

04_инструкция_администратора

Инструкция администратора ПО “Аралтыш”

Назначение документа

Документ описывает эксплуатацию, настройку и сопровождение серверного компонента ПО “Аралтыш”.

Обязанности администратора

Администратор отвечает за:

- выпуск и хранение токена MAX-бота;
- настройку домена и HTTPS;
- настройку webhook;
- настройку PostgreSQL;
- хранение секретов;
- резервное копирование базы данных;
- мониторинг доступности сервиса;
- обновление версии приложения;
- обработку обращений пользователей.

Поддержка базовой версии ПО оказывается правообладателем лично:

- email: yasg1982@yandex.ru;
- телефон: +7 902 664-63-13;
- мессенджер MAX: на номере +7 902 664-63-13;
- регламент ответа: от 1 до 3 рабочих дней.

Управление конфигурацией

Все параметры задаются переменными окружения. Запрещается размещать значения токенов и паролей в исходном коде.

Критичные переменные:

- MAX_BOT_TOKEN ;
- MAX_WEBHOOK_URL ;
- DB_HOST ;
- DB_USER ;

- `DB_PASSWORD` ;
- `ADMIN_USER_IDS` .

Проверка состояния

Проверить HTTP endpoint:

```
curl https://sos.example.ru/health
```

Проверить контейнер:

```
docker ps  
docker logs --tail 200 araltysh
```

Проверить базу данных:

```
psql "$DATABASE_URL" -c "select now();"
```

Регламент штатной эксплуатации

Администратор выполняет следующие регулярные действия:

- ежедневно проверяет доступность `/health`;
- контролирует отсутствие повторяющихся ошибок webhook в журналах контейнера;
- проверяет, что сервис имеет доступ к MAX Bot API;
- контролирует свободное место на сервере и в хранилище резервных копий;
- проверяет успешность резервного копирования PostgreSQL;
- после обновления версии проходит контрольный сценарий родителя и ребенка;
- хранит токены, пароли и параметры базы данных вне репозитория исходного кода.

Контрольный сценарий после обновления:

1. Родитель запускает бота и получает код подключения.
2. Ребенок вводит код.
3. Родитель подтверждает семейную связь.
4. Ребенок отправляет тестовую тревогу.
5. Родитель получает уведомление и закрывает тестовое событие.

Резервное копирование

Рекомендуемый регламент:

- ежедневная резервная копия PostgreSQL;

- хранение не менее 7 ежедневных копий;
- отдельная копия перед обновлением версии;
- периодическая проверка восстановления.

Пример:

```
pg_dump "$DATABASE_URL" > aralysh_backup_$(date +%F).sql
```

Обновление версии

1. Проверить release notes новой версии.
2. Создать резервную копию PostgreSQL.
3. Получить новый Docker-образ.
4. Обновить контейнер или сервис.
5. Проверить `/health`.
6. Пройти контрольный сценарий родителя и ребенка.

Журналы

Сервис пишет журналы в stdout/stderr контейнера.

В журналах не должны публиковаться:

- токены;
- пароли;
- полные секретные значения;
- приватные параметры окружения.

Обработка инцидентов

При ошибках webhook:

1. Проверить доступность `/health`.
2. Проверить валидность `MAX_BOT_TOKEN`.
3. Проверить `MAX_WEBHOOK_URL`.
4. Проверить сетевой доступ сервера к MAX Bot API.
5. Проверить логи приложения.
6. Проверить доступность PostgreSQL.

При компрометации токена:

1. Отозвать старый токен в MAX.

2. Выпустить новый токен.
3. Обновить переменную `MAX_VOT_TOKEN`.
4. Перезапустить сервис.
5. Проверить webhook.

Удаление данных пользователя

По обращению пользователя администратор должен иметь возможность удалить:

- профиль пользователя;
- семейные связи;
- коды привязки;
- историю тревог, связанную с пользователем.

Перед удалением рекомендуется сохранять служебный акт обращения, если это предусмотрено политикой оператора.