

Инструкция по установке Программного обеспечения

**«Интеллектуальная производственная система ОАИ
Livestock**

**для оптимизации кормления, кормового поведения и
физиологического состояния коров с применением
компьютерного зрения и идентификацией
для молочно-товарных комплексов»**

г. Уфа, 2026

Список обозначений и сокращений

Сокращение	Расшифровка
БД	база данных
ВП	веб-приложение
НСМ	нейросетевые модули
ОЗУ	оперативное запоминающее устройство
ПАК	программно-аппаратный комплекс
ПК	персональный компьютер
ПО	программное обеспечение
СУБД	система управления базами данных
ЦПУ	центральный процессор
JDK	Java Development Kit

Архитектура

Кратко архитектура описана на рисунке 1.

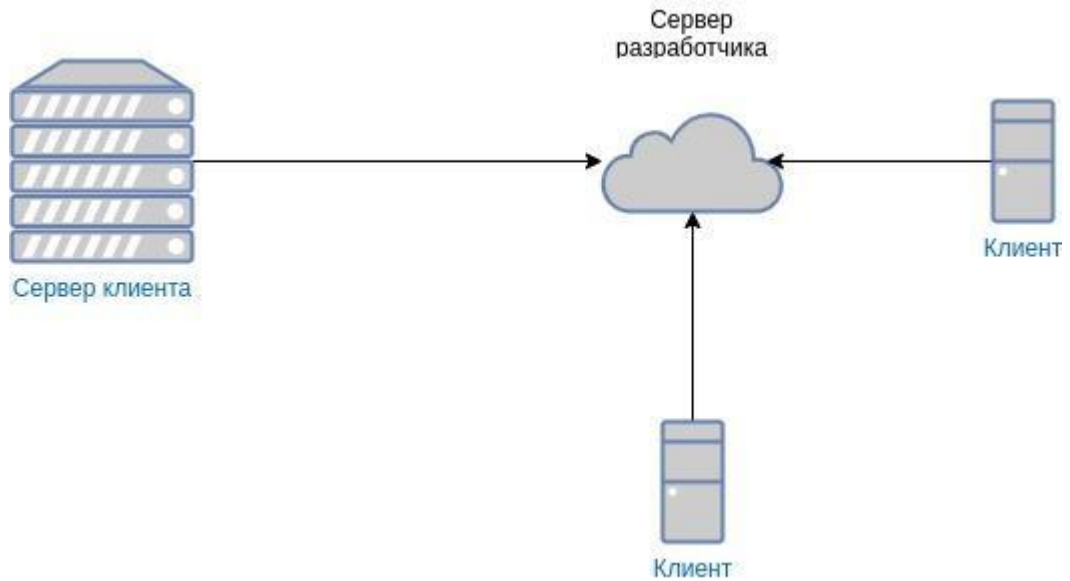


Рисунок 1 – Краткая архитектура системы

ПО ПАК состоит из:

- серверной части:
- сервер клиента (например, МТК Калинин);
- сервер разработчика;
- клиентской части.

К серверной и клиентской частям предъявляются разные требования. Подробнее архитектура описана в документе «Архитектура системы».

Требования к аппаратному обеспечению

Требования к серверной части:

- минимальный объём оперативной памяти: 4 ГБ;
- минимальные требования к ЦПУ: 2 ядра, Intel Core i7-12700 2,1/4,9 ГГц либо аналогичный;
- минимальные требования к графическому процессору: PCI-E MSI GeForce RTX 4060 либо аналогичная;
- необходимо 150 ГБ на жёстком диске (для хранения изображений сегментации и детекции за 2 недели).

Требования к клиентской части:

- процессор: с частотой 1 ГГц (или выше);
- оперативная память: минимум 512 МБ;
- свободное место на диске: минимум 600 МБ.

Требования к программному окружению

Требования к серверной части:

1. Операционная система (ОС):
 - рекомендуемая, предпочтительно семейства Debian (например, Debian 12 / Ubuntu 24.04);
 - любая ОС, поддерживающая JDK версии 17.
2. JDK версии 17.
3. PostgreSQL, рекомендуемая версия 17.

Требования к клиентской части:

Веб-браузер с включённым JavaScript:

- Google Chrome версии 120 и выше;
- Mozilla Firefox версии 115 и выше;
- Яндекс-браузер версии 23.10 и выше.

