

Watchtower: автоматически обновляем докер-контейнеры

Сайт: <https://github.com/containrrr/watchtower>

Документация: <https://containrrr.dev/watchtower/>

Настройки

Watchtower наблюдает за запущенными docker-контейнерами, периодически проверяет обновления версий образов (по умолчанию раз в сутки). Если вышла новая версия контейнера - скачивает ее и перезапускает контейнер. Причем все это можно гибко настраивать под ваши хотелки.

Установка:

```
docker run -d \
  --name watchtower \
  --restart=always \
  -v /var/run/docker.sock:/var/run/docker.sock \
  containrrr/watchtower
```

Однако далеко не всегда требуется жить на последней версии (те самые любимые всеми latest). Поэтому можно настроить исключения, необходимо добавить метку (label): <https://containrrr.dev/watchtower/container-selection/>

```
LABEL com.centurylinklabs.watchtower.enable="false"
```

Или сразу при запуске:

```
docker run -d --label=com.centurylinklabs.watchtower.enable=false someimage
```

При обновлении контейнеров остаются старые версии. Можно, конечно, запускать в кроне что-то вроде

```
docker image prune -f
```

но гораздо удобнее указывать флаг

```
--cleanup
```

Так же зачастую нам требуется однозначно указать время обновления (например ночью, когда нагрузка минимальна), это можно сделать флагом

```
--schedule "0 0 4 * * *"
```

в привычном формате cron'a, но состоящем из 6 полей вместо 5 привычных (добавлены секунды), подробнее тут:

https://pkg.go.dev/github.com/robfig/cron@v1.2.0#hdr-CRON_Expression_Format

Еще очень удобен флаг

```
--rolling-restart
```

для перезапуска контейнеров по очереди, а не сразу всех.

Так же можно работать с другим Docker-хостом:

```
--host, -H
```

, указав адрес в виде

```
"tcp://hostname:port"
```

Еще много полезного можно посмотреть на страничке с документацией (ссылка в самом начале заметки).

Мониторинг

За всеми такими действиями нужен глаз да глаз, чтобы если что-то пошло не так можно было быстрее понять причину. Для этого можно включить сбор метрик при помощи Prometheus:

<https://containrrr.dev/watchtower/metrics/>

Чтобы заработало, нужно:

- Задать API token:

```
--http-api-token YOU_TOKEN
```

- Включить API endpoint: `</code>-http-api-metrics</code>`

Уведомления

Для понимания что происходит при работе - можно использовать уведомления, подробно расписано тут: <https://containrrr.dev/watchtower/notifications/>

Уведомления в Telegram: создаем нашего бота в Telegram и получаем его токен, узнаем id нашего чата и добавляем переменные среды (env) как указано ниже:

```
docker run -d \  
  --name watchtower \  
  --restart=always \  
  -e WATCHTOWER_LIFECYCLE_HOOKS=1 \  
  --
```

```
-e WATCHTOWER_NOTIFICATIONS=shoutrrr \
-e
WATCHTOWER_NOTIFICATION_URL=telegram://TOKEN_ВАШЕГО_БОТА@telegram/?channels
=ID_ВАШЕГО_ЧАТА \
-v /var/run/docker.sock:/var/run/docker.sock \
containrrr/watchtower
```

Шаблон для сервиса

```
version: '3'

services:
  app-monitored-by-watchtower:
    image: myapps/monitored-by-watchtower
    labels:
      - "com.centurylinklabs.watchtower.enable=true"

  watchtower:
    image: containrrr/watchtower
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    command: --cleanup --schedule "00 50 23 * * *" --http-api-update
    environment:
      - WATCHTOWER_HTTP_API_TOKEN=mytoken
    labels:
      - "com.centurylinklabs.watchtower.enable=false"
    ports:
      - 8080:8080
```

[docker](#), [image](#), [watchtower](#), [обновление](#)

From:
<https://wiki.rtzra.ru/> - **RTzRa's hive**

Permanent link:
<https://wiki.rtzra.ru/devops/watchtower/watchtower-setup?rev=1705173419>

Last update: **2024/01/13 22:16**

