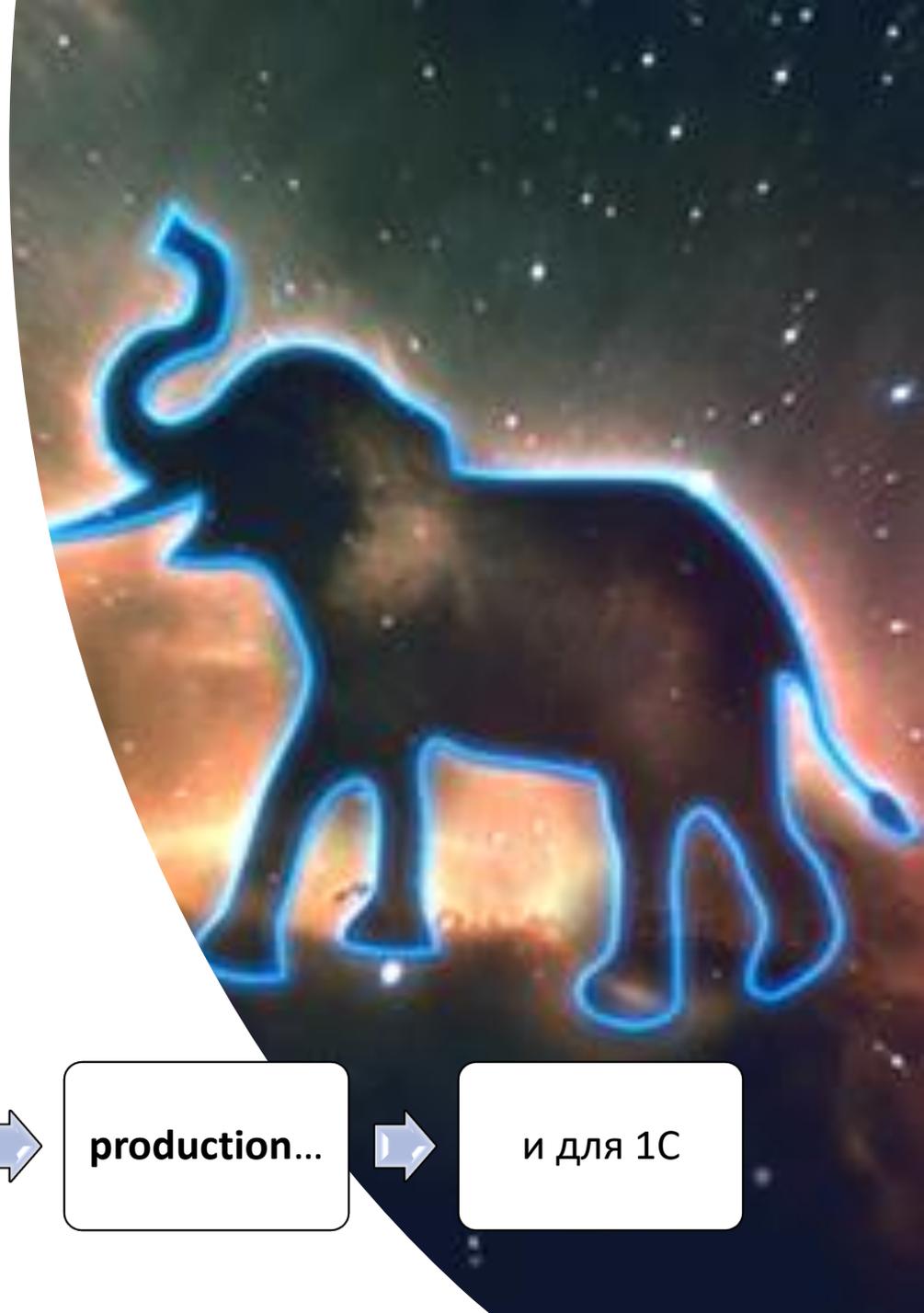
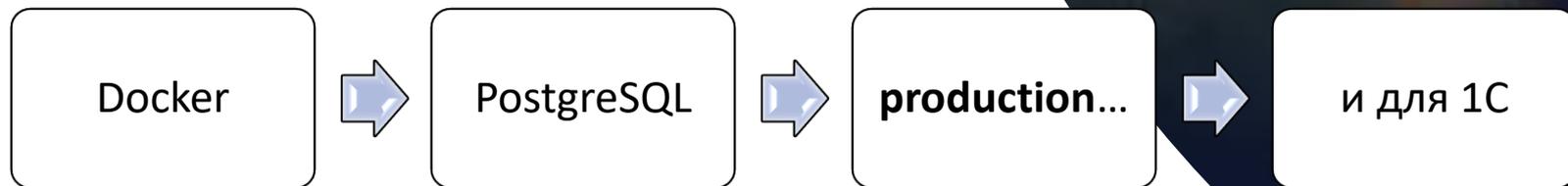


