**Руководство по развертыванию**

**Системы контроля и выдачи заданий Odinn**

**Версия 0.0.1 (РУС)**

Листов 17

Москва 2025

# Оглавление

[**Оглавление 2**](#_ifmqw59jciih)

[**1. Глоссарий 4**](#_q1f36laedurn)

[1.1. Термины и определения 4](#_4482z8mkeisx)

[1.2. Обозначения и сокращения 5](#_9z1bfe9rt749)

[**2. Назначение документа 6**](#_um8p193nc60y)

[**3. Состав продукта 7**](#_binylm1wmlbq)

[3.1. Компоненты системы 7](#_6hktm33pkeb0)

[3.1.1. Odinn ID 7](#_jwvo7mfz1klc)

[3.1.2. Odinn 7](#_iifoo8l5el41)

[3.1.3. Odinn ML 7](#_25d4md8kqxgr)

[**4. Требования к окружению 8**](#_onmehdgn7hw)

[4.1. Аппаратные требования 8](#_f50fzsmeuwix)

[4.2. Программные требования 8](#_9zsxmetvqdvm)

[4.2.1. Поддерживаемые операционные системы 8](#_40qgy5z21qar)

[4.2.2. Дополнительное программное обеспечение 8](#_ts66mg39kukf)

[**5. Развертывание в типовой конфигурации 10**](#_guutws5uo3au)

[5.1. Подготовка окружения 10](#_3wfffwrh4dla)

[5.1.1. Установка Docker 10](#_nup2y7uh05kr)

[5.1.2. Подключение приватного репозитория Docker 12](#_rrahsopejbt4)

[5.1.3. Установка и настройка Git 12](#_8jg04o6tzs56)

[5.2. Установка и настройка основных компонентов 12](#_k1ak5df8sazh)

[5.2.1. Установка Odinn 12](#_uqohu8ls9noy)

[5.2.2. Установка Odinn ID 15](#_l55k56gh2pn0)

[5.2.3. Установка Odinn ML 16](#_rziohy3g8img)

[5.2.4. Проверка работоспособности 17](#_6rpx75ofjgg)

[**6. Заключение 17**](#_s81ilpqv80hi)

# Глоссарий

## Термины и определения

| Термин | Определение |
| --- | --- |
| Docker | Платформа контейнеризации для развертывания приложений в изолированных средах (контейнерах) |
| Docker Compose | Инструмент для оркестрации многоконтейнерных приложений через YAML-файлы конфигурации |
| Контейнер | Легковесная изолированная среда выполнения приложения со всеми зависимостями |
| Netbox | Веб-приложение для IP-менеджмента и документирования сетевой инфраструктуры |
| API Token | Уникальный ключ аутентификации для доступа к программным интерфейсам системы |
| Standalone- инсталляция | Вариант развертывания с размещением всех компонентов на одном сервере |
| Распределенная инсталляция | Архитектура с горизонтальным масштабированием компонентов на несколько серверов |

## Обозначения и сокращения

| **NMS** | Network Management System |
| --- | --- |
| **API** | Application Programming Interface |
| **URL** | Uniform Resource Locator |
| **IP** | Internet Protocol |
| **SNMP** | Simple Network Management Protocol |
| **SSH** | Secure Shell |
| **HTTP/HTTPS** | HyperText Transfer Protocol/Secure |
| **LTS** | Long-Term Support |
| **vCPU** | Virtual CPU |
| **NTP** | Network Time Protocol |
| **CLI** | Command Line Interface |
| **GUI** | Graphical User Interface |

# Назначение документа

Настоящий документ охватывает ключевые аспекты, включая состав дистрибутива, требования к целевому окружению, а также пошаговое руководство по установке, первичной настройке и проверке работоспособности системы Odinn.

