

**Бэкофис программы для ЭВМ «Система  
управления электронной коммерцией**

**GateLine eCommerce»**

**Инструкция по установке**

## **Оглавление**

1 Backend приложения .....	2
1.1 Репликация данных.....	2
1.2 Мониторинг .....	2
1.3 Настройка zabbix в staging / rc окружениях .....	2
2 Архитектура проекта .....	3
3 API .....	3
3.1 Deployment.....	4
3.1.1 Процессы .....	4
3.1.2 Переменные окружения .....	4
3.1.3 Файлы конфигурации .....	8
3.1.4 JWT ключи.....	9
3.1.5 Зависимости .....	9
3.1.6 CI / CD.....	9
3.2 Rake задачи .....	10
3.2.1 Front.....	13
3.3 Агенты.....	13
4 Frontend приложения .....	14
4.1 Зависимости .....	14
4.2 Артефакт.....	14
4.3 Развертывание .....	14

## **Перечень сокращений**

<b>Термин</b>	<b>Определение</b>
БД	База данных.
ОС	Операционная система.
ПО	Программное обеспечение

## **1 Backend приложения**

### **1.1 Репликация данных**

Репликация данных работает на основе обычной master-slave репликации MySQL. Ears - это golang демон, который постоянно пулит replication log из MySQL, получает последние события из этого лога. На основе этих событий генерирует задачи для Sidekiq процесса, после чего сохраняет текущую позицию в логе и само название лога в Redis. При перезапуске демон начинает работать с последней сохраненной позиции.

### **1.2 Мониторинг**

Сбор метрик осуществляется с помощью zabbix и statsd.

Для анализа багов используется sentry.

Для быстрого конфигурирования zabbix используется специальная rake задача.

### **1.3 Настройка zabbix в staging / rc окружениях**

- Staging - http:// [IP]:[PORT]
- RC - http:// [IP]:[PORT]

Login / password = Admin / zabbix

Открыть Configuration -> Host -> Create host

- Host name: [IP]
- Visible name: [IP]
- Groups: Linux Servers  Agent interfaces: [IP]

Нажать Add.

Открыть Templates -> Import.

Через Import files открыть заранее подготовленный файл с помощью rake задачи шаблона.

Нажать Import.

Открыть Configuration -> Hosts -> [IP] -> Templates Выбрать

Templates Gateline, нажать Select.

Нажать Update.

Открыть Monitoring -> Latest data. Hosts:

[IP]

Нажать Apply.

Через 5 минут метрики должны появиться.

## 2 Архитектура проекта

Архитектура проекта представлена на рисунке 1.