PGConf.Russia 2018

Снижаем сроки
«протечек абстракций»



PGConf.Russia 2018

<https://github.com/allustin>

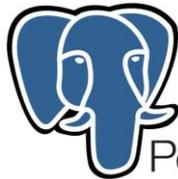


docker

sonarqube



GitLab



PostgreSQL



Jenkins

RabbitMQ



kafka

etc.

1C+<4ToTo>



Alexey Lustin

allustin

Tech Conflicts Resolver

OpenSource & TCO

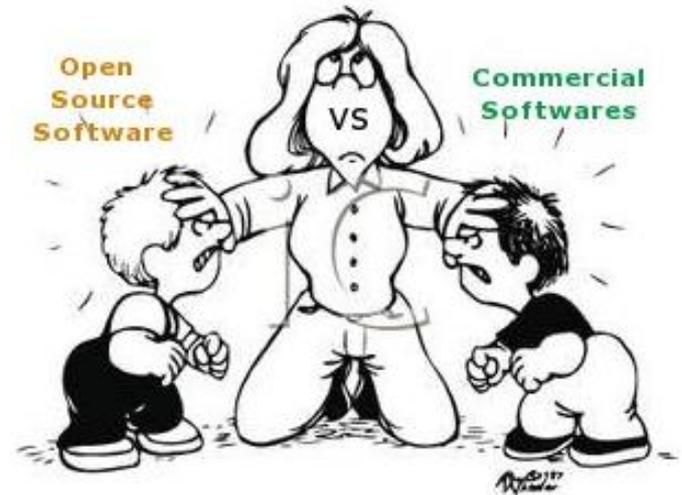
MSSQL

МОЖНО

ПОСТАВИТЬ И

ПОЛГОДА НЕ

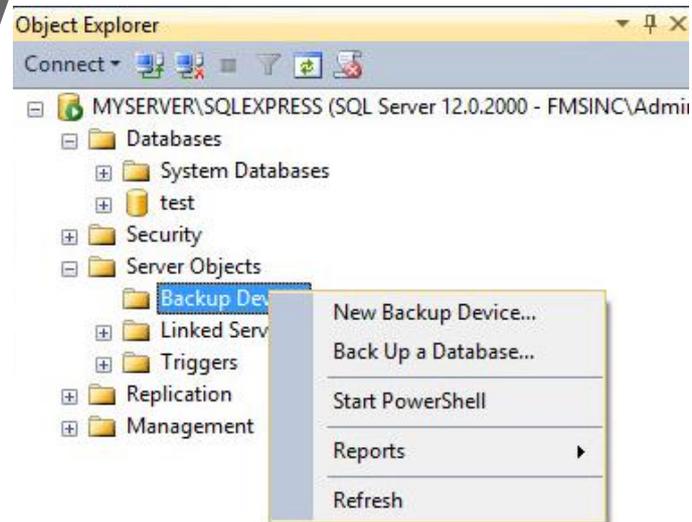
ТРОГАТЬ



PGConf.Russia 2018

PostgreSQL & Backups

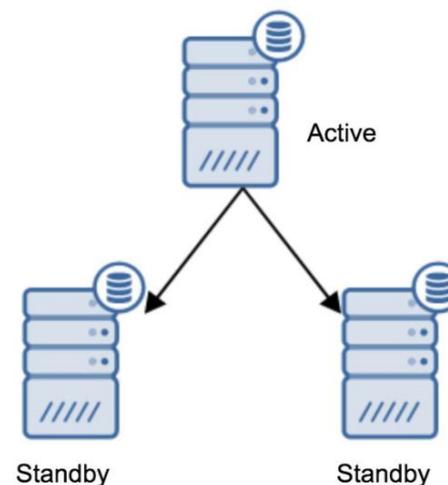
Как можно
настроить
бэкапирование
как в
коммерческих
СУБД



```
$ pg_dump -s prod1_db > prod1_db.ddl_only.sql
```

Dev & Prod

Как
разрабатывать
с учетом
продуктивной
нагрузки





sameersbn / docker-postgresql

Code Issues 27 Pull requests 11

Dockerfile to build a PostgreSQL container image