# Состав продукта

## Компоненты системы

Система Odinn состоит из нескольких компонентов, каждый из которых выполняет определенную функцию. Ниже приведен перечень основных компонентов системы и их краткое описание:

## Odinn ID

Odinn ID является основной платформой для хранения данных пользователей, их аутентификации и авторизации. Этот компонент отвечает также за аутентификацию сервисов при межсервисном взаимодействии.

## Odinn

Odinn — это ядро системы, осуществляющее выдачу заданий, контроль их выполнения.

## Odinn ML

Odinn отвечает за хранение, выгрузку, обработку данных машинного обучения. Если вы не планируете использование поддтверждения с помощью ИИ, данный компонент не обязателен..

# Требования к окружению

Для успешного развертывания и работы системы необходимо обеспечить соответствие целевого окружения следующим требованиям.

## Аппаратные требования

Аппаратные требования для развертывания системы:

* Оперативная память (RAM): 8 ГБ.
* Процессор (vCPU): 4 виртуальных ядра.
* Жесткий диск (SSD): 100 ГБ свободного пространства.

## Программные требования

### Поддерживаемые операционные системы

Поддерживается установка на следующих операционных системах

Linux:

* + - * Ubuntu 20.04 LTS или выше.
      * CentOS 7 или выше.
      * Debian 10 или выше.
      * Astra Linux SE (версия 1.7 или выше).
      * Другие дистрибутивы Linux, поддерживающие Docker, также могут быть использованы, но требуют дополнительной проверки на совместимость.

Рекомендации по выбору дистрибутива:

* + - * Использование LTS-версий дистрибутивов предпочтительно для обеспечения стабильности и долгосрочной поддержки.
      * Необходимо установить последние обновления безопасности и исправления для операционной системы.

### Дополнительное программное обеспечение

Для корректной работы системы Odinn требуется установка и

настройка следующего дополнительного программного обеспечения:

* + - * **Среда контейнеризации Докер (Docker)**
        + Минимальная версия Docker: **20.10.x** и выше.
        + Требуется установка **Docker Compose** (версия 2.x и выше) для управления контейнерами.
      * **Git**
        + Для работы с репозиториями и управления версиями конфигураций рекомендуется установка системы контроля версий Git.
        + Минимальная версия Git: 2.35.x и выше.
      * **База Данных (Postgres 16)**
      * **Брокер сообщений (RabbitMQ)**
      * **Кеш (Redis)**
      * **Сетевые порты**
        + Наличие статического IP-адреса для сервера, на котором развертывается система.
      * **Открытые порты для работы системы:**
        + **443/TCP** (HTTPS) — для защищенного доступа к веб- интерфейсу.
        + **80/TCP** (HTTP) — для доступа к веб- интерфейсу (при отсутствии HTTPS).
        + **22/TCP** (SSH) — для удаленного управления сервером.

# Развертывание в типовой конфигурации

## Подготовка окружения

Данная инструкция подготовлена для дистрибутивов Linux на основе Debian/Ubuntu. Для других дистрибутивов Linux процесс установки и настройки может отличаться. В таких случаях рекомендуется обратиться к официальной документации используемого дистрибутива.

Обязательным условием для инсталляции Odinn является предварительная настройка рабочего окружения с установкой всех необходимых программных компонентов. В данном разделе описаны процессы установки Docker, входа в репозиторий Docker, а также установки и настройки Git.

**Примечание:**

Перед началом установки необходимо запросить учетные данные для входа в приватный репозиторий для доступа к образам Docker контейнеров, а также системы хранения версий Git, обратитесь к вендору для получения доступа.

### Установка Docker

Для установки Docker выполните следующие шаги:

Перед установкой Docker рекомендуется обновить пакеты системы:

sudo apt update && sudo apt upgrade -y

Так же рекомендуется удалить все возможные конфликтующие пакеты:



Установка необходимых пакетов для работы Docker и добавление официального GPG-ключа Docker производиться следующими командами:



Обновите пакеты и установите Docker:



Для управления контейнерами Docker без использования sudo можно добавить текущего пользователя в группу docker. Выполните следующую команду:

sudo usermod -aG docker $USER

После этого требуется перезапустить сеанс пользователя (выйти из системы и войти заново либо выполнить newgrp docker), чтобы изменения вступили в силу.

