1. Мониторинг технических метрик с Zabbix.

Мониторинг с Zabbix настроен следующим образом:
- В docker-compose подняты zabbix-server, zabbix-web и postgres DB.
- Приложение запущено локально.
- Локально же установлен zabbix-агент для сбора технических метрик.

Как оказалось, сам zabbix предоставляет уже довольно широкий набор технических метрик для мониторинга. По сути требовалось только настроить правильное взаимодействие. На скриншоте показаны самые простые метрики — утилизация CPU и использование памяти. Всплески по CPU связаны с тем, что утилитой HEY в 20 параллельных запросов осуществлялось нагрузочное тестирование. Очевидно, оно повлияло на CPU, но не затронуло использование памяти, из чего делаем вывод, что мониторинг работает верно.

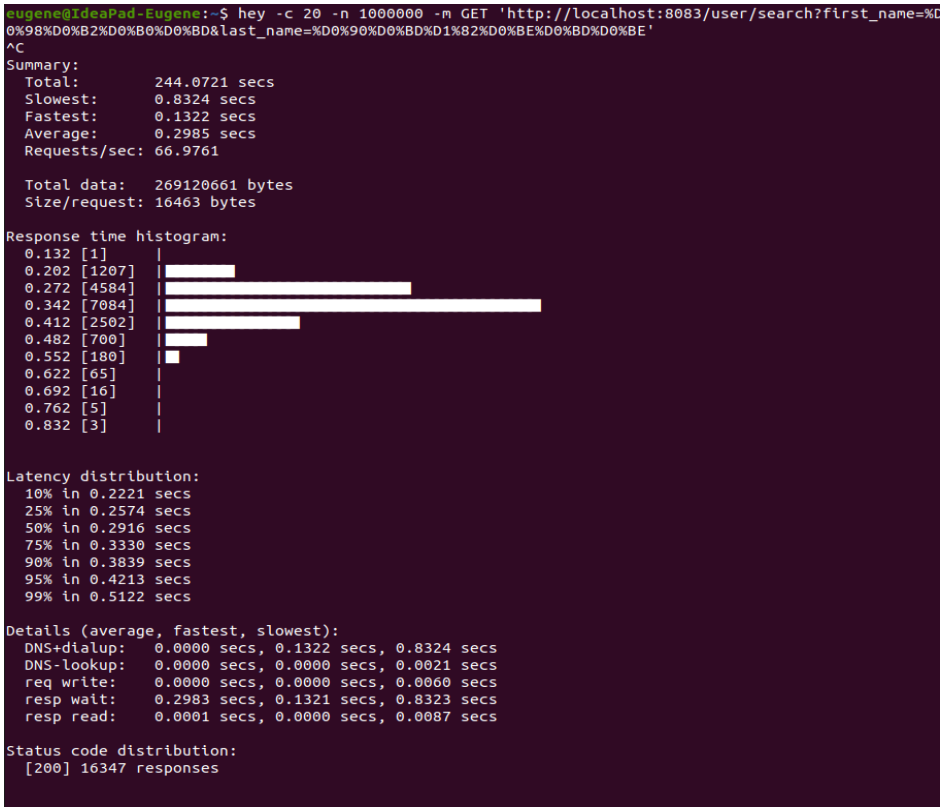

Рис. 1. Нагрузка приложения

Рис. 2. Мониторинг в Zabbix

2. Мониторинг в Prometheus.

По заданию предполагается мониторинг по принципу RED, что означает сбор и отображение метрик по Rate, Error и Duration.
Мониторинг реализован следующим образом:
- в docker-compose вместе с кластером БД postgresql (который планируется мониторить) поднимаются pg-exporters, по одному на каждый инстанс БД (всего 3), которые собирают метрики и готовы отдавать скраперу prometheus.
- Сам prometheus, как и grafana для визуализации, установлены локально как systemd-сервисы.