Механизм развертывания системы «OAI Livestock»

Клиент, приобретая права на OAI Livestock, получает право, чтобы разработчик развернул ПАК на сервере клиента. Описание механизма развёртывания приведено для ознакомления с процессом. Подробное описание механизма развёртывания системы приведена для ОС Ubuntu 24.04. Для других ОС, особенно не Linux (например, Windows/MacOS), развёртывание системы может отличаться.

Для установки ПАК необходим запуск скрипта `install.sh`. Текст скрипта приведён в приложении 1 для справки. Помимо скрипта есть другие папки и файлы, которые используются для развёртывания системы:

- `cownewarch` – папка с HCM;
- `sdm` – папка со скриптами для синхронизации изображений системы;
- `env` – папка со службой;
- `cow-backend` – папка с ВП.

Также важно учесть, что система разворачивается на сервере клиента, например МТК Калинин. На сервере разработчика система уже развёрнута под конкретного клиента.

Подразумевается, что в ОС создан пользователь `deployer` как на сервере клиента, так и на сервере разработчика.

Создать необходимые данные в ВП

Для того чтобы приложение заработало, необходимо создать кормовые столы, секции, камеры.

Для начала в локальной сети клиента в любом браузере, удовлетворяющем требованиям, необходимо перейти по адресу `http://localhost:8080`.

Далее авторизоваться как администратор. Логин: `admin`, пароль: `admin`. После первоначальной настройки клиентом устанавливается свой логин и пароль.

Создание кормового стола

Перейти в раздел «Кормовые столы». Нажать на кнопку «Добавить кормовой стол» (рисунок 2).

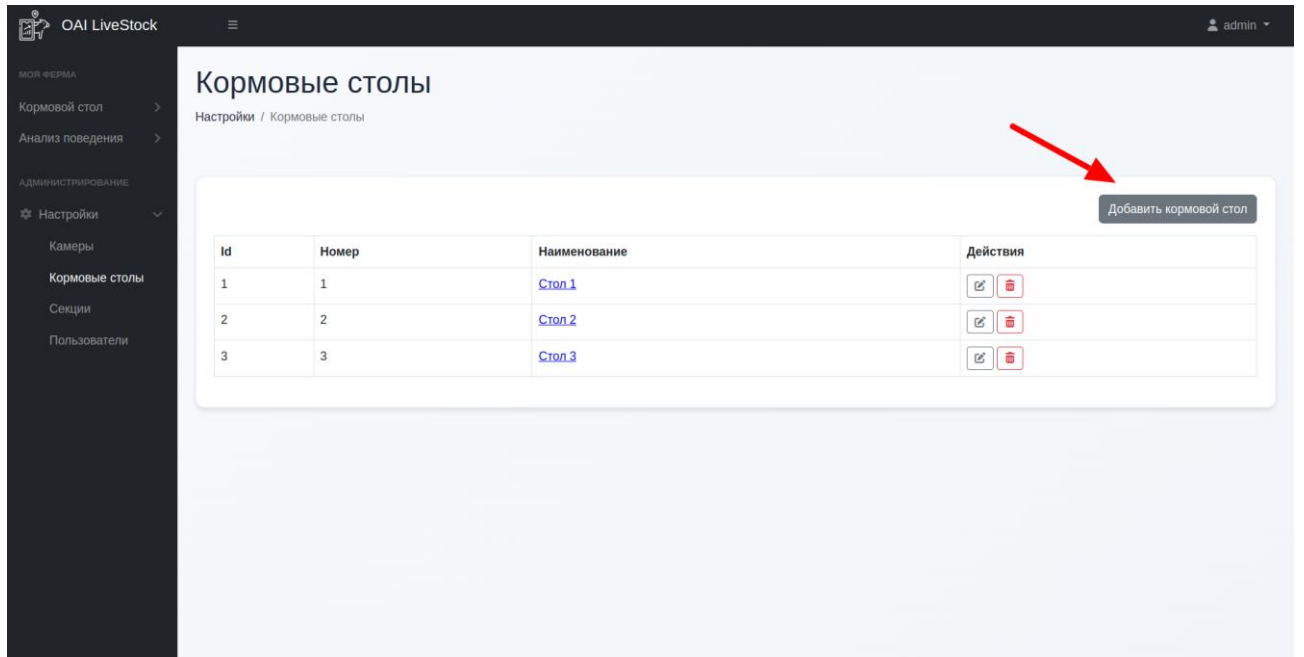


Рисунок 2 – Список кормовых столов

В появившейся форме ввести название кормового стола (рисунок 3) и нажать кнопку «Сохранить».

