PG BootCamp Russia 2025 Ekaterinburg

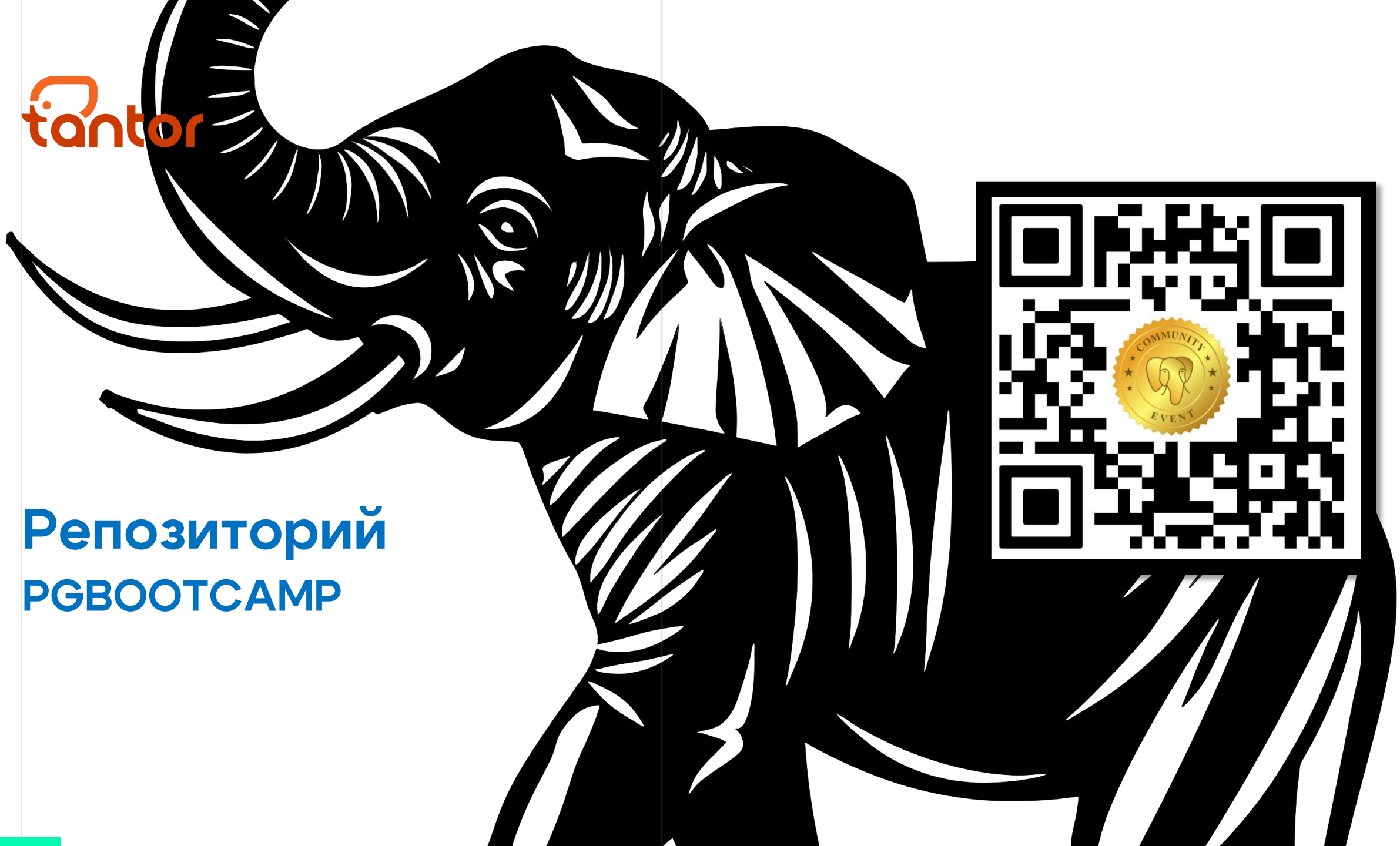
PGBootCamp.ru

pg_orchestrator: новый взгляд на тестирование PostgreSQL

Лев Николаев

разработчик, «Тантор Лабс»

tantor



Репозиторий

PGBOOTCAMP

Обо мне

Образование:

- Аспирант «МАДИ»

Опыт работы в проектах:

- «МАДИ» TKG, 3D-Modelling
- Tantor Labs

Начнём с вопроса

Кто из вас сталкивался
с **обновлением**
баз данных или **тестированием**
времени выполнения запросов?