paunin / PostDock

Code Issues 11 Pull requests 1

PostDock - Postgres & Docker -

allustin план релизов

6 contributors

77 lines (61 sloc) | 3.34 KB

```
1 FROM sameersbn/ubuntu:14.04.20170123
2
3 LABEL maintainer.base="sameer@damagehead.com" \
4     maintainer.current="team@silverbulleters.org"
5
6 ENV DEBIAN_FRONTEND=noninteractive
7
8 ENV PG_APP_HOME="/etc/docker-postgresql"\
9     PG_VERSION=9.6 \
```

Builds » VanessaDockers » ya-docker-postgresql-1C » master » 38

 SUCCESS

Finished:

2 months ago (06:42)

Previous:

36

Parallelism:

1x out of 4x

Queued:

00:00 waiting + 00:00 in queue

Концептуальное решение

DBA (Database архитекторы)

- Пишут Dockerfile

DevOps

- Собирают образа и архитектуру через Docker-Compose

Dev

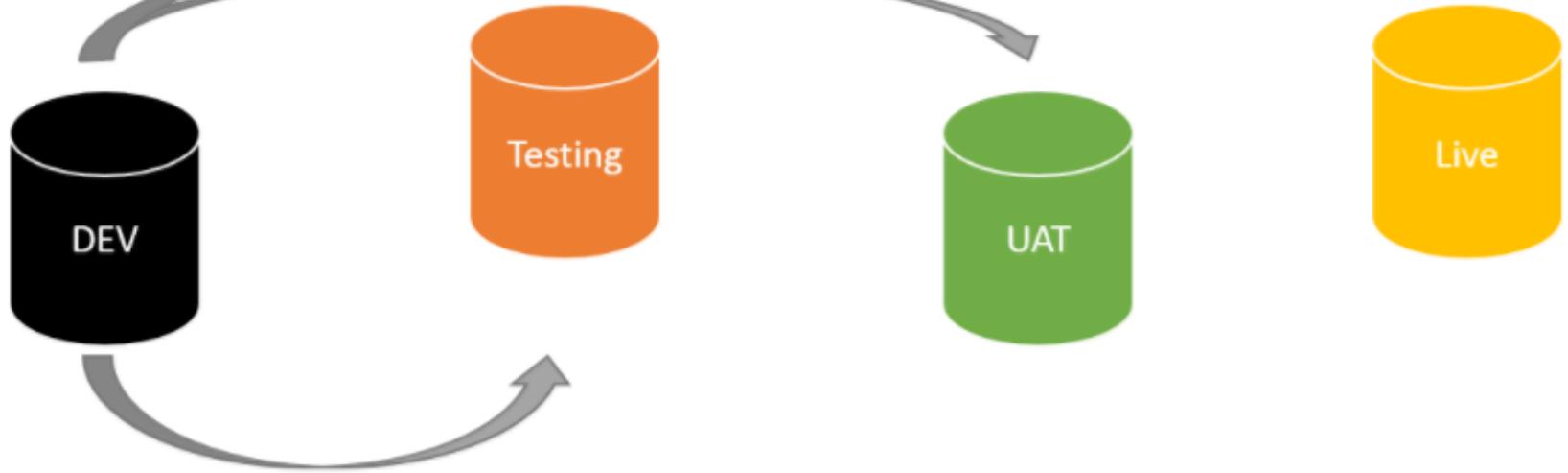
- Используют образа при разработке и тестировании