Добавление кормового стола

Настройки / [Кормовые столы](#) / Новый

Номер:

Префикс наименования:

Не обязательное. Будет использоваться при выводе объектов: Стол 1, Стол 2

Рисунок 3 – Форма добавления кормового стола

Создание секции

Зайти в раздел «Секции» и нажать на кнопку «Добавить секцию» (рисунок 4).

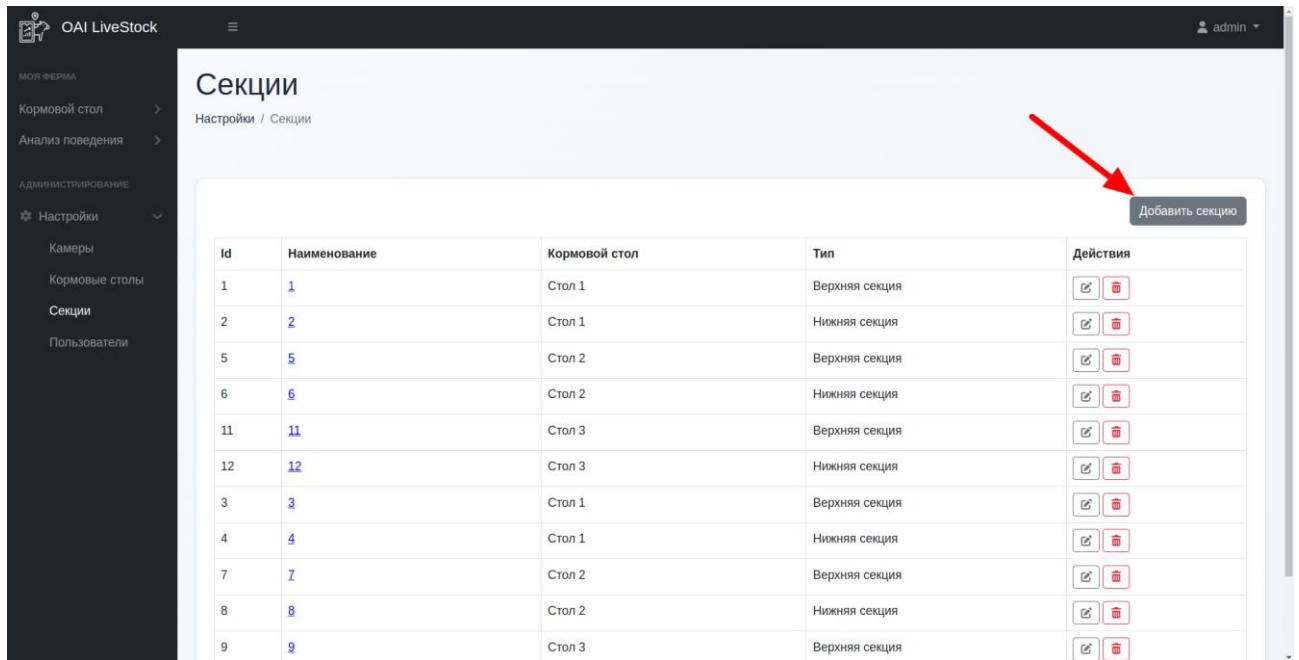


Рисунок 4 – Список секций

В появившейся форме ввести название секции, выбрать тип секции (верхняя либо нижняя) и кормовой стол, затем нажать «Сохранить» (рисунок 5).

