

Отчет по практической работе №2

Студента Свичко Д.Е.

Группы 090303-ПИБ.о21

Основы работы с технологиями контейнеризации и ботами

Telegram

Цель работы.

Создание сервера с постоянно работающим Telegram ботом

Описание работы

В начале работы была предпринята успешная попытка подключения к студенческому серверу, а затем и к рабочему серверу на котором и будет находиться бот.



```
student@ruvds-74upo:~/2103030100$ cd ..
student@ruvds-74upo:~$ ssh student@10.8.0.2
student@10.8.0.2's password:
Linux debian 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar  5 15:54:49 2024 from 10.8.0.1
```

Рис 2.1

Далее необходимо было создать свой каталог с номером зачетной книжки, но так как это была не первая попытка запуска бота, я зашел в уже заранее созданный каталог.

С помощью команды `$ pip install telepot` установил необходимые пакеты данных.

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar  5 15:54:49 2024 from 10.8.0.1
student@debian:~$ cd 210303010
student@debian:~/210303010$ source env/bin/activate
(env) student@debian:~/210303010$ pip install telepot
Collecting telepot
  Using cached telepot-12.7-py3-none-any.whl
Collecting aiohttp>=3.0.0
  Using cached aiohttp-3.9.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
Collecting urllib3>=1.9.1
  Using cached urllib3-2.2.1-py3-none-any.whl (121 kB)
Collecting aiosignal>=1.1.2
  Using cached aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Collecting attrs>=17.3.0
  Using cached attrs-23.2.0-py3-none-any.whl (60 kB)
Collecting frozenlist>=1.1.1
  Using cached frozenlist-1.4.1-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2014_x86_64.whl (175 kB)
Collecting multidict<7.0,>=4.5
  Using cached multidict-6.0.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (114 kB)
Collecting yarl<2.0,>=1.0
  Using cached yarl-1.9.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (291 kB)
Collecting idna>=2.0
  Using cached idna-3.6-py3-none-any.whl (61 kB)
Installing collected packages: urllib3, multidict, idna, frozenlist, attrs, yarl, aiohttp, aiosignal
Successfully installed aiohttp-3.9.3 aiosignal-1.3.1 attrs-23.2.0 frozenlist-1.4.1 idna-3.6 multidict-6.0.5 urllib3-2.2.1 yarl-1.9.4
```

Рис 2.2

Следующим шагом создал файл bot.py в который был вписан основной код бота. В самом коде пришлось настроить отступы, так как отступы при котировании не давали боту работать как надо.

В код программы был добавлен заранее созданный токен для связи с телеграмм ботом.

После чего бот был запущен и все 2 команды работали как надо.