Сложности сопровождения PostgreSQL в продакшене

Сложность миграций и обновлений

- Риски простоев при выполнении pg_upgrade и minor-обновлений
- Ошибки, связанные с ручной настройкой и миграцией расширений

Риски резервного копирования и восстановления

- Ошибки при создании дампов и восстановлении данных
- Высокая зависимость от ручных процедур

Трудности анализа производительности запросов

- Сравнение EXPLAIN-планов, замер времени выполнения
- Сложности в обнаружении регрессий и неэффективных запросов

Проблемы с изолированными тестовыми окружениями

- Трудности создания стабильных и повторяемых условий для тестирования
- Влияние внешних факторов на результаты тестов

Кто сталкивается с этими проблемами?

1

Обновления и миграции

- DBA
- DevOps

2

Анализ производительности

- Performance инженеры
- Разработчики

3

Резервное копирование и тестовые окружения

- DBA
- DevOps

Что такое PostgreSQL Orchestrator?

Ключевые возможности

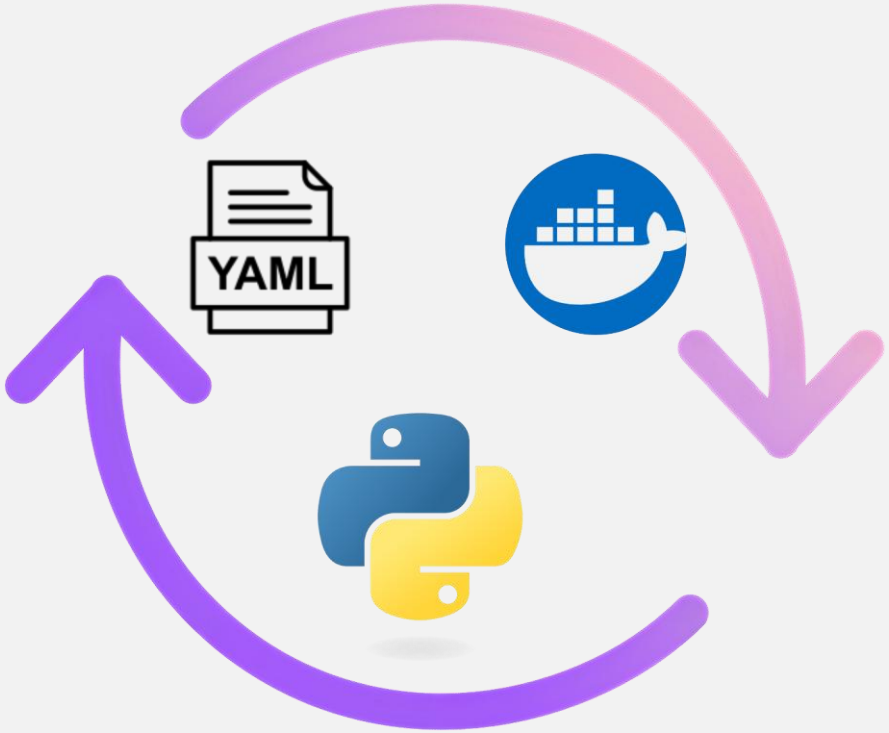
Проверка обновлений
в безопасном тестовом контуре ->

- Обнаружение ошибок и проблем в процессе миграции до релиза.

Сравнение производительности ->

- Замер времени выполнения и анализ EXPLAIN-планов для оценки изменений между версиями СУБД.