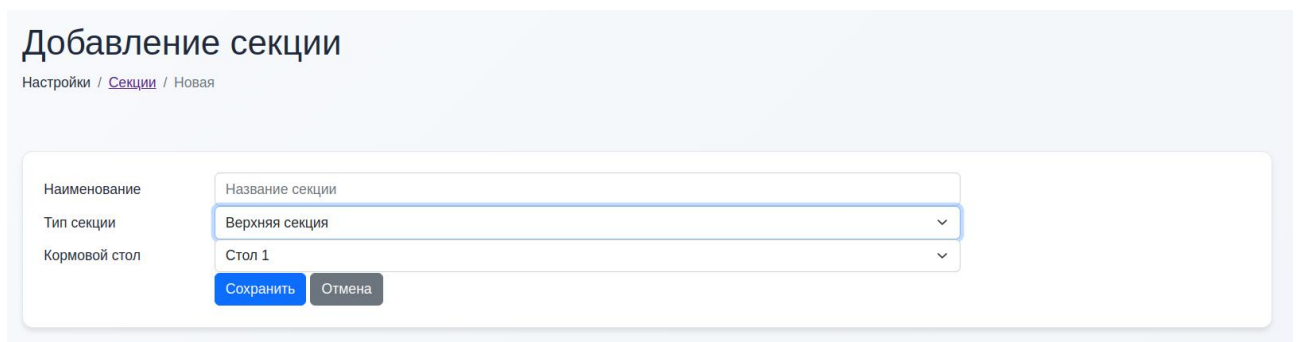


Рисунок 5 – Форма добавления секции

Создание камеры

Зайти в раздел «Камеры», нажать на кнопку «Добавить камеру» (рисунок 6).

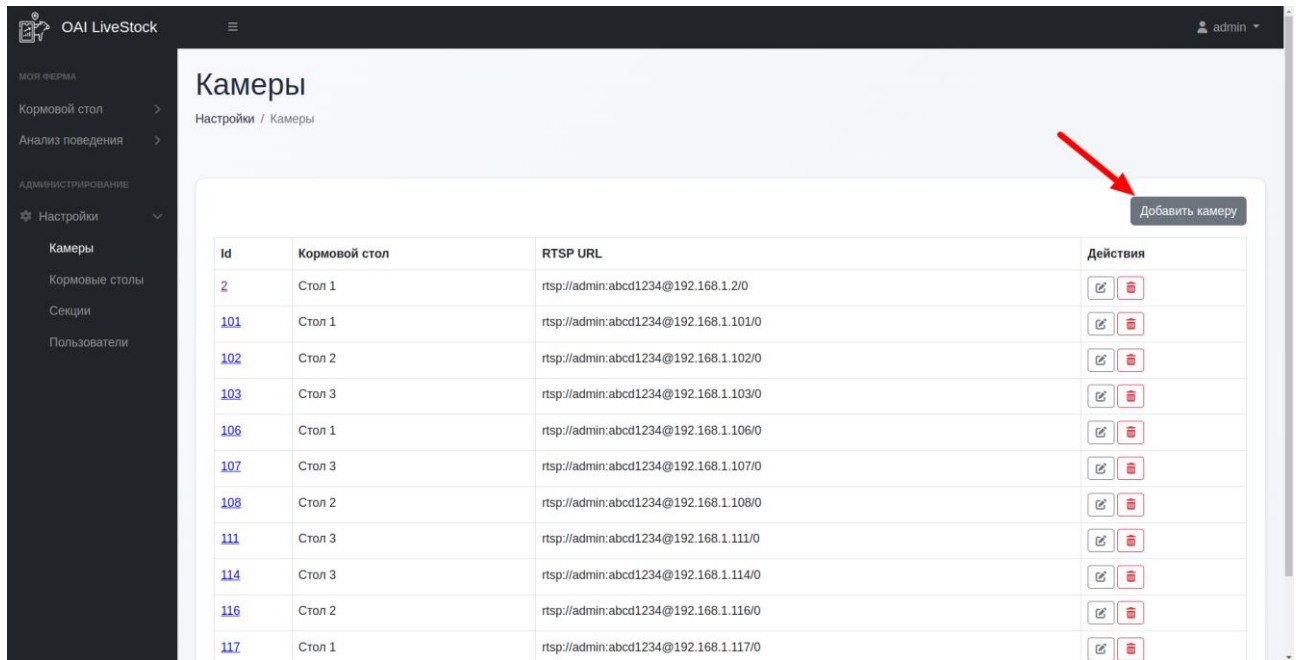


Рисунок 6 – Список камер

В появившейся форме необходимо ввести следующие данные (рисунок 7) и нажать на кнопку «Сохранить»:

- уникальный ID камеры;
- RTSP-URL камеры;
- коэффициент перевода px в метры;
- файл с параметрами калибровки (чаще всего необходимо указать tiandy.json);
- выбрать из выпадающего списка подходящий кормовой стол;
- выбрать верхнюю секцию из выпадающего списка;
- выбрать нижнюю секцию из выпадающего списка;
- включить оба модуля: модуль сегментации и модуль детекции.