Для проверки корректности установки Docker запустите тестовый контейнер *hello-world:*

sudo docker run hello-world

Если Docker установлен корректно, вы увидите сообщение с подтверждением успешного запуска контейнера.

С подробной инструкцией по установке Docker можно ознакомиться в официальной документации: <https://docs.docker.com/engine/install/>.

### Подключение приватного репозитория Docker

Контейнеры системы Odinn размещены в приватном Docker- репозитории. Для доступа к ним необходимо выполнить вход в этотрепозиторий.

Используйте команду docker login для авторизации:

docker login ansgar.gitlab.yandexcloud.net:5050

При запросе введите имя пользователя и пароль, предоставленные для доступа к приватному репозиторию. После успешной авторизации вы увидите сообщение *Login Succeeded.*

### Установка и настройка Git

Для установки Git выполните следующие шаги:

Обновите пакеты и установите Git с помощью пакетного менеджера:



Проверьте версию Git для подтверждения успешной установки:

git --version

Вы должны увидеть версию установленного Git (например, git version 2.35.1).

Подробная инструкция по установке и настройке Git доступна в официальной документации: <https://git-scm.com/book/ru/v2>.

**Примечание:**

Если подключение к Git-репозиторию недоступно, развертывание системы можно выполнить, используя локальный дистрибьютив. Для его получения системы обратитесь к производителю или технической поддержке.

## Установка и настройка основных компонентов

Установка системы выполняется с использованием Docker и Docker Compose. Процесс включает редактирование конфигурационных файлов и запуск контейнеров. Ниже приведена пошаговая инструкция.

### Установка Odinn

Создайте директорию odinn и перейдите в неё.

| mkdir odinn cd odinn |
| --- |

Создайте файл docker-compose.yml

| nano docker-compose.yml |
| --- |

Пример содержимого файла (Версию образа замените на рекомендованную вендором):

| services:  app: &app-service-template  image: ansgar.gitlab.yandexcloud.net:5050/odinn/odinn:1.0.4  env\_file:  - .env  command: bash -c "  if [ ! -d /var/www/vendor ] ; then composer install --no-interaction ; fi &&  if [ ! -d /var/www/node\_modules ] ; then npm install && npm run build ; fi &&  if [ ! -d /var/www/public/storage ] ; then php /var/www/artisan storage:link ; fi &&  php /var/www/artisan octane:start --no-interaction --server=roadrunner --host=0.0.0.0 --rpc-port=6002 --port=8000  "  volumes:  - storage\_data:/var/www/storage  - public\_data:/var/www/public  - ./logs:/var/www/storage/logs  - type: bind  source: /home/forseti/odinn/.rr.yaml  target: /var/www/.rr.yaml  #- .env:/var/www/.env  working\_dir: /var/www  restart: unless-stopped  networks: [ odinn\_network ]  extra\_hosts:  - 'host.docker.internal:host-gateway'  deploy:  resources:  limits:  memory: 1024M   queue:  <<: \*app-service-template  command: bash -c "  php /var/www/artisan rabbitmq:queue-declare odinn-app &&  php /var/www/artisan queue:work --timeout=0"  networks: [ odinn\_network, internal]  volumes: []  scale: 3  deploy:  resources:  limits:  memory: 1024M   scheduler:  <<: \*app-service-template  networks: [ odinn\_network]  command: php /var/www/artisan schedule:work  deploy:  resources:  limits:  memory: 256M   # Reverb  websockets:  <<: \*app-service-template  # Debug  command: php /var/www/artisan reverb:start --verbose --host=0.0.0.0 --port=8080  healthcheck:  test: [ 'CMD', 'curl', 'http://localhost:8080/app/${REVERB\_APP\_KEY}' ]  interval: 30s  timeout: 5s  retries: 5  networks: [ odinn\_network ]  depends\_on:  - queue  deploy:  resources:  limits:  memory: 1024M   # lora:  # <<: \*app-service-template  # command: php /var/www/artisan lora:start   nginx:  image: ansgar.gitlab.yandexcloud.net:5050/odinn/odinn/nginx:main  ports:  - "8080:80"  depends\_on:  - app  volumes:  - storage\_data:/var/www/storage  - public\_data:/var/www/public  networks: [ odinn\_network ]  restart: unless-stopped  networks:  odinn\_network:  name: odinn\_network  volumes:  storage\_data:  public\_data: |
| --- |

