

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ
АКАДЕМИКА С. П. КОРОЛЕВА» (САМАРСКИЙ
УНИВЕРСИТЕТ)

Отчёт по дополнительному заданию по курсу «Операционные системы».

Принудительное завершение сеанса пользователя в ТЛН по заданному
времени предельной длительности сеанса

Выполнил:
Филиппов Е.А.
группа 6303

Проверил:
Востокин С. В.

Задание: принудительное завершение сеанса пользователя в TLJH по заданному времени предельной длительности сеанса.

Процедура выполнения:

Если у вас уже есть Ubuntu и установлен TLJH и Python:

1. Создайте файл с названием `session_timeout.py` через команду:

```
touch session_timeout.py
```

или создайте его в проводнике

2. Откройте файл `session_timeout.py`:

```
nano session_timeout.py
```

или через проводник

3. Вставьте данный код:

```
import os
import time
import psutil
import requests

def get_jupyter_sessions(hub_url, token):
    jupyter_not_admin_sessions = []
    for process in psutil.process_iter(['pid', 'username']):
        username = process.info["username"]
        if username.startswith("jupyter-") and not
is_admin_user(username[8:], hub_url, token):
            jupyter_not_admin_sessions.append(process.info['pid'])
    return jupyter_not_admin_sessions

def is_admin_user(username, hub_url, token):
    try:
        response = requests.get(f"{hub_url}/users/{username}",
headers={"Authorization": f"token {token}"}, )
        response.raise_for_status()

        if response.status_code == 200:
            return response.json().get("admin")
        return True
    except Exception as e:
        print("Network error: ", e)
        return True

def main():
    max_time = 60 # Максимальное время сеанса в секундах
    url = "localhost" # Домен или айпи
    token = "5a2a4827b018412dbebd18c2869ddbc9" # Токен

    hub_url = "http://" + url + "/hub/api"

    jupyter_not_admin_sessions = get_jupyter_sessions(hub_url, token)
```

```

for session_pid in jupyter_not_admin_sessions:
    session_start_time = os.path.getctime(f'/proc/{session_pid}')
    elapsed_time = time.time() - session_start_time
    if elapsed_time > max_time:
        os.system(f"sudo kill {session_pid}")
        print(f"{session_pid} disabled")

if __name__ == "__main__":
    main()

```

Измените значение `max_time` на то количество секунд, которое дается сеансу.

Измените значение `url` на домен или ip вашего JupyterHub (по умолчанию `localhost`).

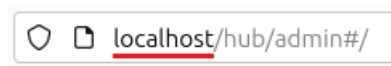


Рисунок 1 – Адресная строка JupyterHub

Измените значение `token` на токен вашего JupyterHub. Для его генерации зайдите на вкладку Token в JupyterHub.

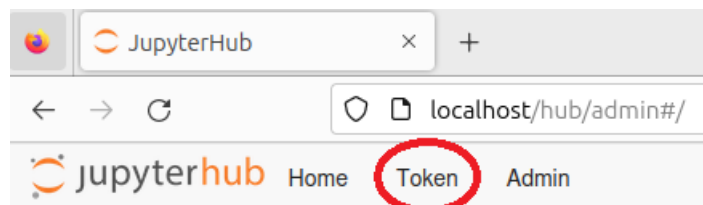


Рисунок 2 – Вкладка Token

Нажмите на кнопку Request new API token. При желании можете указать имя токена в строке Note.

Request new API token

Note

note to identify your new token

This note will help you keep track of what your tokens are for.

Token expires in

Never ▼

You can configure when your token will expire.

Рисунок 3 – Генерация токена

Замените значение token в python скрипте на сгенерированный токен.

Сохраните изменения через CTRL+S и выйдите.

4. Зайдите в root:

```
sudo su - root
```

После этой команды необходимо будет ввести пароль от текущего пользователя Ubuntu.

5. Введите строку:

```
apt install python3-psutil
```

6. Выйдите из root:

```
exit
```

7. Зайдите в планировщик автоматического запуска:

```
crontab -e
```

8. В самом конце введите строку:

```
* * * * * /usr/bin/python3 /home/username/session_timeout.py
```

Замените в данной строке username на ваше имя пользователя Ubuntu.

Данная вариация строки написана с учётом, что python3 находится в директории /usr/bin и python скрипт находится в директории /home/username. Измените, если потребуется.

Данная строка будет запускать python скрипт каждую минуту.