Технологическая база проекта



Основные языки:

- Python
- YAML

Контейнеризация:

- Docker для изолированных тестовых окружений

Основные библиотеки и инструменты:

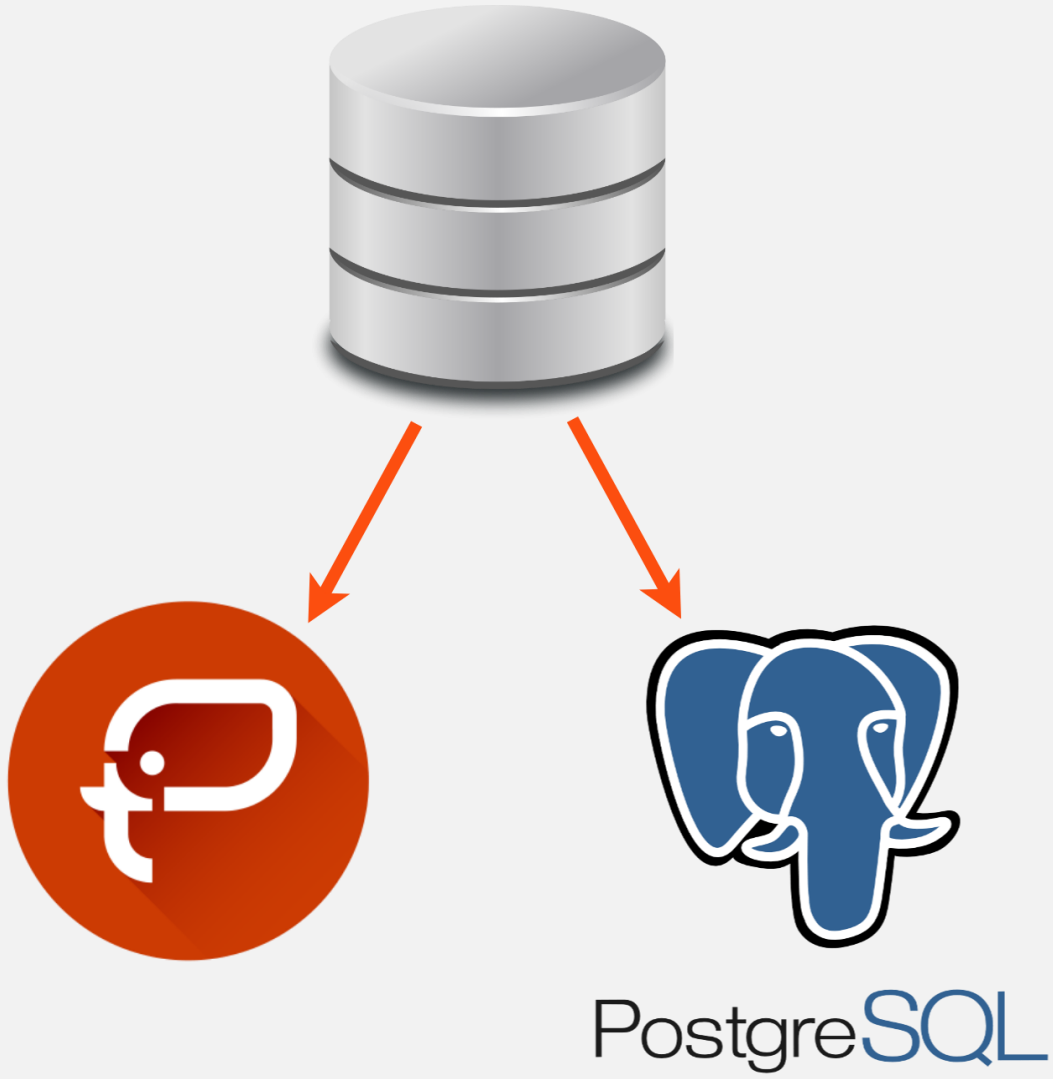
- PyYAML – для работы с YAML
- Pydantic – для валидации конфигураций
- docker-py – для управления Docker-контейнерами
- asyncpg – для асинхронного взаимодействия с PostgreSQL

