

# 07\_технические\_средства\_хранения\_и\_сборки

## Технические средства хранения исходного кода, объектного кода и сборки

### Назначение документа

Документ описывает рекомендуемый контур хранения и сборки ПО “Аралтыш” для подготовки к включению в реестр российского ПО.

### Текущее состояние

Основной российский репозиторий исходного кода размещен в GitVerse:

```
https://gitverse.ru/yasg1988/araltysh
```

GitVerse используется как российский контур хранения исходного кода для подготовки к включению ПО в реестр российского ПО.

Дополнительные публичные каналы распространения проекта размещены на:

- Docker Hub;
- npm.

Docker Hub и npm используются как дополнительные публичные каналы распространения. Для целей заявления основным контуром исходного кода является GitVerse.

### Рекомендуемая целевая схема

Основной контур для реестра:

- исходный код: GitVerse <https://gitverse.ru/yasg1988/araltysh>;
- сборка: сервер в Российской Федерации 185.23.34.142;
- объектный код: архив релизного Docker-образа на сервере в Российской Федерации;
- резервные копии: российское облако или отдельный сервер в Российской Федерации.

Сервер сборки и хранения артефактов:

```
185.23.34.142
```

Хостинг-провайдер: Beget.

Локация услуги: Россия, Санкт-Петербург.

Основание: публичная информация провайдера Beget о VPS/VDS с выбором локации **Россия, Санкт-Петербург**, а также данные личного кабинета правообладателя по используемому серверу.

Справочная ссылка провайдера:

```
https://beget.com/ru/vps
```

Дополнительные публичные каналы распространения:

- Docker Hub;
- npm;

## Синхронизация с публичной витриной

Публичная страница проекта для заявления размещена на сайте правообладателя:

```
https://araltysh.yakunin-sg.ru/
```

Для заявления и экспертной проверки указывается GitVerse как основной российский репозиторий исходного кода.

## Требования к сборке

Сборка должна выполняться из исходного кода, размещенного в основном российском контуре.

Команда сборки:

```
docker build -t araltysh:X.Y.Z .
```

После сборки рекомендуется фиксировать:

- версию;
- Git commit hash;
- дату сборки;
- hostname/сервер сборки;
- checksum образа или архива.

## Техническая архитектура экземпляра

Базовая архитектура ПО:

```
MAX Bot API -> HTTPS webhook -> Nginx/reverse proxy -> Docker container -> FastAPI/Uvicorn  
-> PostgreSQL
```

Инфраструктурные программные средства:

- базовая операционная система: Linux, в том числе Astra Linux, ALT Linux, РЕД ОС или иной совместимый дистрибутив;
- контейнеризация: Docker или совместимая OCI-среда;
- веб-сервер/reverse проху: Nginx или функционально аналогичное средство публикации HTTPS;
- сервер приложений: Uvicorn ASGI server внутри контейнера;
- программная платформа: Python 3.12, FastAPI;
- СУБД: PostgreSQL;
- система виртуализации для текущего VPS: KVM;
- балансировщик нагрузки: не требуется для базового экземпляра, может использоваться при горизонтальном масштабировании;
- средства хранения данных: файловая система сервера для конфигурации/артефактов и PostgreSQL для прикладных данных.

Приложение не зависит от конкретного дистрибутива Linux при наличии Python/Docker-совместимой среды и сетевого доступа к MAX Bot API.

## Реестровая сборка версии 0.1.7

Реестровая сборка версии 0.1.7 выполнена на сервере 185.23.34.142 из исходного кода GitVerse.

Исходный код:

```
https://gitverse.ru/yasg1988/araltysh.git
```

Версия:

```
v0.1.7
```

Commit:

```
b7bdd14cfa1980ec6256a8b26a6bec70d5c0d0fd
```

Команды:

```
git clone https://gitverse.ru/yasg1988/araltysh.git source
cd source
git checkout v0.1.7
python3 -m py_compile app.py
docker build -t araltysh-registry:0.1.7 .
```

### Runtime-проверка:

```
docker run --name aralysh-registry-test --env-file <production-env-copy> -e  
    MAX_REGISTER_WEBHOOK=false -e APP_RELEASE=registry-0.1.7 -p 127.0.0.1:18016:8000  
    aralysh-registry:0.1.7  
curl http://127.0.0.1:18016/health
```

### Результат проверки:

- Docker-образ успешно собран;
- временный контейнер успешно запущен;
- /health вернул status=ok;
- webhook-регистрация при проверке была отключена;
- продакшн-сервис не изменялся.

### Артефакты сохранены на сервере:

```
/opt/aralysh-registry-artifacts/0.1.7/
```

### Файлы:

- aralysh-source-0.1.7.tar.gz;
- aralysh-registry-0.1.7-image.tar;
- SHA256SUMS.txt;
- BUILD\_PROTOCOL.md.

### SHA256:

```
ff9e921a7482a44f0cb9e44f91d280b1d1ba1c96974d17653b7144724d82e31  aralysh-source-  
0.1.7.tar.gz  
5a8c26f148c5d076c17a92e3f461434df402b84b9edc6443fff5c73ef10c40de  aralysh-registry-0.1.7-  
image.tar
```

## Регламент резервного копирования артефактов

Реестровые артефакты создаются и резервируются перед выпуском каждого релиза, который планируется использовать для подачи в реестр, эксплуатации или передачи пользователям.

Для каждого релиза сохраняются:

- архив исходного кода;
- архив контейнерного образа или иной объектный код;
- файл checksums;
- протокол сборки;
- release notes.

Основное место хранения:

```
/opt/aralysh-registry-artifacts/<version>/
```

Периодичность:

- перед каждым публичным релизом;
- перед каждой подачей или обновлением сведений в реестре;
- перед существенным обновлением production-сервиса.

Срок хранения:

- весь срок поддержки соответствующей версии;
- не менее 3 лет после вывода версии из эксплуатации.

Ответственный за хранение и резервирование: ИП Якунин Сергей Георгиевич.

Рекомендуемый порядок резервирования:

1. Сформировать артефакты релиза.
2. Рассчитать SHA256 checksums.
3. Сохранить артефакты в основном каталоге на сервере.
4. Скопировать артефакты на резервный носитель или в отдельное российское хранилище.
5. Проверить контрольные суммы после копирования.

## Что нужно заполнить перед подачей

- указать конкретное резервное хранилище, если оно потребуется при подаче.