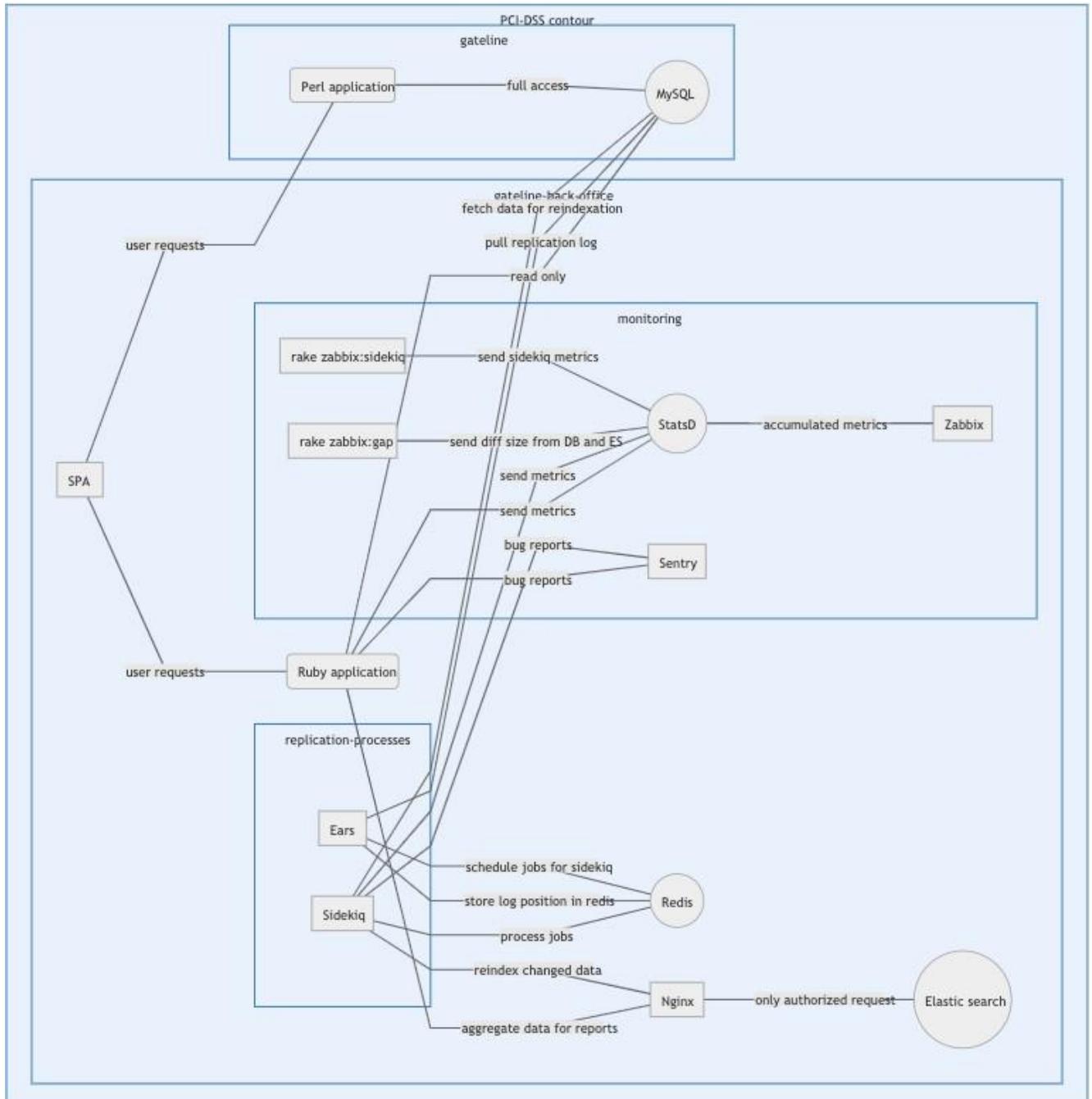


Рисунок 1 – Архитектура проекта

## 3 API

API проекта расположено:

1. <https://adm-api.sandbox.gateline.net/doc>
2. <https://adm-api.dev.gateline.net/doc>

### 3.1 Deployment

#### 3.1.1 Процессы

Основные процессы описаны в данном разделе.

Таблица 1 – Процессы

Название	Описание	Команда запуска
Puma	Веб сервер	bin/bundle exec rails s
Sidekiq	Процесс обработки фоновых задач	bin/bundle exec sidekiq
Ears	Процесс репликации данных	bin/ears -config config/ears.yml -env production

#### 3.1.2 Переменные окружения

В настоящем разделе описаны переменные окружения.

Таблица 2 – Переменные окружения

Имя	Пример	Описание
RAILS_ENV	production, development, staging	Название окружения
CORS_ORIGINS	localhost:xxxx, adm.sandbox.gateline.net	Доверенные хосты для CORS
GATELINE_MYSQL_HOST	[IP]	Адрес MySQL хоста приложения
GATELINE_MYSQL_PORT	[PORT]	Номер порта MySQL хоста приложения

Имя	Пример	Описание
GATELINE_MYSQL_USER	billing_ro	Имя пользователя MySQL хоста приложения
GATELINE_MYSQL_PASSWORD	[password]	Пароль пользователя MySQL хоста приложения

GATELINE_MYSQL_DB	billing	Имя базы данных приложения
EARS_MYSQL_HOST	[IP]	Адрес MySQL хоста процесса репликации
EARS_MYSQL_PORT	[PORT]	Номер порта MySQL хоста процесса репликации
EARS_MYSQL_USER	backoffice	Имя пользователя MySQL хоста процесса репликации
EARS_MYSQL_PASSWORD	[password]	Пароль пользователя MySQL хоста процесса репликации
EARS_BUFFER_SIZE	1000	Максимальное кол-во буферезированных задач Ears демона
EARS_BUFFER_TIMER	1000	Максимальное время (миллисекунды) между отправлением задач Ears демона в очередь
ES_HOST	elastic	Адрес nginx / точки доступа до Elastic Search
ES_LOGIN	admin	Логин для доступа к Elastic Search
ES_PASSWORD	pantiesOnTheHead	Пароль для доступа к Elastic Search
SLAVE_SERVER_ID	4	ID процесса репликации, необходимая настройка для MySQL репликации.