Ops (Администраторы инфраструктуры)

- Используют при развертывании на продуктиве

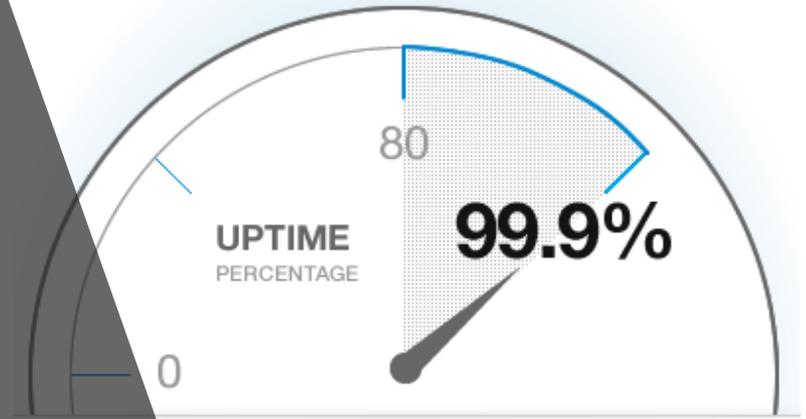
PGConf.Russia 2018

```
PORTS                                NAMES
443/tcp, 0.0.0.0:8880->80/tcp        yadockerpostgresql1c_PgAdminDBA_1
0.0.0.0:8899->8888/tcp                yadockerpostgresql1c_PowaWeb_1
5432/tcp                              yadockerpostgresql1c_PostgreSQLSlave_1
0.0.0.0:10000->8080/tcp              yadockerpostgresql1c_pghero-db1_1
0.0.0.0:10050->10050/tcp             mamonsu
0.0.0.0:5432->5432/tcp                yadockerpostgresql1c_PostgreSQL_1
```



```
ya-docker-postgresql-1C>docker-compose up -d
```