Создайте файл .env и заполните его необходимыми переменными окружения.

Запустите контейнеры:

| docker compose up -d |
| --- |

Запустите необходимые миграции БД:

| docker compose exec app php artisan migrate |
| --- |

### Установка Odinn ID

Создайте директорию users и перейдите в неё.

| mkdir users cd users |
| --- |

Создайте файл docker-compose.yml

| nano docker-compose.yml |
| --- |

Пример содержимого файла (Версию образа замените на рекомендованную вендором):

| services:  app\_users: &app-service-template  image: ansgar.gitlab.yandexcloud.net:5050/odinn/users:1.0.1  container\_name: app\_users  working\_dir: /var/www  restart: unless-stopped  ports:  - "8081:8080"  extra\_hosts:  - 'host.docker.internal:host-gateway'  env\_file:  - .env  networks:  - odinn\_network  users\_scheduler:  <<: \*app-service-template  ports: []  container\_name: app\_users\_scheduler  command: php /var/www/artisan schedule:work  networks:  odinn\_network:  name: odinn\_network  external: true |
| --- |

Создайте файл .env и заполните его необходимыми переменными окружения.

Запустите контейнеры:

| docker compose up -d |
| --- |

Запустите необходимые миграции БД:

| docker compose exec app\_users php artisan migrate --seed |
| --- |

### Установка Odinn ML

Создайте директорию ml и перейдите в неё.

| mkdir ml cd ml |
| --- |

Создайте файл docker-compose.yml

| nano docker-compose.yml |
| --- |

Пример содержимого файла (Версию образа замените на рекомендованную вендором):

| services:  app\_ml:  container\_name: app\_ml  image: ansgar.gitlab.yandexcloud.net:5050/odinn/ml:prod  working\_dir: /var/www  restart: unless-stopped  volumes:  - .env:/var/www/.env:ro  - ./storage:/var/www/storage  networks:  - odinn\_network  extra\_hosts:  - 'host.docker.internal:host-gateway' networks:  odinn\_network:  name: odinn\_network  external: true |
| --- |

Создайте файл .env и заполните его необходимыми переменными окружения.

Запустите контейнеры:

| docker compose up -d |
| --- |

Запустите необходимые миграции БД:

| docker compose exec app\_ml php artisan migrate --seed |
| --- |

### Проверка работоспособности

Проверьте доступность веб-интерфейса:

Откройте браузер и перейдите по URL, указанному в файле .env (например, http://localhost). Убедитесь, что веб-интерфейс системы доступен.

# Заключение

Настоящая инструкция содержит руководство по развертыванию системы Odinn. В документе рассмотрены:

Требования к окружению – минимальные версии ПО, аппаратные и программные предпосылки для корректной работы системы.

Подготовка окружения – установка Docker, настройка приватного репозитория, развертывание Git.

Установка и настройка компонентов – загрузка контейнеров, конфигурация переменных окружения, запуск системы.

После выполнения всех этапов система готова к эксплуатации. Для дальнейшей настройки рекомендуется обратиться к официальной документации или в службу технической поддержки.