Имя	Пример	Описание
SIDEKIQ_PROCESS_ID	4	ID процесса репликации, необходимый для работы Ears процесса

REDISKA_HOST	redis	Адрес Redis хоста приложения и процесса репликации
REDISKA_PORT	[PORT]	Номер порта Redis хоста приложения и процесса репликации
REDISKA_DB	0	Номер базы данны Redis хоста приложения и процесса репликации
REDISKA_PASSWORD	redisPassword	Пароль к Redis
REDISKA_CLIENT_POOL_SIZE	3	Размер connection pool - а до Redis хоста веб сервера
REDISKA_SERVER_POOL_SIZE	12	Размер connection pool - а до Redis хоста Sidekiq воркеров
REDISKA_POLL_INTERVAL	1	Кол-во секунд между выборкой новых задач для Sidekiq воркера
REDISKA_TIMEOUT	2000	время (милисекунды) на запрос до редиса
REDISKA_OPEN_TIMEOUT	500	время (милисекунды) на открытие соединения до редиса
RAILS_MAX_THREADS	10	Максимальное количество процессов веб сервера
SIDEKIQ_CONCURRENCY	10	Максимальное количество процессов Sidekiq
RACK_ENV	staging, rc, production	Имя окружения приложения

Имя	Пример	Описание
-----	--------	----------

TZ	Europe/Moscow	Временная зона приложения, необходимая для корректной обработки дат базы данных
SECRET_KEY	-	Секретный ключ для сессия веб сервера (рандомная строка длиной ~ 100 символов)
SENTRY_DSN	-	Sentry DSN ключ
EARS_KEY	-	Ключ по которому Ears хранит свою позицию
STATSD_HOST	-	Адрес StatsD демона
JWT_KEY_PATH	./config/keys/rc.public.key	Путь до публичного JWT ключа
CHEWY_TIMEOUT	2	время (секунды) на запрос
CHEWY_OPEN_TIMEOUT	1	время (секунды) на открытие коннекта до эластика
SYSLOG_NAME	gateline	имя лога (актуально только production)
SIDEKIQ_USERNAME	test	логин для доступа к sidekiq
SIDEKIQ_PASSWORD	pass	пароль для доступа к sidekiq
SIDEKIQ_DOOR	/sidekiq	эндпоинт веб морды sidekiq
BATCH_SIZE	1000	размер пачек запрашиваемых из БД
PREAGGREGATE_PAUSE	5	Размер паузы (секунды) между преагрегациями.

Имя	Пример	Описание
LOG_LEVEL	debug, info, error	Уровень логирования
STATSD_PROCESS_NAME	super_duper	Имя процесса для метрик сборщика мусора.
ES_CERT	config/keys/elastic/elastic.crt	Самоподписанный сертификат для elasticsearch.
ES_KEY	config/keys/elastic/elastic.key	Самоподписанный ключ для elasticsearch.
ADMIN_API_SUBDOMAIN	adm-api-sbp.sandbox	Поддомен Admin API
CLIENT_API_SUBDOMAIN	client-api-sbp.sandbox	Поддомен Client API

Для отправления метрик сборщика мусора в zabbix используется переменная окружения STATSD\_PROCESS\_NAME. Исходя из значения этой переменной генерируются имена item-ов, например: STATSD\_PROCESS\_NAME.gc.heap\_allocatable\_pages. Для генерации конфигурация забикса со всеми необходимыми именами item-ов есть специальная rake таска zabbix:gc\_template.

### 3.1.3 Файлы конфигурации

В настоящем разделе представлены файлы конфигурации

Таблица 3 – Файлы конфигурации

Путь	Описание
chewy.yml	Конфигурация chewy - gem для работы с ES
credentials.yml.enc	ключ Rails приложения
database.yml	Конфигурация коннекта к базам данных
ears.yml	Конфигурация процесса репликации
secrets.yml	Конфигурация секретных ключей Rails
sidekiq.yml	Конфигурация Sidekiq

Все конфигурационные файлы используются как шаблоны, куда подставляются переменные окружения, поэтому вручную их менять не нужно. При разработке приложения для

конфигурации переменных окружения используются .env файлы, в production среде их возможно лучше удалить, так как они могут переопределять системные переменные и вызывать вопросы у администраторов.

