Установка Psiphon (psiphon-docker) на Ubuntu

Прокси-туннель для обхода блокировок на базе psiphon-tunnel-core — swarupsengupta2007/psiphon-docker. Поднимает HTTP- и SOCKS5-прокси с выбором региона выхода.

Каждый блок кода ниже рассчитан на копирование целиком.

> Предполагается, что Docker и Docker Compose plugin уже установлены (см. страницу «Установка Docker и Docker Compose на Ubuntu»).

1. Подготовка каталога

```
sudo mkdir -p /opt/psiphon/config && sudo chown -R 1000:1000 /opt/psiphon && cd /opt/psiphon
```

`/opt/psiphon` выбран как каталог для стороннего софта по FHS — не привязан к домашней директории конкретного пользователя. Каталог `config` монтируется в контейнер и хранит `psiphon.config` — должен быть доступен на запись пользователю с указанными ниже `PUID`/`PGID` (в примере — `1000:1000`, при других значениях `chown` нужно поправить соответственно).

2. docker-compose.yml

```
nano docker-compose.yml
```

Содержимое:

```
services:
  psiphon:
    image: swarupsengupta2007/psiphon:latest
```

```
    container_name: psiphon
    environment:
      - PUID=1000
      - PGID=1000
      - HTTP_PORT=8080
      - SOCKS_PORT=1080
      - DEVICE_REGION=RU
      - EGRESS_REGION=SG
    volumes:
      - ./config:/config
    ports:
      - 127.0.0.1:1080:1080
      - 127.0.0.1:8080:8080
    restart: unless-stopped
```

> **Docker и UFW.** Как отмечено на странице первоначальной настройки сервера, Docker прописывает правила `ports:` напрямую в iptables в обход UFW. Публикация на `127.0.0.1:...` вместо `0.0.0.0` (голого порта) не даёт достучаться до прокси снаружи — открывайте наружу только если это осознанное решение.

Переменные окружения

Применяются **только при первом запуске**, пока в `/config` ещё нет `psiphon.config`:

Переменная	Описание	По умолчанию
`PUID`	UID процесса psiphon	1000
`PGID`	GID процесса psiphon	1000
`HTTP_PORT`	Порт HTTP-прокси	8080
`SOCKS_PORT`	Порт SOCKS-прокси	1080
`DEVICE_REGION`	Регион устройства	RU
`EGRESS_REGION`	Регион выхода трафика	SG

`PUID`/`PGID` текущего пользователя:

```
id -u && id -g
```

Какой EGRESS_REGION выбрать из RU

Psiphon не публикует официальный актуальный список стран, где у него сейчас подняты серверы выхода — набор доступных `EGRESS_REGION` **динамический** и различается от инстанса к инстансу. Проверить фактически доступный список для конкретного контейнера — по его логам, строка `AvailableEgressRegions`:

```
docker logs psiphon-lt | grep AvailableEgressRegions
```

На практике встречался, например, такой список: `AT, AU, BE, BR, CA, CH, DE, DK, ES, FR, GB, IE, IN, IT, JP, LT, NL, NO, PL, RS, SE, SG, US` — обратите внимание, что скандинавских/прибалтийских кодов вроде `FI`, `EE`, `LV` там могло не быть вовсе, хотя по географии они ближе всего к RU. Если в `EGRESS_REGION` указать код, которого нет в списке доступных, клиент не сможет установить туннель (в логе не будет `ConnectedServerRegion`).

Кандидаты, отсортированные по географической близости к RU, которые стоит пробовать по очереди — но перед использованием сверяйтесь с фактическим списком выше:

Приоритет	Код	Регион
1	`LT`	Литва
2	`PL`	Польша
3	`SE`	Швеция
4	`DE`	Германия
5	`NO`	Норвегия
6	`DK`	Дания
7	`NL`	Нидерланды

Если ни один код не подходит (сервер в этом регионе недоступен) или нужен более общий обход — оставьте `EGRESS_REGION=""` (пустая строка), тогда клиент сам выберет наименее нагруженный доступный сервер, без привязки к региону.