Поддерживаемые типы СУБД

PostgreSQL (pgdg)*

СУБД Tantor

* PostgreSQL от Global Development Group



Docker конфигурация

- Используйте образы из официального Docker Hub
- Указывайте свой репозиторий и имя образа для специализированных сборок
- Образ автоматически загружается и контейнер запускается с заданными параметрами

DockerConfig
`image = "ubuntu:22.04"`
`registry = "registry.***.com"`
`host_port = 5430`
`container_port = 5432`
`container_name = "pg_orchestrator_container"`

Возможности модулей

Модуль Upgrade:

- Установка СУБД из пакета для гибкой проверки сборок
- Настройка миграционных сценариев через pre/post-скрипты (хуки)
- Гибкая конфигурация стратегий миграции с передачей дополнительных аргументов

Модуль Perf:

- Сравнение EXPLAIN-планов с несколькими эталонами
- Замер времени запросов с корректировкой через performance_coefficient
- Возможность установки СУБД из пакета для тестирования новых сборок

Use Case – Проблемы и Роли

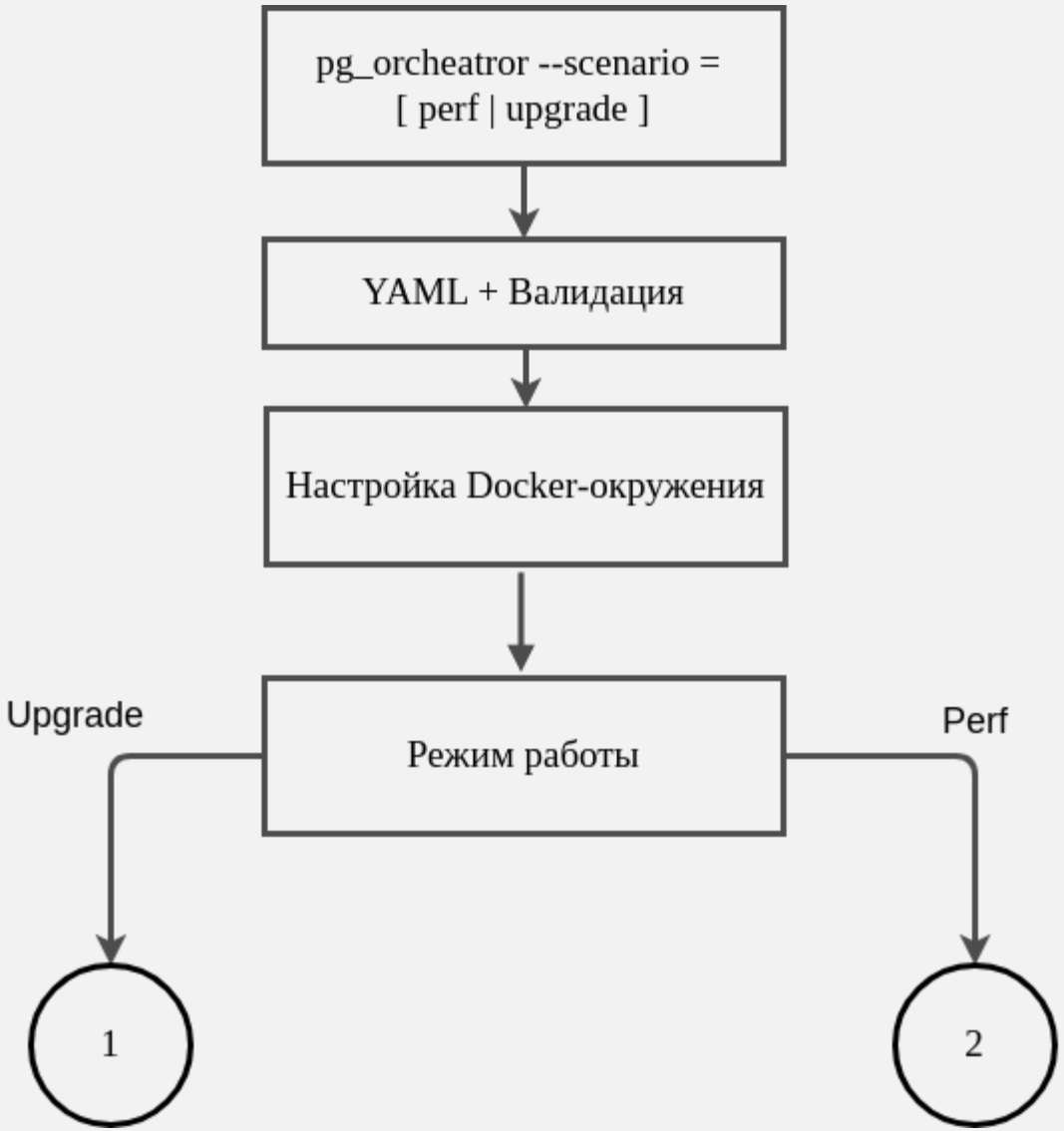
Архитектурный обзор решения

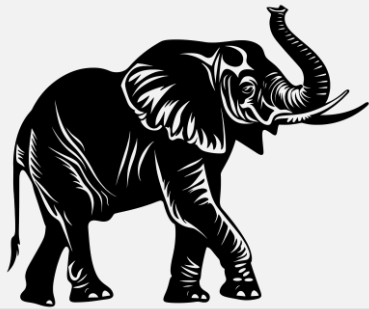
Стандартизация процессов

- YAML-конфигурации позволяют задать единый подход к настройке и запуску сценариев.

Изоляция тестового окружения

- Полностью изолированные Docker-контейнеры гарантируют стабильность и независимость тестов.





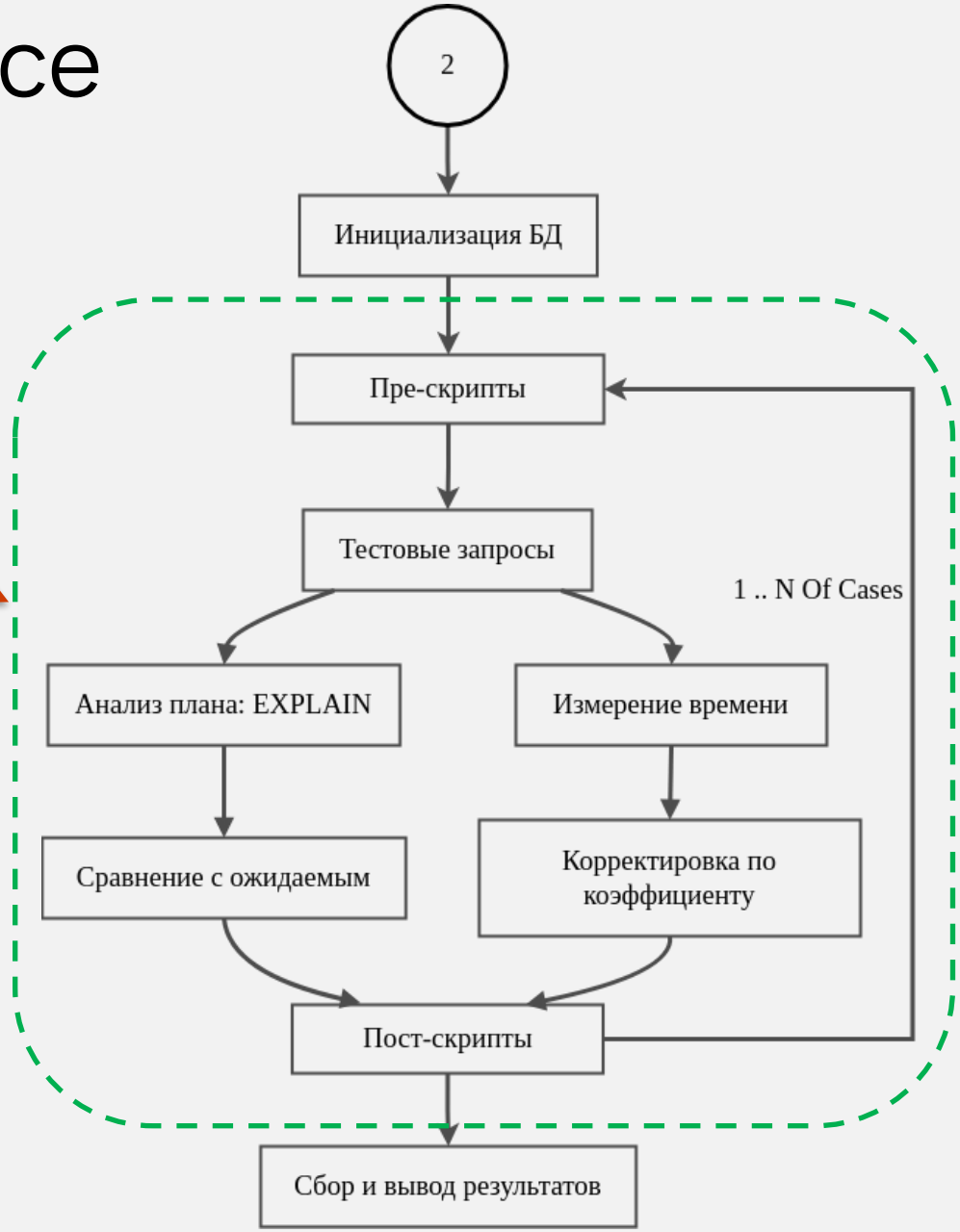
Архитектура модуля Perfomance

```
# Инициализация БД
kind: perf
db_initial_script: fill_db.sh

# Параметры базы данных для тестирования
db_params:
 - db_type: pgdg
   db_version: 16.4
   db_port: 5432
```