И как это
работает



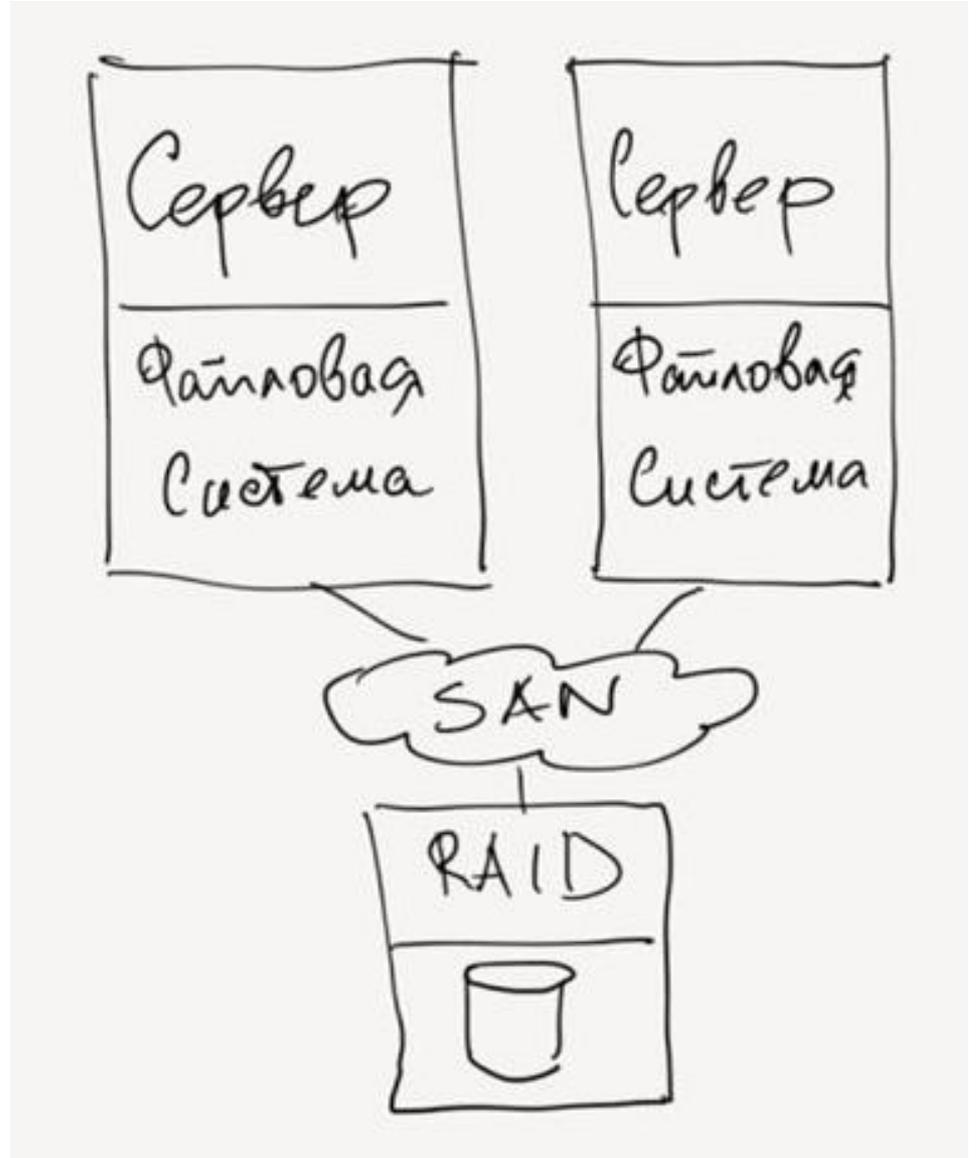
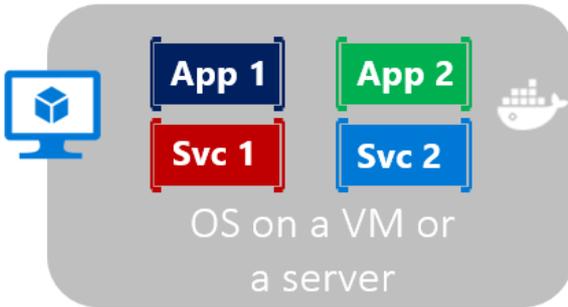
Файловое хранилище

магии не будет

```
ya-docker-postgresql-1C>docker volume ls
```

DRIVER	VOLUME NAME
• local	yadockerpostgresql1c_pg-data-master
• local	yadockerpostgresql1c_pg-data-slave
• local	yadockerpostgresql1c_pg-log-master
• local	yadockerpostgresql1c_pg-log-slave
• local	yadockerpostgresql1c_pg-v81cdata-master
• local	yadockerpostgresql1c_pg-v81cdata-slave
• local	yadockerpostgresql1c_pg-v81cindex-master
• local	yadockerpostgresql1c_pg-v81cindex-slave
• local	yadockerpostgresql1c_pgadmin-data

