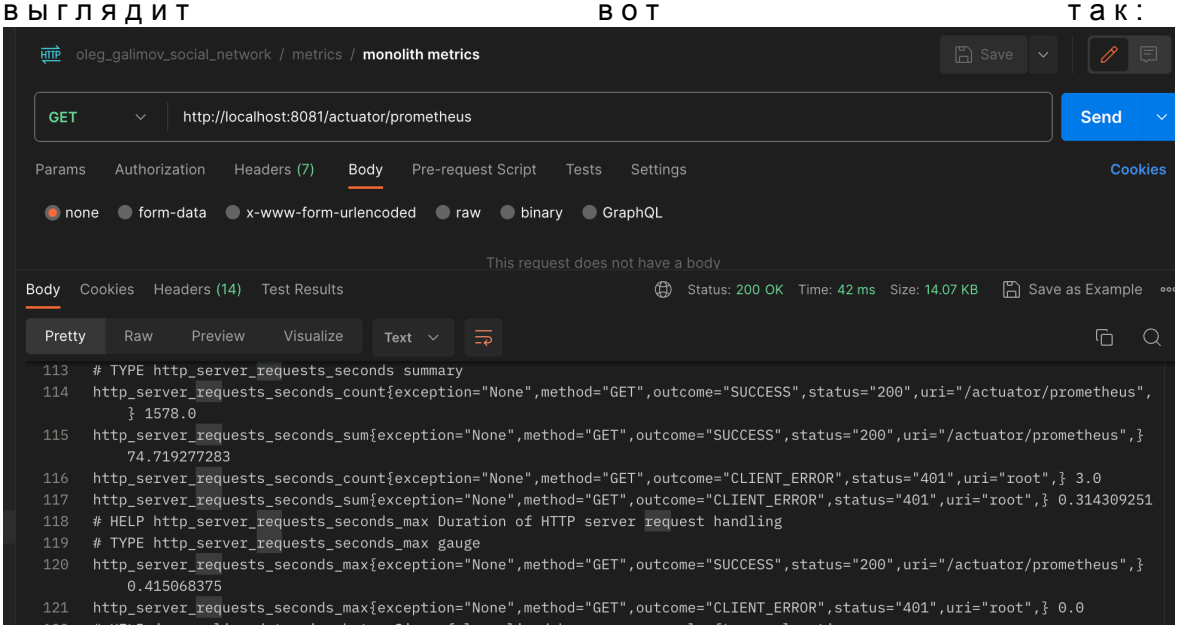
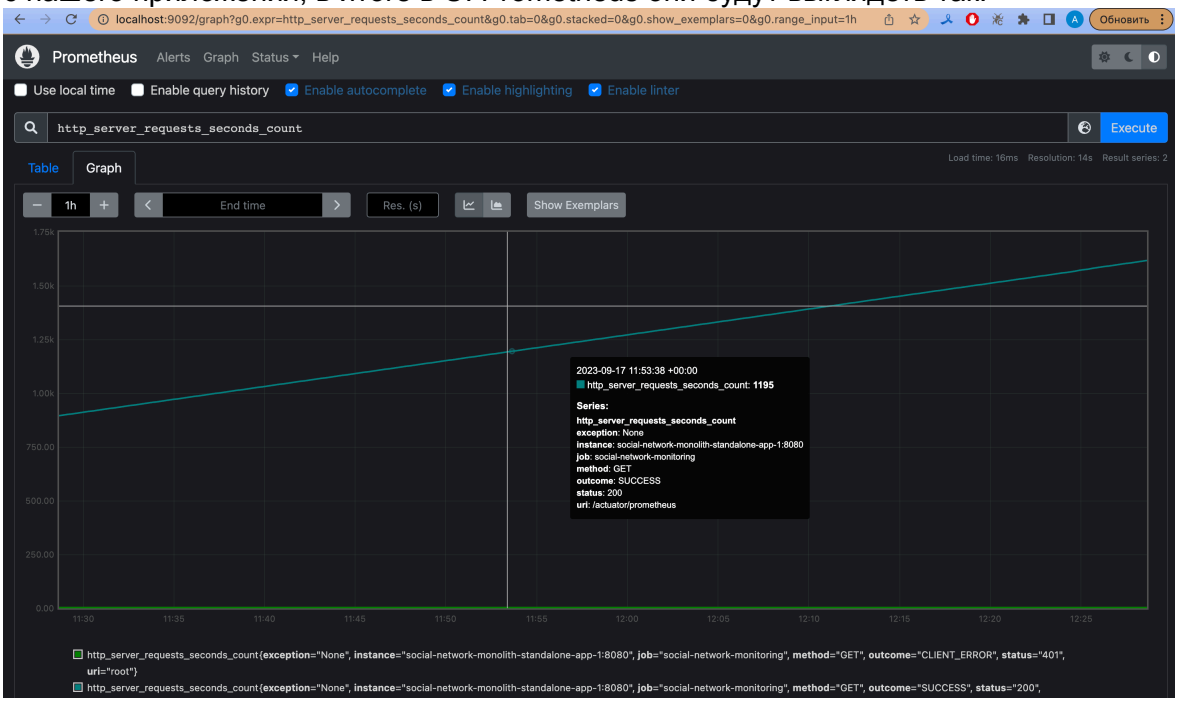
Мониторинг

