

Генеральный директор

ООО «ТД»

Чебыкин И.А.

М.П.



**Atlas**  
программное обеспечение  
документация

**ИНСТРУКЦИЯ ПО УСТАНОВКЕ ЭКЗЕМПЛЯРА  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Дата составления: 31.03.2026

Версия документа: 1.0

## **1. Общие положения**

Настоящая инструкция описывает порядок установки экземпляра программного обеспечения Atlas на сервер под управлением Linux для целей проверки работоспособности программного обеспечения по функциональной документации.

Инструкция описывает поставочный сценарий развертывания из готового комплекта поставки. Сборка программного обеспечения из исходного кода в состав настоящей инструкции не входит.

Для варианта поставки для Российской Федерации используется классификация *ru*.

## 2. Получение дистрибутива и лицензии

Перед началом установки необходимо получить у правообладателя комплект поставки и данные лицензирования.

### 2.1. Что необходимо получить от правообладателя

До начала установки должны быть получены:

- Архив поставки вида `bundle_YYYYMMDDHHMMSS.tar.gz`.
- Архив с набором тестовых данных `test-archive.tar`
- IP-адрес или DNS-имя сервера лицензирования Guardant либо подтверждение, что сервер лицензирования должен быть развернут самостоятельно.

Если сервер лицензирования нужно развернуть самостоятельно, то дополнительно будет получен:

- При установке без доступа в сеть Интернет: установочный пакет Guardant Control Center и, при необходимости, пакет с консольной утилитой `license_wizard` для целевой архитектуры Linux.
- Серийный номер Guardant для активации лицензии.

Архив поставки должен содержать:

- Файл `docker-compose.yml`.
- Файл `.env`.
- Файл `nginx.conf`.
- Каталог `database/ru`.
- Каталог `sign_icons`.
- Каталог `guardant` с файлом `visibility.json`.
- Архив Docker-образа `master-web_<версия>.tar.gz`.
- Скрипт `import_image.sh`.
- Файл `manifest.json`.
- Файл `api_token.txt`

Если файл `visibility.json` предоставлен в готовом виде, использовать следует именно его. Если правообладатель передал только параметры для заполнения, файл `guardant/visibility.json` необходимо настроить вручную в соответствии с разделом 5 настоящей инструкции.

Если установочный пакет Guardant Control Center не был передан в составе поставки, его необходимо загрузить с официального ресурса Guardant в соответствии с разделом 5 настоящей инструкции.

## 3. Требования к программному окружению

### 3.1. Поддерживаемые операционные системы

Установить операционную систему на базе Linux. Допускается использование:

- Ubuntu 22.04 LTS.
- Astra Linux.
- RedOS.

Пошаговый пример ниже приведен для Ubuntu 22.04 LTS. Для Astra Linux и RedOS допускается использование эквивалентных пакетов и штатного менеджера пакетов дистрибутива.

Требование к окружению остается неизменным: на сервере должны быть установлены Docker Engine и Docker Compose Plugin.

### 3.2. Требования к сети

До запуска Atlas необходимо обеспечить:

- Доступ пользователей к серверу по порту 80/tcp.
- Доступ сервера Atlas к серверу лицензирования Guardant по адресу, выданному правообладателем.
- Доступ хоста Atlas к серверу лицензирования Guardant по порту 3189/tcp.
- Доступ сервера лицензирования Guardant к адресу <https://getlicense.guardant.ru> на этапе активации лицензии.

### 3.3. Установка Docker и Docker Compose Plugin

На примере Ubuntu 22.04 LTS выполнить:

#### 3.3.1. Обновление индекса пакетов и установка базовых зависимостей

```
sudo apt update  
sudo apt install -y ca-certificates curl
```

### 3.3.2. Добавление ключа и репозитория Docker

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

sudo tee /etc/apt/sources.list.d/docker.sources <<'EOF'
Types: deb
URIs: https://download.docker.com/linux/ubuntu
Suites: jammy
Components: stable
Signed-By: /etc/apt/keyrings/docker.asc
EOF

sudo apt update
```

Установка Docker Engine и Docker Compose Plugin:

```
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin
sudo systemctl enable --now docker
```

### 3.4. Проверка установки Docker

```
sudo docker run --rm hello-world
sudo docker compose version
```

## 4. Подготовка каталога установки

Архив поставки необходимо заранее скопировать на целевой сервер на электронном носителе либо получить по ссылке на файловое хранилище.

Последовательность действий:

1. Создать рабочий каталог и перейти в него:

```
mkdir -p ~/atlas-master  
cd ~/atlas-master
```

2. Скопировать в каталог архив поставки bundle\_YYYYMMDDHHMMSS.tar.gz.
3. Распаковать архив:

```
tar -xzf bundle_YYYYMMDDHHMMSS.tar.gz  
cd bundle_YYYYMMDDHHMMSS  
chmod +x ./import_image.sh
```

4. Записать api токен:

```
value=$(<api_token.txt)  
sed -i "s|^OVERPASS_API_TOKEN=.*|OVERPASS_API_TOKEN=$value|" .env
```

## 5. Импорт Docker-образа и запуск системы

### 5.1. Импорт образа приложения

В каталоге поставки выполнить:

```
sudo ./import_image.sh
```

### 5.2. Запуск системы

После успешного импорта образа выполнить:

```
sudo docker compose up -d
```

Система запускает следующие сервисы:

- postgres
- minio
- minio-bucket-init
- web
- nginx

Контейнер minio-bucket-init является служебным и после завершения инициализации может находиться в состоянии Exited (0). Это является штатным поведением.

Во внешнюю сеть публикуется веб-интерфейс через контейнер nginx на порту 80/tcp. Контейнеры postgres и minio используются как внутренние сервисы стека и по умолчанию наружу не публикуются.

## 6. Проверка работоспособности

### 6.1. Проверка состояния контейнеров

Проверить состояние контейнеров можно командой:

```
sudo docker compose ps
```

Ожидаемый результат:

- Контейнеры postgres, minio, web, nginx находятся в состоянии Up.
- Контейнер minio-bucket-init находится в состоянии Exited (0) либо отсутствует в списке активных контейнеров после завершения инициализации.

### 6.2. Проверка доступности веб-интерфейса

Открыть в браузере адрес:

```
http://<адрес_сервера>/
```

Если установка выполняется локально на самом сервере, допускается использовать:

```
http://127.0.0.1/
```

### 6.3. Проверка доступности API

Проверить ответ API:

```
curl http://127.0.0.1/api/check/travels
```

Успешным результатом считается получение ответа от сервиса без ошибки соединения.

### 6.4. Проверка журналов

При необходимости проверить журналы контейнеров:

```
sudo docker compose logs --tail=100
```

## 7. Выгрузка тестового проезда

1. Перейти в директорию с полученным архивом тестовых данных, либо копировать его в рабочую директорию.
2. С помощью команды curl выполнить запрос на выгрузку полученного архива:

```
curl -v -X POST "http://<адрес_сервера>/api/edge/passportdata/archive/" -H "Content-Type: application/x-tar" --data-binary "@./test-archive.tar"
```

При успешном завершении команды, можно перейти через браузер по адресу [http://<адрес\\_сервера>/travels/](http://<адрес_сервера>/travels/) и начать работу с обработанным проездом.

Запрос должен быть отправлен из той же директории, где находится архив с данными, либо изменить путь в команде на актуальный.