Добавить камеру

Настройки / [Камеры](#) / Новая

Id	<input type="text" value="Номер камеры"/>
Rtsp url	<input type="text" value="rtsp://..."/>
Px to meters	<input type="text" value="119"/>
Calibration params	<input type="text" value="Файл конфигурации"/>
Кормовой стол	<input type="text" value="Стол 1"/> ▾
Верхняя секция	<input type="text" value="1"/> ▾
Нижняя секция	<input type="text" value="2"/> ▾
Модули	<input checked="" type="checkbox"/> Модуль сегментации <input checked="" type="checkbox"/> Модуль детекции
	<input type="button" value="Сохранить"/> <input type="button" value="Отмена"/>

Рисунок 7 – Форма добавления камеры

Примечание перед подключением к серверу

OAI Livestock получает данные для анализа от серверов клиентов, установленных на фермах. Из-за возможных перебоев интернет-связи на фермах синхронизация данных может происходить с задержкой, что временно влияет на актуальность информации в системе.

Запуск программы «OAI Livestock»

Подключение к серверу

Для подключения к серверу, где размещена система «OAI Livestock», необходимо воспользоваться командной строкой (терминалом).

Для открытия командной строки в ОС Windows на клавиатуре нажмите сочетание клавиш «Win + R». В открывшемся окне введите команду cmd и нажмите «Enter». Далее необходимо ввести команду:

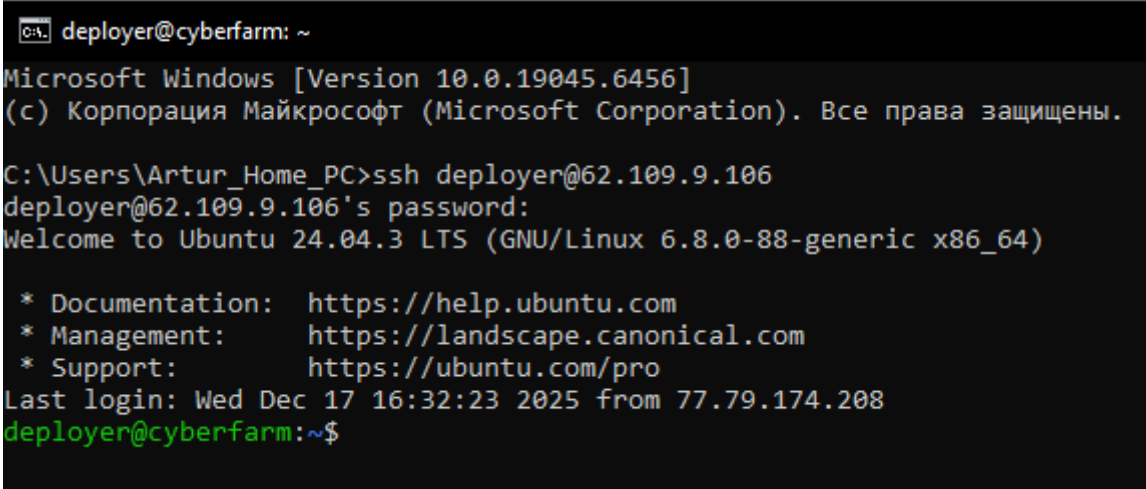
```
ssh deployer@62.109.9.106
```

и нажать «Enter». После ввода команды появится запрос:

```
deployer@62.109.9.106's password:
```

Введите пароль: (при вводе пароля символы не отображаются на экране).

После успешного подключения к серверу отобразится приглашение ОС Ubuntu (рисунок 8).

A screenshot of a terminal window on a Windows machine. The prompt is 'C:\>'. The user enters 'ssh deployer@62.109.9.106'. The terminal shows the connection process, including the password prompt and the Ubuntu login banner. The banner includes the Ubuntu version (24.04.3 LTS), the kernel version (6.8.0-88-generic x86_64), and links for documentation, management, and support. The last login information is also displayed. The prompt changes to 'deployer@cyberfarm:~\$' after a successful login.

```
deployer@cyberfarm: ~
Microsoft Windows [Version 10.0.19045.6456]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Artur_Home_PC>ssh deployer@62.109.9.106
deployer@62.109.9.106's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
Last login: Wed Dec 17 16:32:23 2025 from 77.79.174.208
deployer@cyberfarm:~$
```

Рисунок 8 – Подключение к серверу

Запуск и останов программы «OAI Livestock»

Для запуска программы «OAI Livestock» необходимо ввести следующую команду:

```
sudo systemctl start OAI
```

Для прекращения работы программы:

```
sudo systemctl stop OAI
```

Для просмотра статуса программы:

```
sudo systemctl status OAI
```

Вход в систему

Для экспертной оценки «OAI Livestock» можно воспользоваться уже развернутой системой.