1. Для того, чтобы реализовать мониторинг приложения, добавим в зависимости библиотеку Micrometer, и настроим эндпойнт /actuator/prometheus, через который можно будет запросить все метрики нашего приложения. В сыром виде это выглядит вот так:

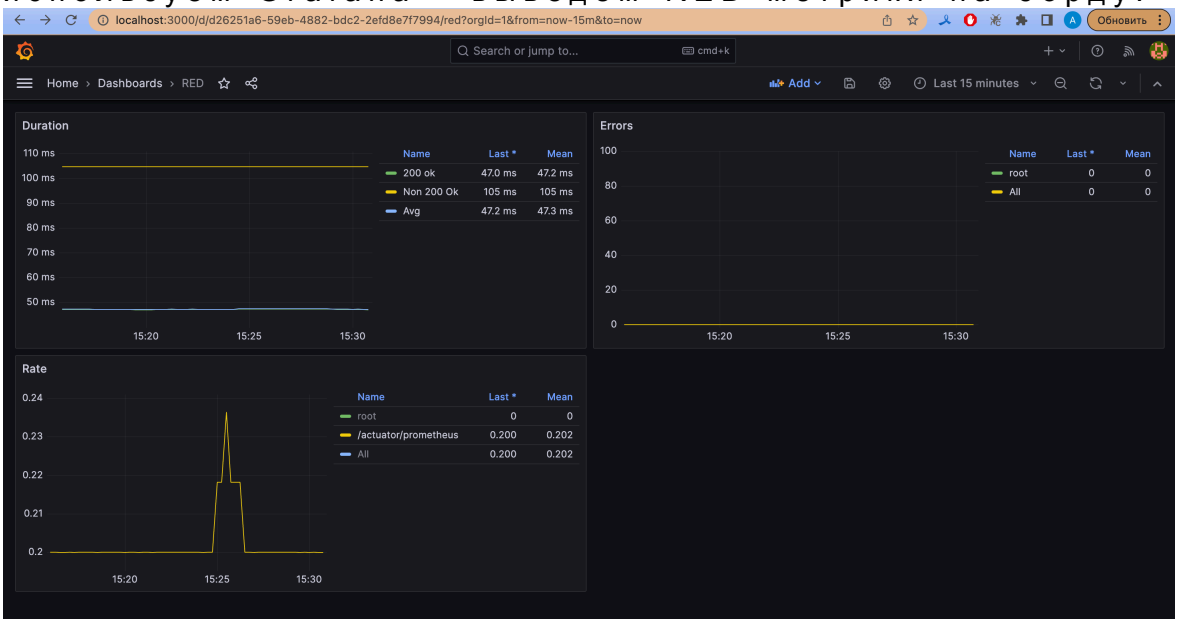


Видно, что здесь присутствуют искомые бизнес-метрики, которые необходимы для RED-подхода.

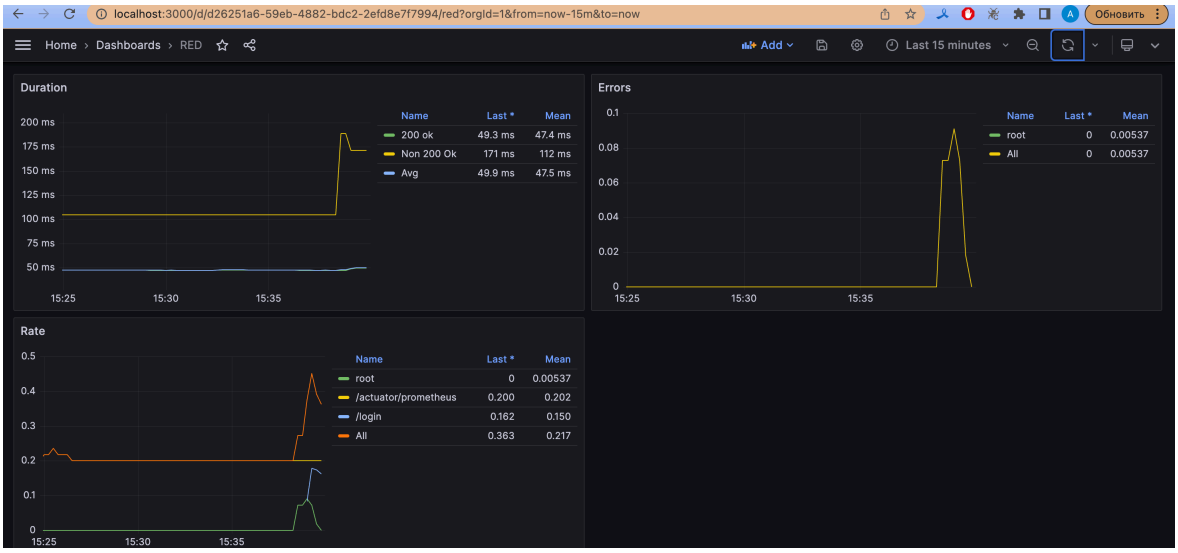
2. Теперь настроим Prometheus для периодического сбора и аггрегации этих метрик с нашего приложения, в итоге в UI Prometheus они будут выглядеть так:



3. Но для того, чтобы иметь возможность удобно наблюдать за метриками используем Grafana - выведем RED-метрики на борду:

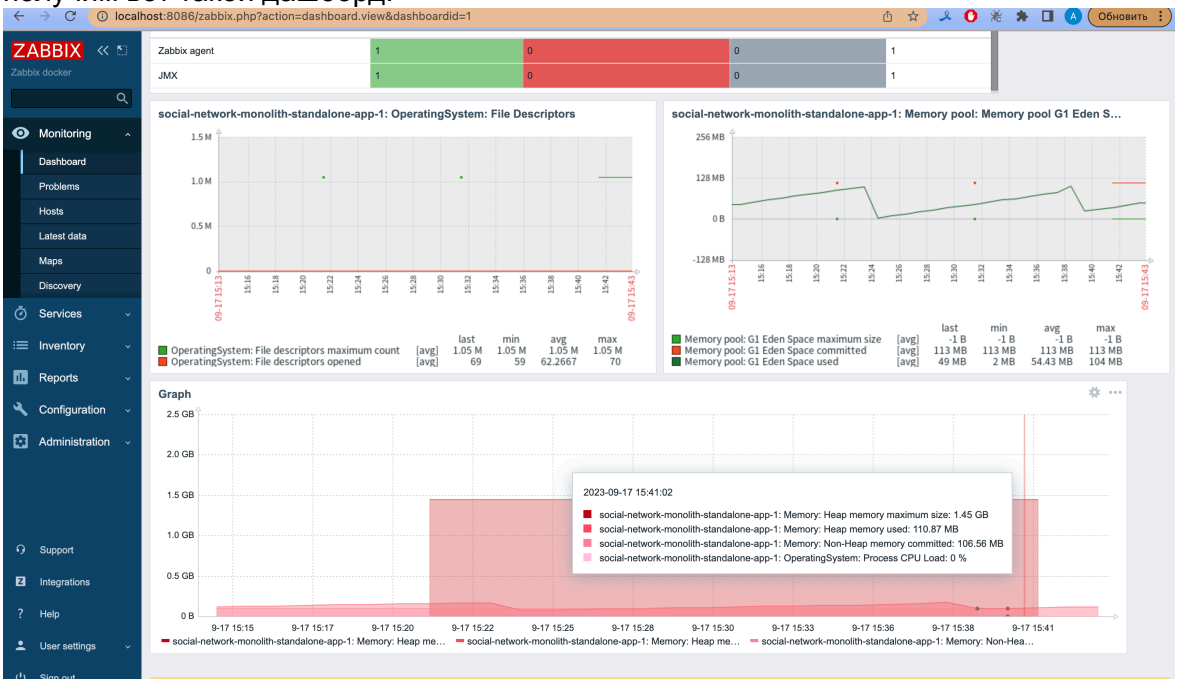


4. Убедимся, что все работает корректно - для этого запросим 10 раз страницу с логином и 5 раз - попробуем получить переписку не залогинившись, вызвав тем самым всплеск 401-х ошибок:



5. Бизнес-метрики настроены, теперь нужно осуществить мониторинг технических метрик. Для этого настроим отдельный инстанс БД Postgres для Zabbix, запустим сам сервис Zabbix, его UI, запустим агента Zabbix и Zabbix-Java-Gateway, чтобы получить связку - Сервер мониторинга+Агенты+Приложение. В UI Zabbix мы добавим новый хост - наше приложение и выведем на борду например метрики утилизации памяти, а так же размеры хипа (как показывает практика, при разборе проблемы именно эти метрики являются наиболее полезными). В результате

получим вот такой дашборд:



Вывод: для приложения настроен мониторинг бизнес-метрик по методу RED, соответствующий дашборд хранится в каталоге grafana вместе с исходным кодом приложения. Также настроен мониторинг технических метрик с помощью Zabbix. Конфигурация развертывания сервисов для мониторинга описана в docker-compose, что позволяет ее быстро переиспользовать.