

**Руководство администратора ПО
«Объектное хранилище»**

1.Введение

1.1. Область применения

Настоящий документ предназначен для сотрудников эксплуатирующей организации и отражает основные функциональные возможности и порядок действий при выполнении операций, связанных с администрированием программного обеспечения «Объектное хранилище» (далее - «Система»)

1.2. Перечень выполняемых функций администратора/оператора

В перечень выполняемых функций администратора Системы входят:

- Установка и настройка Системы
- Реализация планов устранения сбоев и нетиповых нештатных ситуаций
- Выполнение сбора и предоставление в вышестоящую линию технической поддержки информации для воспроизведения технических проблем и выработки решений по их разрешению
- Реализация рекомендаций по устранению нештатных ситуаций, полученных с вышестоящей линии поддержки
- Восстановление работоспособности Системы при сбоях в работе функциональных модулей
- Разработка решения по устранению технических проблем в работе функциональных модулей

1.3. Уровень подготовки администратора/оператора

Администратор/оператор (далее по тексту Администратор) Системы должен уметь пользоваться и настраивать среду функционирования контейнеров или систему оркестрации, используемую на предприятии.

Рекомендуемая численность персонала для эксплуатации Системы — 1 штатная единица.

Администраторы Системы должны пройти обязательную общую и специальную подготовку для работы с Системой.

Общая подготовка должна включать в себя получение знаний и навыков работы с Системой в качестве администратора.

Специальная подготовка должна включать в себя получение знаний и навыков в объеме, необходимом для выполнения своих должностных обязанностей

1.4. Перечень документации

В состав документации, с которой необходимо ознакомиться администратору Системы входят:

- описание функциональных характеристик Системы
- описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения.

2. Установка Системы

В данном разделе будет описана установка Системы на Debian Linux. Предполагается, что были предварительно установлены также Docker, Docker Compose.

2.1. Системные требования к ПО

Минимальные аппаратные требования:

- Операционная система, способная запускать контейнеры. Предпочтительно Linux.
- Система управления контейнерной виртуализацией. Предпочтительно Docker Swarm или Kubernetes.
- Количество логических ядер процессора: 4
- Семейство процессоров: x86
- Частота процессора: 3.0. ГГц
- Объем установленной памяти: 16 Гб

2.1.2. Минимальные требования к сторонним компонентам и/или системам, необходимым для установки и работы ПО

- Debian 11 (Открытая лицензия GNU)
- Docker 24.0.2 (open-source community edition)

2.1.3. Языки программирования

При разработке Системы был использован язык программирования GoLang 1.22 (открытая лицензия BSD)

2.2. Порядок установки

1. Создайте папку /root/pudgeio
2. Смонтируйте диск с дистрибутивом в папку /mnt
3. Скопируйте из дистрибутива исходники из папки /mnt в папку /root/pudgeio
4. Смените текущую папку на /root/pudgeio
5. Отредактируйте файл docker-compose.yml, в соответствии с пунктом 3.2 данного документа
6. Выполните команду
`docker compose -up -d --build`

3. Настройка Системы

3.1. Общие сведения

В данном документе приводятся примеры настройки Системы с использованием среды Docker Compose. Настройка операционной системы и возможная настройка использования систем оркестрации, находятся вне компетенции этого документа и не будут тут описаны.

3.2. Конфигурируемые параметры

Для корректной работы модуля приема запросов, необходимо настроить для него следующие переменные окружения:

- HOST - адрес сервиса, который будет слушать модуль. На этот адрес следует пробросить внешний порт или настроить проксирующий сервер для поддержки протокола https.
- BODY_LIMIT — ограничение размера файла. По умолчанию 10Mb
- ACCESS_KEY — секретный ключ доступа к API
- DB_PATH — путь к папке с файлами внутренней базы данных. По умолчанию «db».
- DB_SYNC_INTERVAL — периодичность (в секундах) сохранения актуальных данных на диск.
- FILE_PATH — путь к сохраняемым файлам объектов.
- LOG_LEVEL - уровень логгирования. Поддерживаемые значения:
 - error
 - warn
 - info
 - debug

Пример настройки модуля:

```
pudgeio:
build:
  context: .
restart: always
ports:
  - '80:80'
volumes:
  - ./db:/db
  - ./files:/files
environment:
  FILE_PATH: /files
  DB_PATH: /db
  ACCESS_KEY: valid-key
  LOG_LEVEL: DEBUG
extra_hosts:
  - "host.docker.internal:host-gateway"
```