```

(env) student@debian:~/210303010$ python3 bot.py
I am listening ...
Got command: /command1
From : 951188093
Got command: /command2
From : 951188093
^Z
[1]+  Stopped                  python3 bot.py
(env) student@debian:~/210303010$ nano requirements.txt
(env) student@debian:~/210303010$ nano Dockerfile
(env) student@debian:~/210303010$ docker build -t 210303010 .
[+] Building 2.0s (13/13) FINISHED

```

Рис 2.3

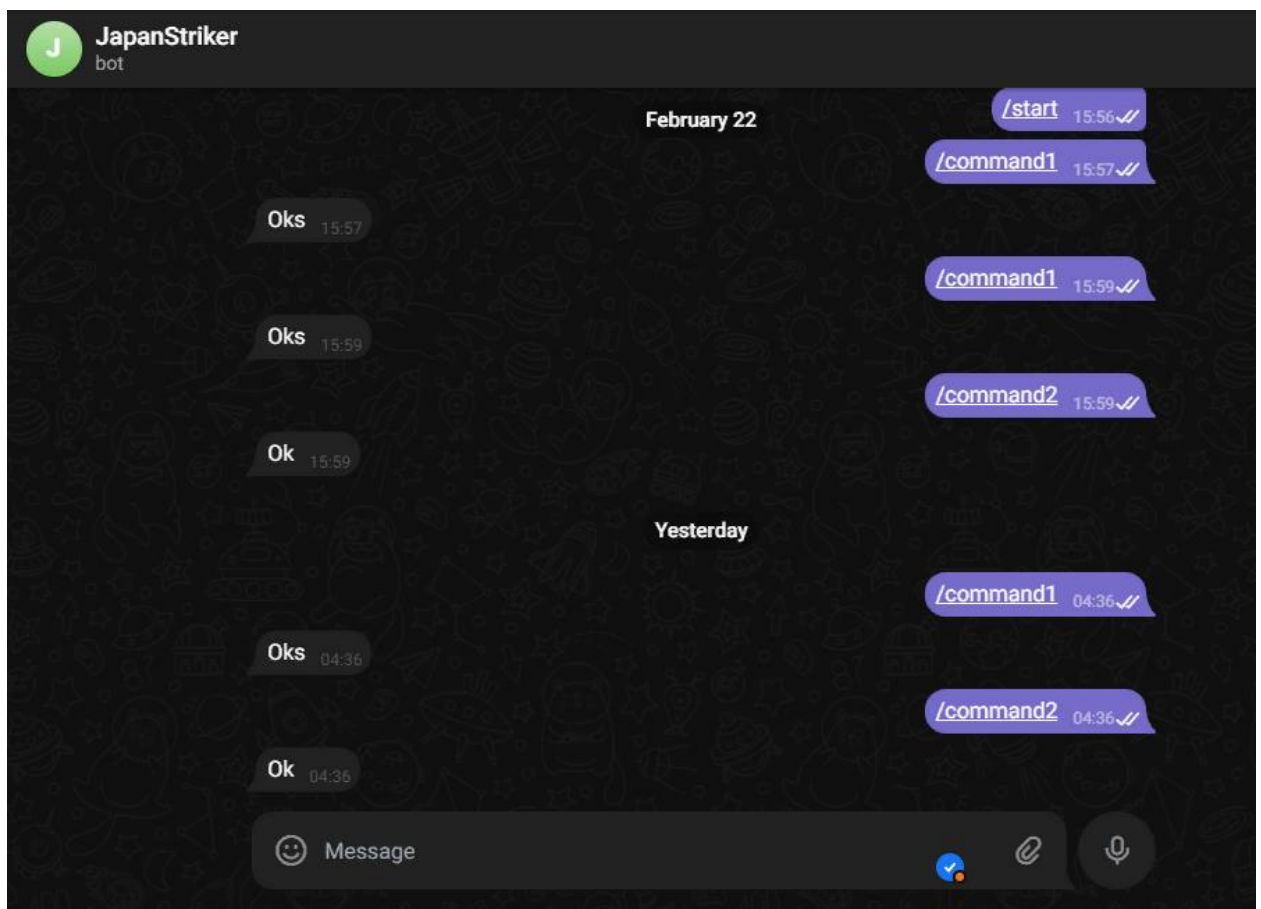


Рис 2.4

Теперь было необходимо подготовить всё нужное для контейнеризации нашего бота для непрерывной работы на сервере. Были созданы файлы requirements.txt, в который добавлялась информация о версии пакета telebot, и docker для сборки образа.

```
^Z
[1]+  Stopped                  python3 bot.py
(env) student@debian:~/210303010$ nano requirements.txt
(env) student@debian:~/210303010$ nano Dockerfile
(env) student@debian:~/210303010$ docker build -t 210303010 .
[+] Building 2.0s (13/13) FINISHED
```

Рис 2.5

Собираем образ контейнера с номером зачетки и запускаем его. В качестве успешного запуска показывается CONTAINER ID.

Проверяю работоспособность бота и останавливаем Docker, так как нам необходимо еще сохранить его в виде docker image. Созряняем наш docker файл под названием /docker_image_210303010.tar и пытаемся сохранить его локально на ПК чего у нас успешно не выходит, так как не получается найти путь докер файла на сервере.