### 3.1.4 JWT ключи

Для авторизации API используются JWT. При шифровании ключа используется RS256 алгоритм, которому необходимо два ключа. Приватный ключ используется Perl приложением для формирования токена, публичный ключ находится на Ruby приложении для верификации токенов. Ключи лежат в папке./config/keys. Имя ключа имеет формат [имя окружения].public.key.

### 3.1.5 Зависимости

Для корректной работы приложения в системе должна быть установлена таймзона Europe/Moscow.

Полный список зависимостей для Debian и сам процесс установки и настройки системы смотреть в Dockerfile.

Таблица 4 – Зависимости

Имя	Версия	URL
ruby	2.6.1	<a href="https://cache.ruby-lang.org/pub/ruby/2.6/ruby-2.6.1.tar.gz">https://cache.ruby-lang.org/pub/ruby/2.6/ruby-2.6.1.tar.gz</a>
libmysqlclient21	0.8.12-1	<a href="https://dev.mysql.com/get/mysql-apt-config_0.8.12-1_all.deb">https://dev.mysql.com/get/mysql-apt-config_0.8.12-1_all.deb</a>
libmysqlclientdev	0.8.12-1	<a href="https://dev.mysql.com/get/mysql-apt-config_0.8.12-1_all.deb">https://dev.mysql.com/get/mysql-apt-config_0.8.12-1_all.deb</a>
elasticsearch	6.6.1	<a href="https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.6.1.deb">https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.6.1.deb</a>
mysql	*	
redis	*	

### 3.1.6 CI / CD

Каждая ревизия проходит 4 этапа сборки:

- build
- check

- package
- deploy

С deployment-ом связаны только два последних.

На этапе package происходит сохранение всех ruby пакетов в локальную папку vendor/cache и архивация всего проекта в файл revision.tar.gz. После успешного прохождения этапа package архив revision.tar.gz доступен как артифакт в последующих этапах сборки и для загрузки (например как здесь [https://Git\\_Server/gateline.net/backoffice/back-/jobs/91879/artifacts/browse](https://Git_Server/gateline.net/backoffice/back-/jobs/91879/artifacts/browse)) на странице job - й.

В упрощенном виде это происходит в 5 команд:

- ./bin/bundle install --local - установка пакетов из папки vendor/cache (некоторые пакеты требуют компиляции)
- ./bin/bundle check - проверка наличия всех ruby зависимостей
- ./bin/bundle exec sidekiq - запуск процесса обработки фоновых задач (может быть запущено любое количество)
- ./bin/bundle exec rails s -p xxxx - запуск веб сервера (может быть запущено любое количество)
- ./bin/ears -config config/ears.yml -env staging - запуск процесса репликации данных (должен быть запущен один на кластер)

Все эти команды запускаются внутри docker-compose который использует dockercompose-staging.yml и docker-compose-rc.yml файлы. Более детально этапы CI/CD для staging окружения смотреть в .gitlab-ci.yml.

### 3.2 Rake задачи

Таблица 5 – 3.2 Rake задачи

<b>Задача</b>	<b>Описание</b>
rake zabbix:template	Генерация конфига для zabbix со списком метрик
rake zabbix:gap	Сбор метрик / разницы между индексом и БД
rake zabbix:sidekiq	Сбор метрик sidekiq

rake index:create_template	Создание всех необходимых темплейтов индексов с настройками
rake sync:all	Полная переиндексация всей БД