Перед началом работы необходимо загрузить браузер, например, Яндекс Браузер. В адресной строке браузера ввести адрес сайта, по которому расположена программа «OAI Livestock»: <https://cyberfarm.su/>.

Вход осуществляется по логину и паролю (рисунок 9):

Логин и пароль предоставляются разработчиком клиенту



The image shows a login interface with a light gray background. At the top, the text 'Please sign in' is displayed in a bold, black font. Below this, there are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. Both fields have a thin black border. At the bottom of the form is a prominent blue button with the text 'Sign in' in white, centered on the button.

Рисунок 9 – Вход в систему

После успешного входа можно приступить к работе в программе «OAI Livestock».

Приложение 1. Текст скрипта установки

```
# Обновление системы
```

```
sudo apt update && sudo apt upgrade -y
```

```
# Установка зависимостей
```

```
sudo apt install -y ca-certificates curl gnupg lsb-release
```

```
# Добавление официального ключа и репозитория Docker
```

```
sudo mkdir -p /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
```

```
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
```

```
/etc/apt/sources.list.d/docker.list > /dev/null
```

```
# Установка Docker Engine, Compose (плагин)
```

```
sudo apt update
```

```
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

```
# Добавление текущего пользователя в группу docker (чтобы не использовать sudo)
```

```
sudo usermod -aG docker $USER
```

```
newgrp docker
```

```
# Установка драйверов NVIDIA (рекомендуется проприетарный)
```

```
sudo apt install -y nvidia-driver-550
```

```
# Установка CUDA Toolkit (через network installer)
```

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64/cuda-  
keyring_1.1-1_all.deb
```

```
sudo dpkg -i cuda-keyring_1.1-1_all.deb
```

```
sudo apt update
```

```
sudo apt install -y cuda-toolkit-12-6
```

```
# Установка NVIDIA Container Toolkit (для работы GPU в Docker)
```

```
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o  
/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
```

```
curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-toolkit.list | sed  
's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg]
```

```
https://#g' | sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
```

```
sudo apt update
```

```
sudo apt install -y nvidia-container-toolkit
sudo nvidia-ctl runtime configure --runtime=docker
sudo systemctl restart docker
```

```
# Установка Git
```

```
sudo apt install -y git
```

```
# Настройка Git (рекомендуемые параметры)
```

```
git config --global core.autocrlf input
```

```
# Установка Git LFS
```

```
curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | sudo bash
```

```
sudo apt install -y git-lfs
```

```
git lfs install
```

```
# Установка JDK
```

```
sudo apt install -y openjdk-17-jdk
```

```
echo "export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64" | sudo tee -a
/etc/environment
```

```
# Загрузка и загрузка образа Triton
```

```
wget https://cloud.predictor.su/index.php/s/gzMRfMHaEsEr426
```

```
docker load -i tritonserver-custom-25.03.tar
```

```
# Сделать файлы исполняемыми
```

```
chmod +x start.sh stop.sh
```

```
# Собрать всё кроме тритона
```

```
./start.sh --build all
```

```
# Собрать тритон
```

```
./start.sh --build triton
```

```
# Запустить HCM, PostgreSQL и Redis
```

```
./start.sh
```

```
# Копирование и запуск службы OAI
```

```
sudo cp env/oai.service /etc/systemd/system/OAI.service
```

```
sudo systemctl enable OAI.service
```

```
sudo systemctl start OAI.service
```

```
# Настройка PostgreSQL для репликации
```

```
grep -q "^wal_level" /var/lib/postgresql/data/postgresql.conf || echo "wal_level = logical" | sudo
tee -a /var/lib/postgresql/data/postgresql.conf
sudo bash -c "echo 'host replication repl_user 192.168.0.0/24 md5' >>
/var/lib/postgresql/data/pg_hba.conf" && sudo systemctl restart postgresql
```

```
# Создать пользователя для репликации
```

```
docker exec postgres_db psql -U postgres -d analytics -c "CREATE ROLE repl_user WITH
REPLICATION LOGIN PASSWORD 'f#sdD_3j)';"
```

```
# Создать публикацию на мастере
```

```
docker exec postgres_db psql -U postgres -d analytics -c "CREATE PUBLICATION my_pub
FOR ALL TABLES;"
```

```
# Запустить службу синхронизации изображений
```

```
chmod +x sdm/sdm_install.sh
```

```
./sdm_install.sh
```

```
sudo systemctl enable sdm.service
```

```
sudo systemctl start sdm.service
```