Docker Host

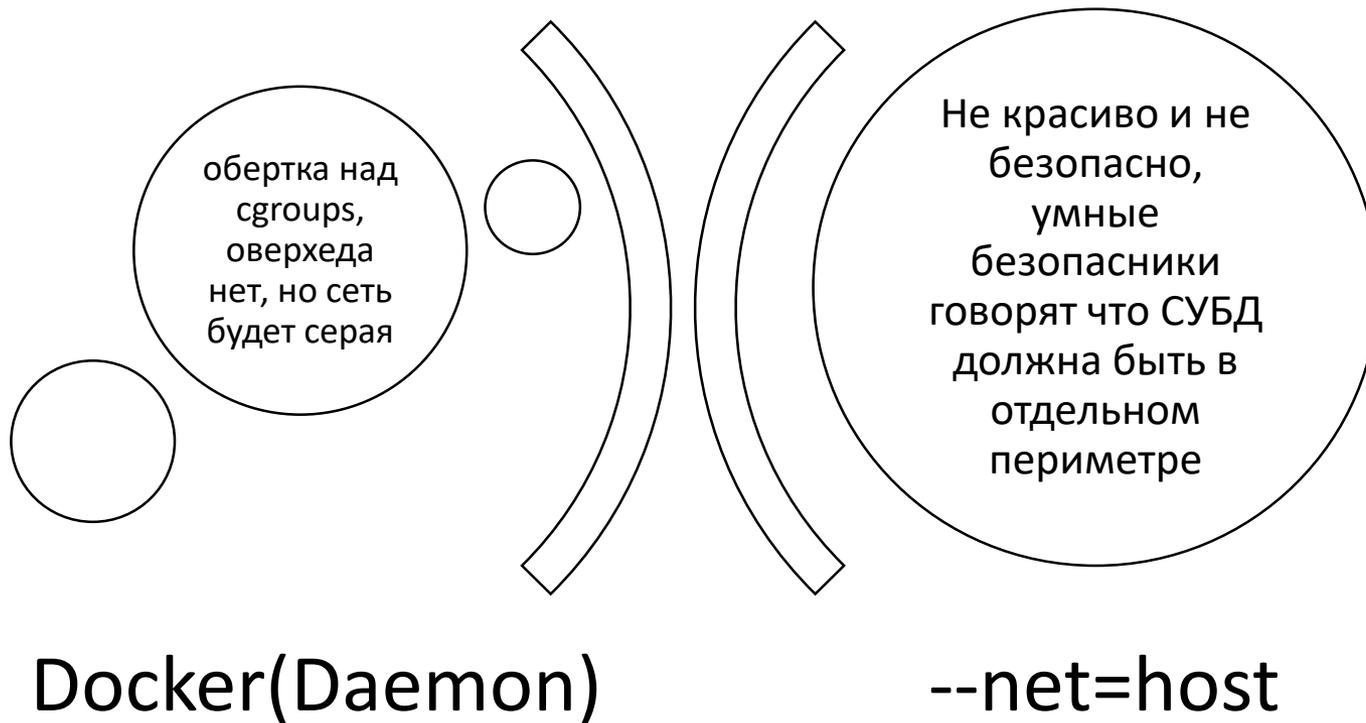


PGConf.Russia 2018

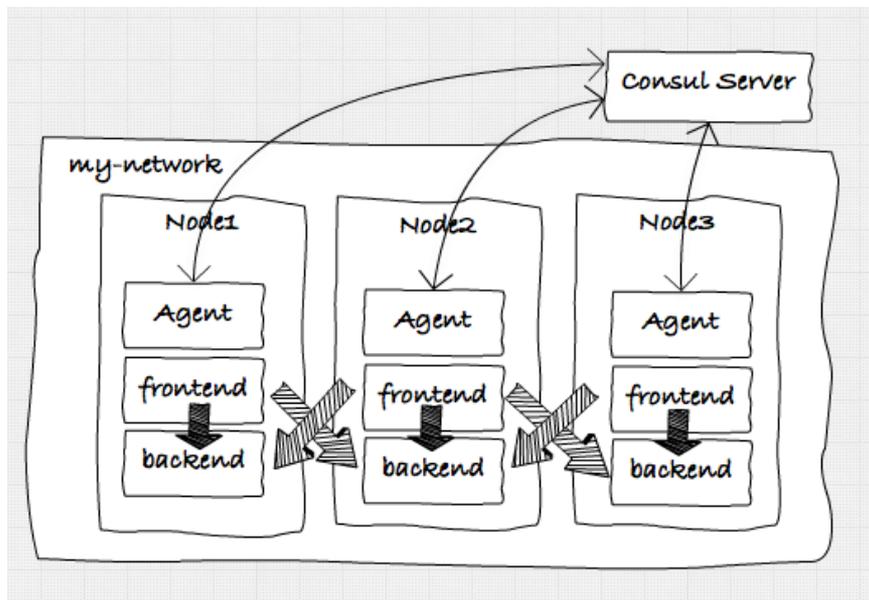
Сетевые особенности



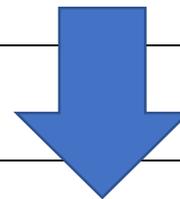
5432 ли порт ?



Если не --net



pg-db-sonardb.service:5432



docker-postgresql-masters:15432



docker_sonardb:5432

А теперь что с postgresql.conf

Контейнер же может быть удален за секунду

s/ya-docker-postgresql-1C/blob/master/runtime/functions

```
527
528  enable_magic() {
529      if [[ ${PG_TUNE_BY_MAMONSU} == true ]]; then
530          echo "MamonSu try to tune PG"
531          exec_as_postgres ${PG_BINDIR}/pg_ctl -D ${PG_DATADIR} -w start >/dev/null
532          mamonsu tune
```

