/postBuild
```

```
[2]: ! ../../../../binder/postBuild
```

После этого можем увидеть что в папке bin создались нужные файлы.

📁 / templet / bin /

<input type="checkbox"/> Name	▲	Last Modified	File Size
<input type="checkbox"/> acta		8 seconds ago	29.4 KB
<input type="checkbox"/> acta.exe		8 hours ago	20.5 KB
<input type="checkbox"/> cgen		9 seconds ago	92.5 KB
<input type="checkbox"/> cgen.exe		8 hours ago	45.5 KB
<input type="checkbox"/> CppVS2015redistributabl...		8 hours ago	147 B
<input type="checkbox"/> everest.sh		8 hours ago	419 B
<input type="checkbox"/> libcrypto-1_1-x64.dll		8 hours ago	2.6 MB
<input type="checkbox"/> libcurl-x64.dll		8 hours ago	999.1 KB
<input type="checkbox"/> libssl-1_1-x64.dll		8 hours ago	505.1 KB
<input type="checkbox"/> skel		10 seconds ago	98 KB
<input type="checkbox"/> skel.exe		8 hours ago	42 KB
<input type="checkbox"/> start4python2.sh		8 hours ago	865 B