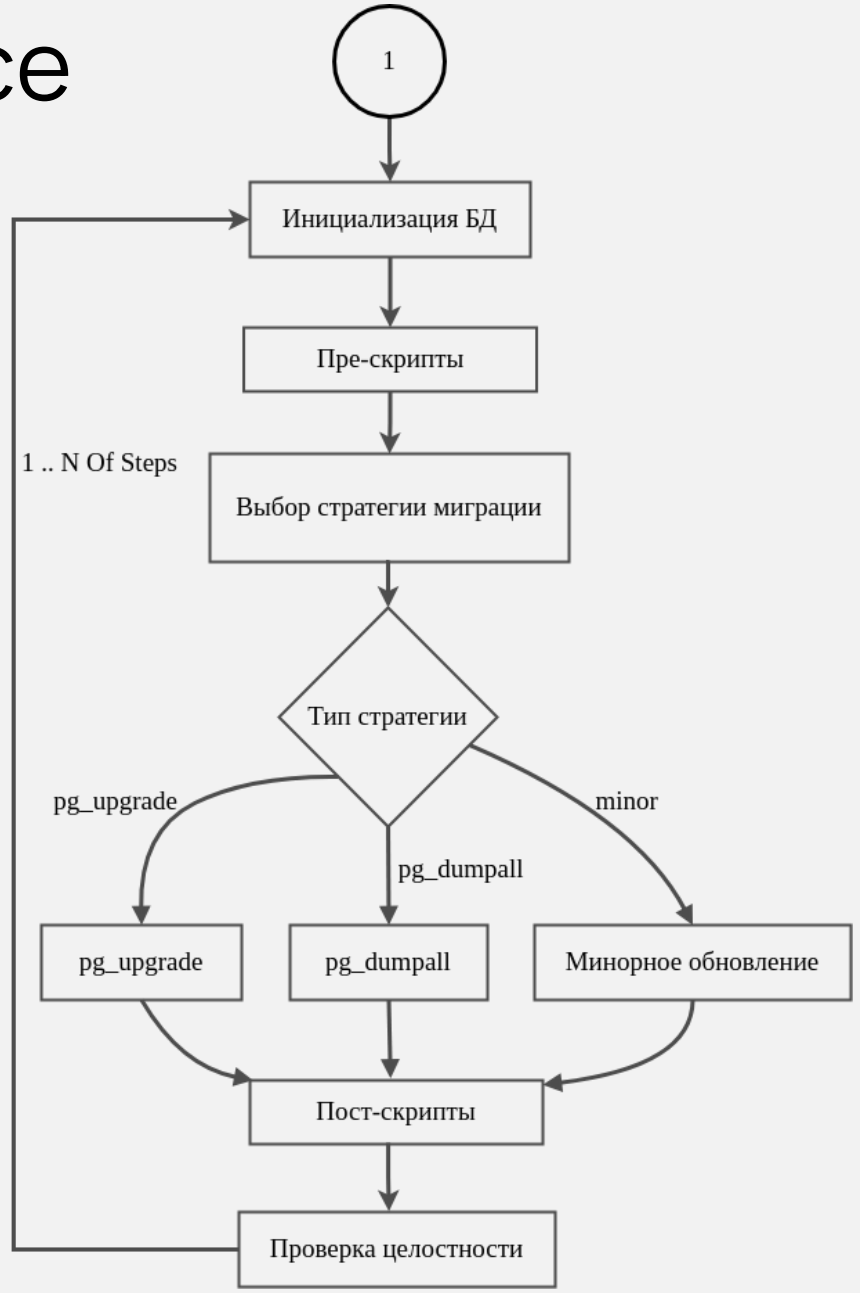
Архитектура модуля Perfomance

```
pre_scripts:
- setup_perf_environment.sh
# "1..N Of Cases"
cases:
 - name: case_1
   explain_queries:
    - query: explain_query.sql
      expected:
       - explain_expected_1.txt
    timing_queries:
    - query: timing_query.sql
      expected_time_ms: 1500
 - name: case_N
post_scripts:
- gather_perf_results.sh
```