Еще немного магии

`set_max_connection`

`set_timezone`

`adaptive_shared_buffers`

`adaptive_temp_buffers`

`adaptive_autovacuum_workers`

`adaptive_work_mem`

`adaptive_maintenance_mem`

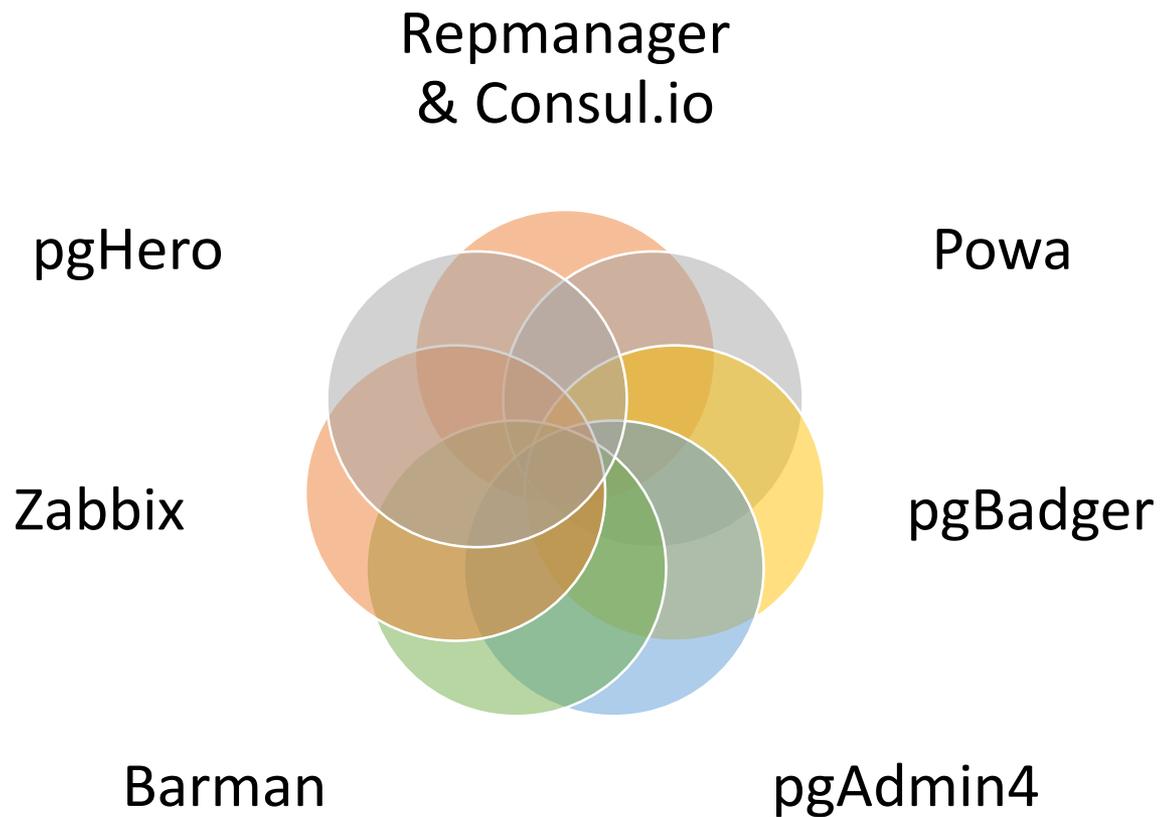
`adaptive_cache_size`

`adaptive_io`

`adaptive_sync`

`enable_magic`

«Суп из топора»



Поведение баз данных

SonarQube – 948 проектов анализа (JDBC)

Branch: master ▾

[docker-sonarqube / 7.0 / run.sh](#)

GitLab – 1290+ репозиториев
(ActiveRecord Ruby)

[sameersbn/gitlab:10.4.2-1](#)

1C – 500+ пользователей документооборота
(libpg.h)

[onecdocker](#)

Образ машины счастливого 1C-ника

Для production

- 1С – постоянный rerack (помимо остального)
pg_repack -- Reorganize tables in PostgreSQL databases with minimal locks

- SonarQube

Модуль GEQO (Genetic Query Optimization, Генетическая оптимизация запросов)

- GitLab – обязателен PostGIS

```
+DB_POSTGIS=${DB_POSTGIS:-false}  
+DB_POSTGIS_HSTORE=${DB_POSTGIS_HSTORE:-true}  
+DB_POSTGIS_TOPOLOGY=${DB_POSTGIS_TOPOLOGY:-true}
```

Хотите попробовать

- `git clone`
<https://github.com/VanessaDockers/ya-docker-postgresql-1c>
- `docker-compose up -d`
- `code-insiders` .

 `docker-compose.yml` ×

```
You, a few seconds ago | 3 authors (Бессонов Евгений and others)