Замерить фактическую задержку для конкретного региона после запуска контейнера:

```
curl -x socks5h://127.0.0.1:1080 -o /dev/null -s -w 'time_total: %{time_total}s\n'
https://ifconfig.me
```

Меняя `EGRESS_REGION` и пересоздавая контейнер (`docker compose up -d --force-recreate`), можно сравнить время ответа по разным кандидатам и оставить самый быстрый из реально работающих.

3. Запуск

```
docker compose up -d
```

4. Проверка

Статус контейнера и логи:

```
docker compose ps && docker logs -f psiphon
```

Healthcheck:

```
docker inspect --format='{{json .State.Health}}' psiphon
```

Ожидаемый статус — `"Status":"healthy"`.

Проверка прокси:

```
curl -x socks5h://127.0.0.1:1080 https://ifconfig.me
curl -x http://127.0.0.1:8080 https://ifconfig.me
```

Вернувшийся IP должен соответствовать выбранному `EGRESS_REGION`.

5. Управление сервисом

```
docker compose restart   # перезапуск
docker compose down      # остановка и удаление контейнера
docker compose pull && docker compose up -d   # обновление образа
```

6. Несколько регионов одновременно

`EGRESS_REGION` — это одно значение на один запущенный контейнер. Чтобы иметь под рукой сразу несколько регионов выхода — поднимают отдельный контейнер на каждый, с уникальными `container_name`, каталогом `config` и внешними портами:

```
services:
  psiphon-lt:
    image: swarupsengupta2007/psiphon:latest
    container_name: psiphon-lt
    environment:
      - PUID=1000
      - PGID=1000
      - HTTP_PORT=8080
      - SOCKS_PORT=1080
      - DEVICE_REGION=RU
      - EGRESS_REGION=LT
    volumes:
      - ./config-lt:/config
    ports:
      - 127.0.0.1:1080:1080
      - 127.0.0.1:8080:8080
    restart: unless-stopped

  psiphon-de:
    image: swarupsengupta2007/psiphon:latest
    container_name: psiphon-de
    environment:
      - PUID=1000
      - PGID=1000
      - HTTP_PORT=8080
      - SOCKS_PORT=1080
      - DEVICE_REGION=RU
      - EGRESS_REGION=DE
    volumes:
      - ./config-de:/config
    ports:
      - 127.0.0.1:1081:1080
      - 127.0.0.1:8081:8080
    restart: unless-stopped

  psiphon-nl:
```

```
    image: swarupsengupta2007/psiphon:latest
    container_name: psiphon-nl
    environment:
      - PUID=1000
      - PGID=1000
      - HTTP_PORT=8080
      - SOCKS_PORT=1080
      - DEVICE_REGION=RU
      - EGRESS_REGION=NL
    volumes:
      - ./config-nl:/config
    ports:
      - 127.0.0.1:1082:1080
      - 127.0.0.1:8082:8080
    restart: unless-stopped
```

Что важно не перепутать:

- **Разные** `container_name` — иначе Docker откажется поднимать второй сервис.
- **Разные каталоги в** `volumes` (`config-lt`, `config-de` и т. д.) — каждый регион хранит свой `psiphon.config`, смешивать их в одной папке нельзя.
- **Разные внешние порты** — внутри контейнера порты одинаковые (1080/8080), наружу пробрасываются на разные (1080/8080, 1081/8081, ...), иначе конфликт портов на хосте.
- Каталоги под каждый регион создаются заранее, как в шаге 1: `sudo mkdir -p /opt/psiphon/config-lt /opt/psiphon/config-de /opt/psiphon/config-nl && sudo chown -R 1000:1000 /opt/psiphon`.

Дальше в приложении просто выбираете локальный порт, соответствующий нужному региону выхода.

Что стоит сделать дополнительно

- Не менять переменные окружения после первого запуска без ручной правки/удаления `config/psiphon.config` — на уже созданный конфиг они не влияют.
- Если прокси должен быть доступен с других машин — не публиковать порт на `0.0.0.0` без дополнительного контроля доступа (файрвол/аутентификация на уровне прокси).
- Настроить бэкап каталога `config`.

Revision #8
Created 2026-09-07 23:52:01 UTC by mrleo1nid
Updated 2026-09-08 00:14:50 UTC by mrleo1nid