Архитектура модуля Perfomance

```
kind: upgrade
args:
  initdb: --encoding=UTF8 --locale=ru_RU.UTF-8
db_version: 14.17
db_type: pgdg
# "1 .. N Of Steps"
steps:
   - db_version: 15.6
     type: pg_upgrade
     args:
       pg_upgrade: --link
   - db_version: 15.12
     type: minor
   - db_version: 16.8
     type: pg_upgrade
```



Предыстория: Ручное обновление и его сложности

- Работа с базой 15 версии, требующей обновления до 16 версии
- Ручной процесс -> множество команд, настройка, миграция
- Сложности обновления расширений, например, pgaudit
- Необходимость предрелизного контроля обновлений

Исходная СУБД Tantor 15 BE

- Перед началом обновления

```
/opt/tantor/db/15/bin/pg_ctl -D dbcluster15/ -l logfile start

postgres=# SELECT tantor_version();
>>
        tantor_version
------------------------------
 Tantor Basic Edition 15.12.0
(1 row)

postgres=# \dx
>>
                  List of installed extensions
  Name   | Version |   Schema   |           Description
---------+---------+------------+---------------------------------
 pgaudit | 1.7     | public     | provides auditing functionality
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
(2 rows)
```

Ручное обновление СУБД (15 → 16)

```
# Создание и инициализация нового будущего кластера СУБД
mkdir dbcluster16/
/opt/tantor/db/16/bin/initdb -D dbcluster16/

# Копирование конфигурации прошлого кластера
cp -r dbcluster15/postgresql.conf dbcluster16/

# Запуск pg_upgrade
/opt/tantor/db/16/bin/pg_upgrade --link \
-b /opt/tantor/db/15/bin \
-B /opt/tantor/db/16/bin \
-d dbcluster15 \
-D dbcluster16

# После завершения обновления создан файл update_extensions.sql
cat update_extensions.sql
>>
  \connect postgres
  ALTER EXTENSION "pgaudit" UPDATE;
```