PG-экспортеры предоставляют довольно широкий спектр метрик. Также отсюда (https://grafana.com/oss/prometheus/exporters/postgres-exporter/?tab=dashboards#metrics-usage) взят пример дашборда для postgres. Здесь QPS, Number of connections и Rows отвечают за Rate, а Conflicts/Deadlocks - за Errors.

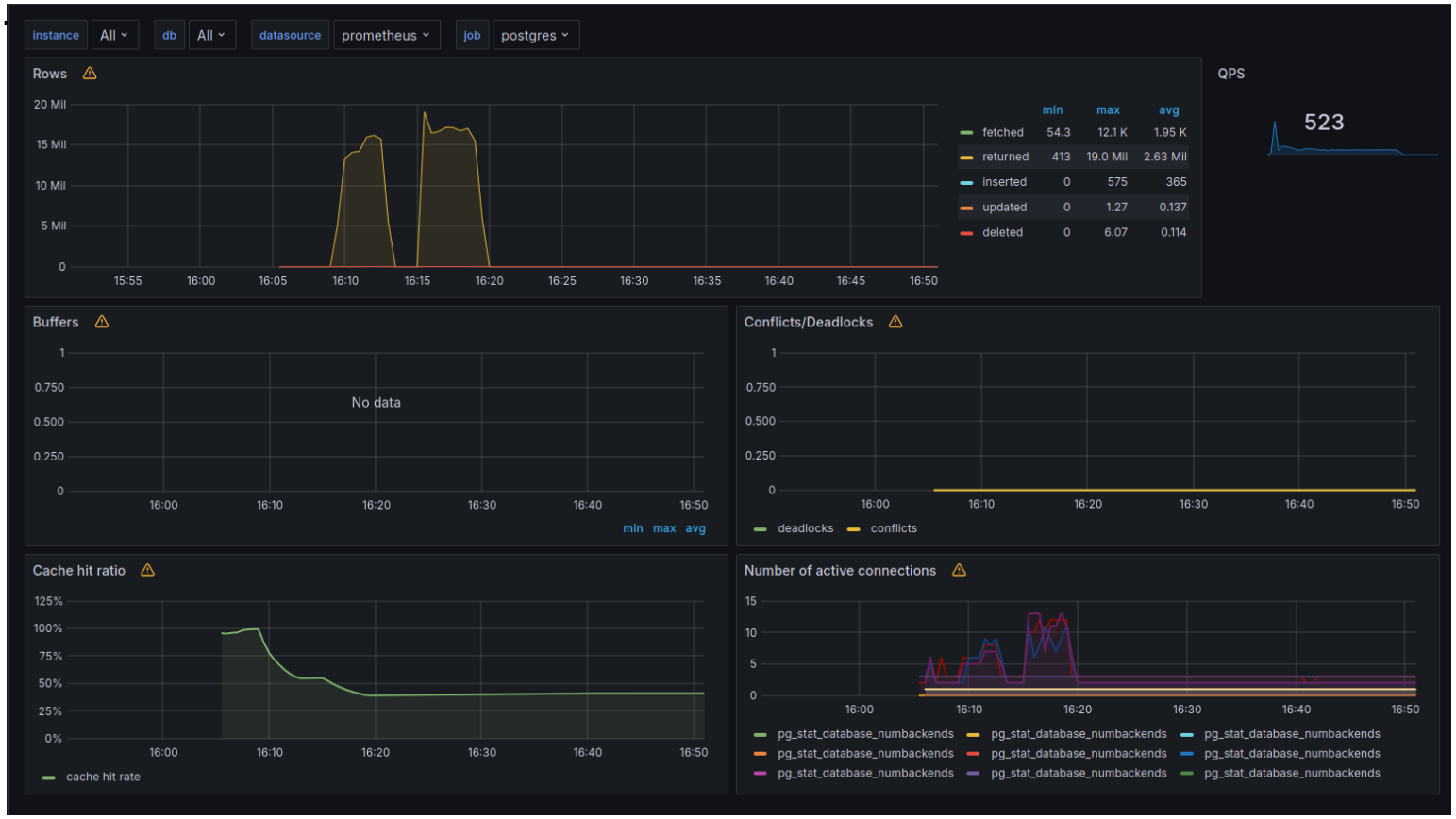

Рис. 3. Дашборд для Postgres

На предыдущем дашборде не хватает метрик, отвечающих за Duration из принципа RED. Поэтому на отдельной дашборде выводим:
 - sum(pg_stat_activity_max_tx_duration{datname="social_net", state="active"}) by (server)
- и метрику, косвенно связанную с duration - количество локов.

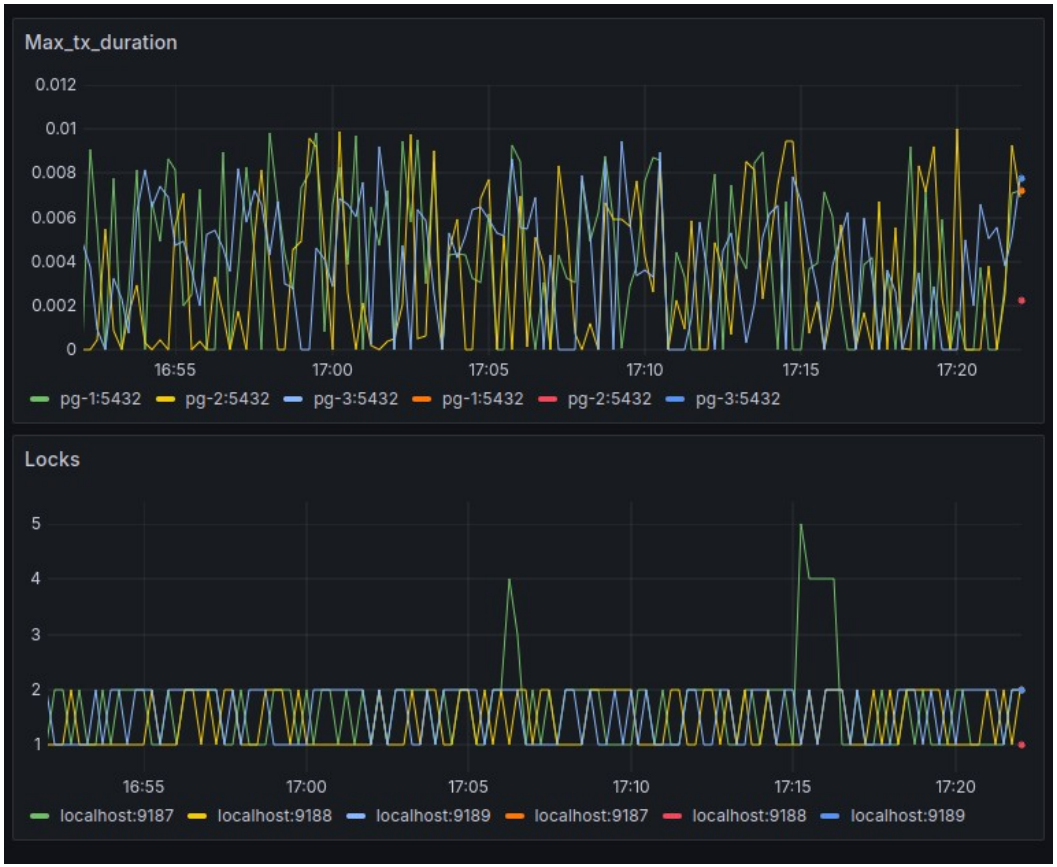

Рис. 4. Еще метрики.

Совсем логично тут было бы использовать total_time, mean_time из pg_stat_statements (запросы для них есть в репозитории в queries.yaml), но их сбор не получилось настроить.