Задача Cron

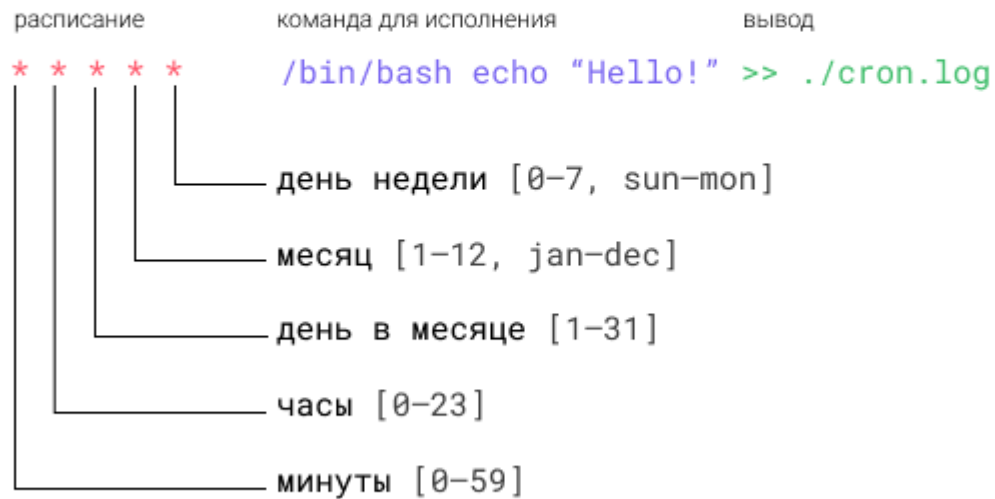


Рисунок 4 – Объяснение строки автозапуска

9. Нажмите CTRL+S и CTRL+X

10. Зайдите в настройки sudo:

```
sudo visudo
```

11. В самом конце введите команду:

```
username ALL=(ALL) NOPASSWD: /bin/kill
```

Замените в данной строке username на ваше имя пользователя в Ubuntu.

12. Нажмите CTRL+S и CTRL+X

Готово! Теперь данный python скрипт будет выполняться каждую минуту и будет завершать все jupyter сеансы, которые не являются администраторами и которые превышают необходимое количество секунд, указанных в python скрипте.

Результаты работы при максимальном времени сеанса в 60 секунд:

