Postgresql и Docker

JUNO

PGConf .Russia 2019

Лев Драгунов

Зачем

- Облегчить создание среды
- Разрулить зависимости
- Унифицировать инфраструктуру

Зачем в Juno

- Версионирование данных
- Способ доставки данных в прод
- Версионирование логики

Стандартный образ

- https://hub.docker.com/_/postgres/
- https://github.com/docker-library/postgres

- Инициализирует базу в $PGDATA
- Создаёт пользователя с паролем
- Выполняет SQL файлы из директории

Запуск контейнера

```
docker run --name pg \
    -v /mnt/data:/var/lib/postgresql/data \
    -v /mnt/code/pgscripts:/docker-entrypoint-initdb.d \
    -e POSTGRES_PASSWORD=pgpass \
    -d
    postgres:latest
```

Когда ничего нет

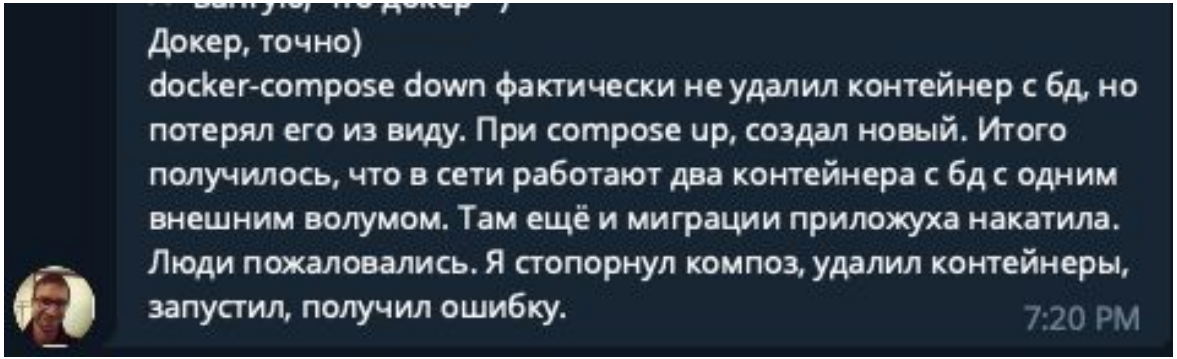
```
docker exec -it pg psql -U postgres
```

Pros and cons

- Стандартный контейнер
- Гибко

- Долгий старт
- Всё зависит от файловой системы

Осторожно! Грабли!



https://t.me/pgsql

Альтернативы

- RDS и аналоги
- Standalone
- Ansible/Chief

Свой образ

Свой образ

- Быстрый старт
- Как можно меньший размер образа
- Версионирование кода

Dockerfile

```
# PG is one layer. Data is another one.
ARG HUB=internal-dockerhub.junolab.net:5000
ARG PG_TAG=latest

# Building stage
FROM ${HUB}/postgis-bundle:${PG_TAG} as builder
COPY ./input/* /input/
COPY run.sh /

RUN /bin/bash /run.sh

# Production stage
FROM ${HUB}/postgis-bundle:${PG_TAG}
COPY --from=builder --chown=postgres /pgdata/ /pgdata/
```

run.sh

```
gosu postgres pg_ctl -w start

psql -v ON_ERROR_STOP=1 -d "ms_db" -f  \
                                        /data/some_script.sql

psql -v ON_ERROR_STOP=1 -d "ms_db" -c  \
"create table data_version as select now() as version;"

sed -i -e "s/^#statement_timeout =.*$/statement_timeout = \
                                    1000/" $PGDATA/postgresql.conf

gosu postgres vacuumdb --full --freeze --analyze --all

gosu postgres pg_ctl -w stop
```

Итоги

- Postgres работает в докере
- Стандартный образ для тестов
- Вывод PGDATA в volume опасен

Вопросы?

Лев Драгунов
Lev@Dragunov.pro