Задача	Описание
rake sync:activation:between[from,to]	Преиндексация таблицы activation за указанный период (например 30.05.2019)
rake sync:apachelog:between[from,to]	Преиндексация таблицы apachelog за указанный период (например 30.05.2019)
rake sync:between[from,to]	Преиндексация таблицы between за указанный период например.05.2019)
rake sync:clientlog:between[from,to]	Преиндексация таблицы clientlog за указанный период (например 30.05.2019)
rake sync:driverlog:between[from,to]	Преиндексация таблицы driverlog за указанный период (например 30.05.2019)
rake sync:gdsdata:between[from,to]	Преиндексация таблицы gdsdata за указанный период (например 30.05.2019)
rake sync:log:between[from,to]	Преиндексация таблицы log за указанный период (например 30.05.2019)
rake sync:mpilog:between[from,to]	Преиндексация таблицы mpilog за указанный период (например 30.05.2019)
rake sync:notify:between[from,to]	Преиндексация таблицы notify за указанный период (например 30.05.2019)

rake sync:notifylog:between[from,to]	Преиндексация таблицы notifylog за указанный период (например 30.05.2019)
rake sync:operation:between[from,to]	Преиндексация таблицы operation за указанный период (например 30.05.2019)

Задача	Описание
rake sync:order:between[from,to]	Преиндексация таблицы order за указанный период (например 30.05.2019)
rake sync:ordermessage:between[from,to]	Преиндексация таблицы ordermessage за указанный период (например 30.05.2019)
rake sync:siterulelog:between[from,to]	Преиндексация таблицы siterulelog за указанный период (например 30.05.2019)
rake sync:tasklog:between[from,to]	Преиндексация таблицы tasklog за указанный период (например 30.05.2019)
rake sync:token:between[from,to]	Преиндексация таблицы token за указанный период (например 30.05.2019)
rake sync:vtbmessage:between[from,to]	Преиндексация таблицы vtbmessage за указанный период (например 30.05.2019)
rake preaggregate:all[from,to]	Планирование задач на преагрегацию всего за указанный период.
rake preaggregate:loop	Преагрегация текущих значений всех счетчиков в цикле с паузой в PREAGGREGATE_PAUSE секунд.
rake preaggregate:clean[from,to]	Удалить все счетчики за указанный период.

rake preaggregate:cash_flow[from,to]	Предагрегация денежного оборота за указанный период (по умолчанию за сегодня).
rake preaggregate:cash_flow:current	Преагрегация текущих значений счетчиков оборота.
rake preaggregate:cash_flow:between[from,to]	Планирование задач на преагрегацию счетчиков оборота за указанный период.
Задача	Описание
rake preaggregate:cash_flow:clean[from,to]	Удалить счетчики оборота за указанный часпериод
rake preaggregate:operation_counters:current	Преагрегация текущих значений счетчиков операций.
rake preaggregate:operation_counters:between[from,to]	Планирование задач на преагрегацию счетчиков операций за указанный период.
rake preaggregate:operation_counters:clean[from,to]	Удалить счетчики операций за указанный период.
rake zabbix:gc_template	Генерация шаблона zabbix, для сбора метрик с GC.

Полный список доступных rake команд можно получить через rake -T.

### 3.2.1 Front

```
docker run --rm -ti -p 8081:80 --net=host -v
${PWD}/lib/nginx/front_nginx.conf:/etc/nginx/conf.d/default.conf
hub.onelya.ru:4567/gateline.net/backoffice/front:develop
```

### 3.3 Агенты

Так как из-за очереди и батчинга мы не можем контролировать порядок индексации дочерних и родительских моделей, то приходится внутри #insert обыгрывать оба кейса:

- дочерняя модель индексируется первой;
- родительская модель индексируется первой.

Код должен отрабатывать в обоих случаях.

Так же он должен учитывать то, что #insert используется при переиндексации.

## 4 Frontend приложения

### 4.1 Зависимости

Nginx (LTS)

### 4.2 Артефакт

[https://Git\\_Server/gateline.net/backoffice/front/-/jobs/artifacts/master/download?job=build-prod](https://Git_Server/gateline.net/backoffice/front/-/jobs/artifacts/master/download?job=build-prod)

### 4.3 Развёртывание

Для разворачивания необходимо настроить в NGINX 1 маршрут:

/ - папка со статикой (см. артефакт, содержимое папки build)

Пример файла конфигурации:

[https://Git\\_Server/gateline.net/backoffice/front/blob/master/nginx/default\\_rc.conf](https://Git_Server/gateline.net/backoffice/front/blob/master/nginx/default_rc.conf)

CORS настроен в бэкенде, однако, если потребуется настроить во фронтенде, то файл конфигурации находится здесь:

[https://Git\\_Server/gateline.net/backoffice/front/blob/master/nginx/cors.conf](https://Git_Server/gateline.net/backoffice/front/blob/master/nginx/cors.conf)