Ручное обновление СУБД (15 → 16)

```
# Запуск нового кластера
/opt/tantor/db/16/bin/pg_ctl -D /tmp/new_data start
>>
  server started

# Запуск обновления расширения
/opt/tantor/db/16/bin/psql -f update_extensions.sql
>>
  You are now connected to database "postgres" as user "postgres".
  2025-04-06 11:50:27.974 UTC [694] STATEMENT:  ALTER EXTENSION "pgaudit" UPDATE;
  psql:update_extensions.sql:2: ERROR:  extension "pgaudit" has no update path from version
  "1.7" to version "16.0"
```

Проверка новой версии 16 BE

```
# Проверим версию СУБД
postgres=# SELECT tantor_version();
>>
       tantor_version
----------------------------
 Tantor Basic Edition 16.8.1
(1 row)

postgres=# SELECT * FROM pg_available_extensions() WHERE name = 'pgaudit';
>>
  name   | default_version |            comment
---------+-----------------+--------------------------------
 pgaudit | 16.0            | provides auditing functionality
(1 row)
```

Сравнение версий расширений

\\wsl.localhost\Ubuntu\tmp\1.7\pgaudit\Makefile

pgaudit 1.7

```
# contrib/pg_audit/Makefile

MODULE_big = pgaudit
OBJS = pgaudit.o $(WIN32RES)

EXTENSION = pgaudit
DATA = pgaudit--1.7.sql
PGFILEDESC = "pgAudit - An audit loggi

REGRESS = pgaudit
REGRESS_OPTS = --temp-config=$(top_src
```

\\wsl.localhost\Ubuntu\tmp\16.0\pgaudit\Makefile

pgaudit 16.0

```
# contrib/pg_audit/Makefile

MODULE_big = pgaudit
OBJS = pgaudit.o $(WIN32RES)

EXTENSION = pgaudit
DATA = pgaudit--16.0.sql
PGFILEDESC = "pgAudit - An audit logging exter

REGRESS = pgaudit
REGRESS_OPTS = --temp-config=$(top_srcdir)/cor
```

```
# Отсутствие скрипта обновления расширения
pgaudit--1.7--16.0.sql
```

Решение проблемы

```
psql:update_extensions.sql:2: ERROR:  extension "pgaudit" has no update path from version
  "1.7" to version "16.0"
```

```
# Удаление расширения
DROP EXTENSION pgaudit CASCADE;

# Создание расширения
CREATE EXTENSION pgaudit;
```

```
# Скрипт обновления
  pgaudit—-1.7-—16.0.sql
```

Создание патча

```
diff --git a/Makefile b/Makefile
index e4bf64f..744c91e 100644
--- a/Makefile
+++ b/Makefile
@@ -4,7 +4,7 @@ MODULE_big = pgaudit
 OBJS = pgaudit.o $(WIN32RES)

 EXTENSION = pgaudit
-DATA = pgaudit--16.0.sql
+DATA = pgaudit--1.7--16.0.sql pgaudit--16.0.sql
 PGFILEDESC = "pgAudit - An audit logging extension for PostgreSQL"

 REGRESS = pgaudit
diff --git a/pgaudit--1.7--16.0.sql b/pgaudit--1.7--16.0.sql
new file mode 100644
index 0000000..970c1c2
--- /dev/null
+++ b/pgaudit--1.7--16.0.sql
@@ -0,0 +1 @@
+-- migration from version 1.7 to 16.0
\ No newline at end of file
```

pg_orchestrator: протестируй наше обновление!

Будущее проекта

Генерация HTML/DOC отчетов ->

- Автоматическое создание подробных отчетов по результатам тестов и миграций

Кэширование результатов тестов ->

- Запоминание результатов предыдущих запусков с возможностью сравнения нескольких версий

Работа с продовыми базами ->

- Возможность подключения к уже запущенным БД для тестирования запросов и EXPLAIN-планов

Интеграция CI/CD ->

- Автоматизированный запуск сценариев тестирования и обновлений в рамках CI/CD-пайплайна

PG BootCamp Russia 2025 Ekaterinburg

PGBootCamp.ru

Спасибо!

www.tantorlabs.ru