```
(env) student@debian:~/210303010$ docker build -t 210303010 .
[+] Building 2.0s (13/13) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 295B
=> [internal] load metadata for docker.io/library/python:3.10-slim
=> [internal] load metadata for docker.io/library/python:3.10
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [builder 1/3] FROM docker.io/library/python:3.10@sha256:b54e76c629a98430ac9c92e4f6bddeb672396a895b44a85022d12ee2f7239144
=> [internal] load build context
=> => transferring context: 528B
=> [stage-1 1/4] FROM docker.io/library/python:3.10-slim@sha256:4bd9a0e5da619b155a214342bb4f4dc774a947fec9061016a123fa4073431c9cb
=> CACHED [stage-1 2/4] WORKDIR /code
=> CACHED [builder 2/3] COPY requirements.txt .
=> CACHED [builder 3/3] RUN pip install --user -r requirements.txt
=> CACHED [stage-1 3/4] COPY --from=builder /root/.local /root/.local
=> [stage-1 4/4] COPY ./bot.py .
=> exporting to image
=> => exporting layers
=> => writing image sha256:e726a2054d5b52f273550983cdc848cbec01050100aaa29eddc8a2f10a0096a8
=> => naming to docker.io/library/210303010
(env) student@debian:~/210303010$ docker run -d --restart=always 210303010
```

```
(env) student@debian:~/210303010$ docker logs 911ddf8db43c383aa51c94b8c8ea2420d16f7d82e18d7ce7ddaa9fdc75be0b52
I am listening ...
(env) student@debian:~/210303010$ docker stop 911ddf8db43c
911ddf8db43c
(env) student@debian:~/210303010$ docker save -o ./docker_image_210303010.tar 210303010
(env) student@debian:~/210303010$ docker save -o ./docker_image_210303010.tar 210303010
(env) student@debian:~/210303010$ ls -l
total 139640
-rw-r--r-- 1 student student 434 Mar 5 20:35 bot.py
-rw-r--r-- 1 student student 256 Mar 5 20:37 Dockerfile
-rw-r--r-- 1 student student 248 Feb 24 08:12 Dockerfile.save
-rw-r--r-- 1 student student 142959616 Mar 5 20:50 docker_image_210303010.tar
drwxr-xr-x 5 student student 4096 Feb 24 07:50 env
drwxr-xr-x 2 student student 4096 Feb 24 08:28 pythonLern
drwxr-xr-x 2 student student 4096 Feb 24 07:52 PythonLern
-rw-r--r-- 1 student student 14 Mar 5 20:37 requirements.txt
(env) student@debian:~/210303010$ scp: stat local "/.docker_image_210303010.tar": No such file or directory
bash: scp: command not found
(env) student@debian:~/210303010$ AC
(env) student@debian:~/210303010$ ls -l /home/student/210303010/
total 139640
-rw-r--r-- 1 student student 434 Mar 5 20:35 bot.py
-rw-r--r-- 1 student student 256 Mar 5 20:37 Dockerfile
-rw-r--r-- 1 student student 248 Feb 24 08:12 Dockerfile.save
-rw-r--r-- 1 student student 142959616 Mar 5 20:50 docker_image_210303010.tar
drwxr-xr-x 5 student student 4096 Feb 24 07:50 env
drwxr-xr-x 2 student student 4096 Feb 24 08:28 pythonLern
drwxr-xr-x 2 student student 4096 Feb 24 07:52 PythonLern
-rw-r--r-- 1 student student 14 Mar 5 20:37 requirements.txt
(env) student@debian:~/210303010$ docker load -i ./docker_image_210303010.tar
Loaded image: 210303010:latest
(env) student@debian:~/210303010$ docker run -d --restart=always 210303010
029c60b85794d7eea2d73674be42c4ad000fc7afa605c3b9c737172e1335f5f9
(env) student@debian:~/210303010$
```

Рис 2.6-7

Вывод

Впервые научился запускать бота не локально, а взаимодействуя с сервером. Узнал, что такое контейнеризация и зачем она вообще нужна. К сожалению, не удалось сохранить архив контейнера локально из-за ошибки, что файл не был найден по нужному пути на сервере.

Выполнил	Свичко Д.Е
Проверил	Шайхутдинов Д.В.