1  version: '2'           Alexey Lustin, 3 months ago • First workin
2  services:
3     PostgreSQL:
4         restart: always
5         build:
6             context: .
7         image: silverbulleters/ya-docker-postgresql-1c:latest
8         # tmpfs:
9         #   - /var/lib/postgresql/temptblspc
10        ports:
11            - "5432:5432"
12        environment:
13            - USETMPFS_FOR_TMP=true
14            - DEBUG=true
```

Минусы

Когда все таки придется дорабатывать образ – напрямую в продуктиве это сделать получится, но ваши изменения внутри контейнера могут не сохранится. Выход – кодить образ через Dockerfile

- Придется изучить docs.docker.com

PGConf.Russia 2018

Немного евангелизма DevOps

OpenSource СУБД работает не только у бородатых дядек

- Работает быстро и лучше чем MSSQL

Enterprise он в голове, «большая» база

- это вопрос к ней отношения, а не размер

Платные СУБД вызывают деградацию компетенции у Вас, за которую вы еще отдаете деньги.

- OpenSource «не бесплатен», но это вложение в собственные компетенции (в компетенции команды)

Спасибо за внимание !!!

Roadmap

- Запуск в **Kubernetes**
- Миграция на 10-ку (ждем официального образа от PostgreSQL.Pro)
- Удобный «шардинг»