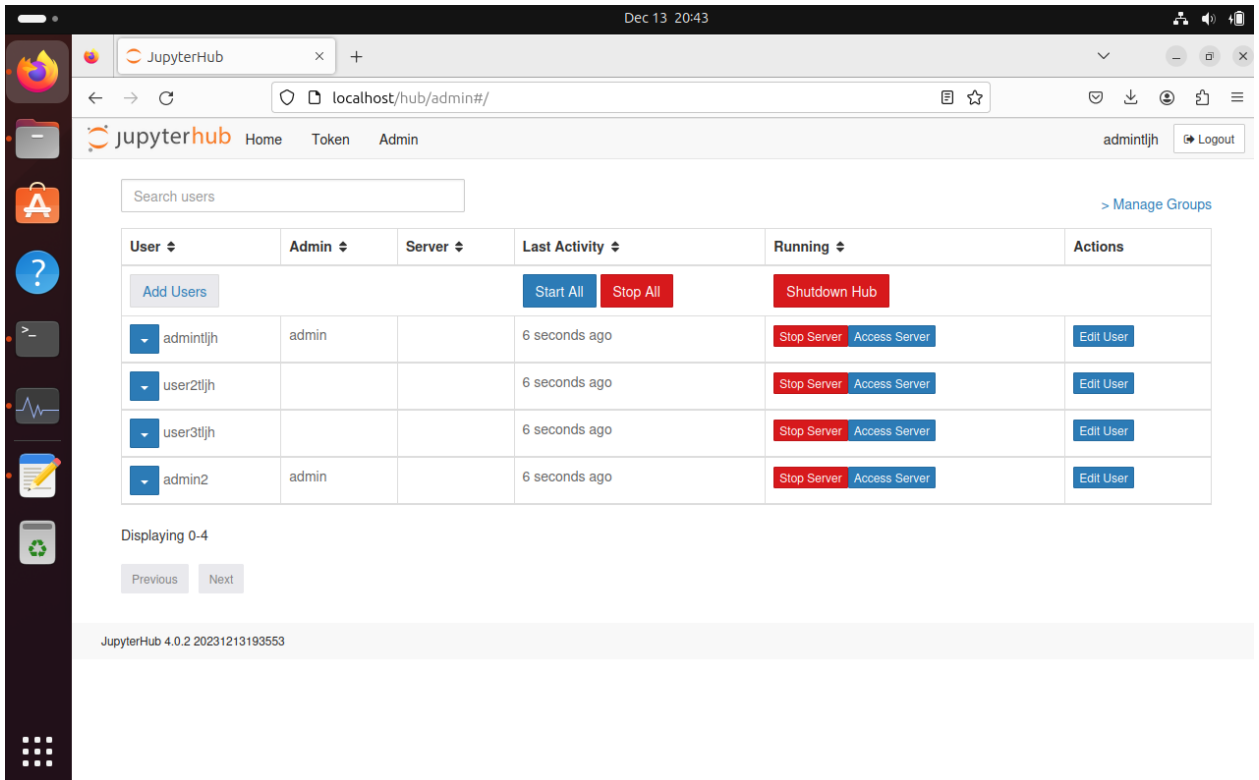


Рисунок 5 – Сеансы до срабатывания скрипта

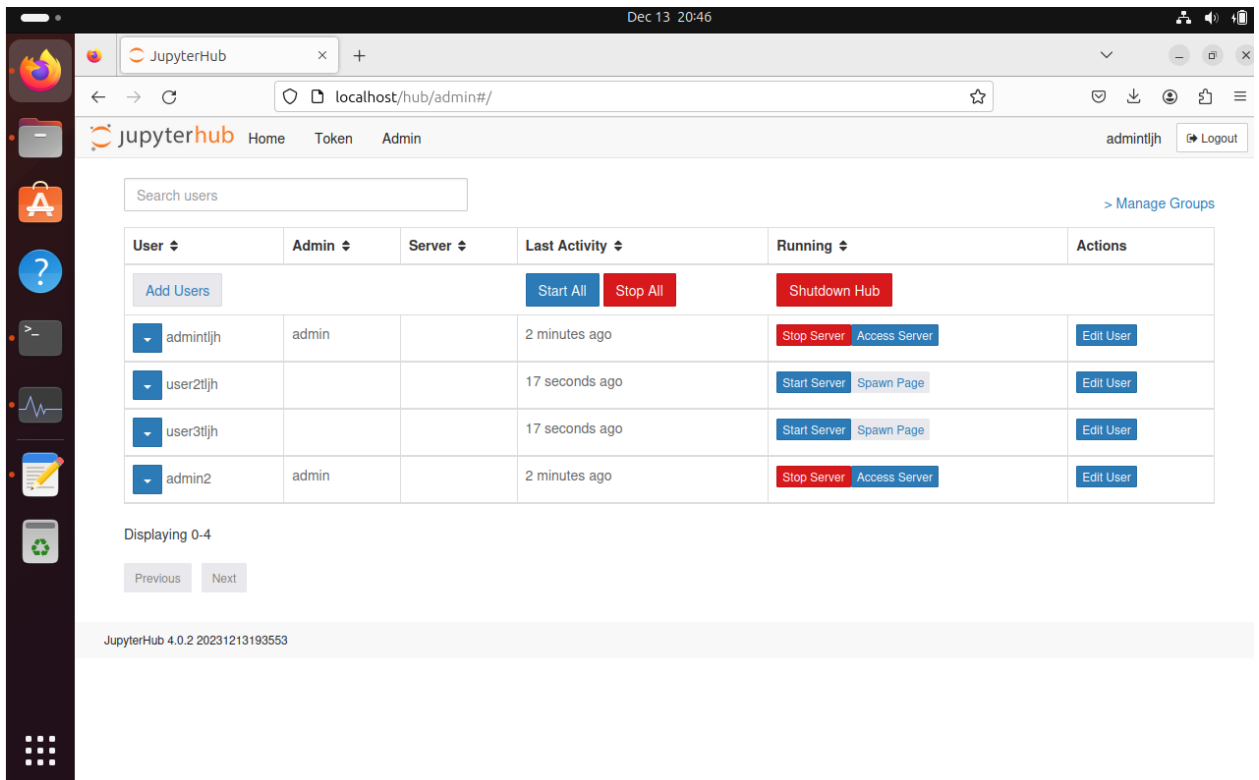


Рисунок 6 – Сеансы после срабатывания скрипта

На рисунке 6 видно, что сеансы администраторов не были закрыты по истечению 60 секунд, а сеансы пользователей были закрыты.