

Предположим, нам нужна система мониторинга доступности IP-адресов. Напишем небольшое приложение, которое позволит регистрировать и удалять IP-адреса, а также выполнять проверку их доступности, где используя результаты проверки мы можем выполнить расчет статистики.

API

- POST /ips - добавить адрес с параметрами (enabled: bool, ip: ipv4/ipv6 address)
- POST /ips/:id/enable - включить сбор статистики ip
- POST /ips/:id/disable - выключить сбор статистики ip
- GET /ips/:id/stats - получить статистику для адреса (time_from: datetime, time_to: datetime)
- * DELETE /ips/:id - выключить сбор и удалить адрес

Требования к задаче

- проверка должна выполняться раз в N времени (например, раз в минуту);
- расчет статистики целиком на уровне базы данных;
- расчет статистики должен включать в себя за период: среднее RTT (время отклика), минимальное RTT, максимальное RTT, медианное RTT, среднеквадратичное отклонение замеров RTT, процент потерянных пакетов ICMP до указанного адреса;
- не использовать rails, за исключением activesupport, Active Record;
- если проверка доступности занимает более одной секунды, то такая проверка считается неудачной (потеря пакетов) и должна быть прервана;
- если какую-то часть времени в выбранном периоде IP-адрес был вне расчета статистики (не был добавлен или был удален) — эту часть времени учитывать не нужно. Например, мы добавили IP-адрес 8.8.8.8 в 1 час, выключили в 2, включили в 3 и выключили в 4. Если запрашиваем статистику с 1 по 4 часа — надо объединить интервалы 1-2, 3-4 и отдать эту статистику по объединенному интервалу. Если IP-адрес не был в расчете статистики всё время или был настолько мало времени, что мы не успели сделать хотя бы 1 замер, – надо вернуть сообщение об ошибке.

Необязательно, но будет плюсом

- не использовать activesupport и Active Record;
- сделать запуск проекта с помощью docker compose;
- добавить поддержку IPv6 IP-адресов;
- настроить GitHub Actions (или любой другой CI, зависит от места размещения тестового задания) с запуском тестов.