



Общество с ограниченной ответственностью

«ОБИТ-Тех»

ИНН: 7842527295

ОГРН: 1147847316968

*Юридический адрес: 191014, г. Санкт-Петербург, пр-кт
Лиговский, д. 13-15, помещ. 19-Н*

**Руководство пользователя
программного обеспечения “Модуль сжатия данных”
по установке и эксплуатации**

г. Санкт-Петербург

2026 год

1. Установка программного обеспечения “Модуль сжатия данных”	3
2. Эксплуатация программного обеспечения “Модуль сжатия данных”	11
2.1. Общая информация	11
2.2. Описание конфигурации обработки изображений	12
2.3. Описание работы и пример	13
2.4. Справочник функций обработки изображений	14
2.5. Обработка PDF	17

Настоящий документ предназначен для описания администрирования и эксплуатации разработанного Модуля сжатия данных (далее -Система, Сервис), целью обеспечения которого является оптимизация объемов файлов оригиналов документов, получаемых из информационных систем пользователя.

1. Установка программного обеспечения “Модуль сжатия данных”

Технические требования к серверным ресурсам и программному обеспечению для размещения и работы Системы:

1. Операционная система: Astra Linux, RedOS, Alt Linux, Debian, Ubuntu
2. 8 ядер CPU/vCPU, производительностью не ниже чем Intel(R) Xeon(R) CPU E5-2699 v4
3. 16 гигабайт оперативной памяти
4. 500 гигабайт свободного места на SSD диске
5. Docker или совместимая с OCI платформа контейнеризации
6. Широкополосный интернет

Сервис поставляется в виде docker контейнера и распространяется с серверов компании Обит-Тех.

Для доступа к сервису используется пара логин и пароль для защиты от несанкционированного доступа. Связку логин-пароль выдаётся заказчику после приёма всех работ.

Для предварительной настройки при использовании ОС Ubuntu/Debian:

Установите ОС Ubuntu 20.04 LTS или Ubuntu 22.04 LTS. Воспользуйтесь официальным источником для установки дистрибутива:

<https://ubuntu.com/download/server>

Внимание! Во время установки ОС выделите под рутовый «/» раздел не менее 32 Гб, под SWAP раздел выделить 8 Гб, оставшееся место выделите под раздел «/var/lib/docker».

Удалите пакеты snapd и ufw с помощью команды:

```
apt autoremove --purge snapd ufw
```

Установите программное обеспечение Docker. Для установки воспользуйтесь официальным источником:

<https://docs.docker.com/install/linux/docker-ce/ubuntu/>

Внимание! Если ПО Docker распаковано из пакета *snapd*, удалите его и выполните установку из официального источника.

Пример последовательности команд для установки docker в Ubuntu:

```
apt-get remove docker docker-engine docker.io containerd runc
apt-get update
apt-get install ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
echo \ "deb [arch=$(dpkg --print-architecture)
```

```
signedby=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu \ $(lsb_release -cs)  
stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null  
apt-get update  
apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Установите дополнительное ПО (см. ниже).

Для установки дополнительного ПО:

- Выполните установку NTP-сервера с помощью команды:
apt install chrony
- Если имеются источники точного времени внутри компании, в файл
/etc/chrony/chrony.conf
- Укажите адреса Ваших серверов NTP в формате:
server ntp1.local
server ntp2.local
server ntp3.local
- Запустите сервис chrony для синхронизации времени:
systemctl enable chrony
systemctl restart chrony
- Для проверки подключения к NTP-серверам используйте следующую команду:
chronyc sources -v
- Укажите параметры хранения журналов в Docker в каталоге
/etc/docker/daemon.json:

```
{  
  "log-driver": "json-file",  
  "log-opts": {  
    "max-size": "1g"  
  }  
}
```
- Выполните:
systemctl restart docker
- Запустите командную строку.
- Подключитесь к репозиторию разработчика в Docker для скачивания контейнеров.
docker login -u <login> -p <password> docker-hub.obit.ru

Примечание: Учётные данные для аутентификации в репозитории предоставляются разработчиком системы.

- Создайте рабочий каталог проекта:
mkdir -p /opt/compressor && cd /opt/compressor
- В каталоге создаём файлы:

▪ **docker-compose.yml:**

version: '2'

services:

compressor:

image: docker-hub.obit.ru/obit/rosseti/dev/compress:v1.4.0-dev29

restart: always

volumes:

- ./backup:/root/backup

- ./config.yml:/root/config.yml:ro

- ./default-pipeline-image.json:/root/default-pipeline-image.json:ro

- ./default-pipeline-pdf.json:/root/default-pipeline-pdf.json:ro

ports:

- 8081:8081

- 8000:8000

- 8001:8001

environment:

- GOGC=50

depends_on:

- ocr-proxy

tc1:

hostname: app

restart: always

expose:

- "8888"

image: docker-hub.obit.ru/obit/rosseti/dev/compress-ocr:v1.0.0-dev4

environment:

PSM: 3

deploy:

mode: replicated

replicas: 20

tc2:

hostname: app2

restart: always

expose:

- "8888"
- image: docker-hub.obit.ru/obit/rosseti/dev/compress-ocr:v1.0.0-dev4
- environment:
 - PSM: 3
- deploy:
 - mode: replicated
 - replicas: 20
- nginx:
 - image: docker-hub.obit.ru/public/nginx:1.22.1-alpine
 - restart: always
 - hostname: nginx
 - expose:
 - "80"
 - ports:
 - "8088:80"
 - depends_on:
 - ocr-proxy
 - volumes:
 - ./nginx.conf:/etc/nginx/nginx.conf:ro
- ocr-proxy:
 - image: docker-hub.obit.ru/public/haproxy-alpine:2.4
 - restart: always
 - hostname: ocr-proxy
 - volumes:
 - ./haproxy.cfg:/usr/local/etc/haproxy/haproxy.cfg:ro
 - ports:
 - 8404:8404
 - 8480:80
- networks:
 - default:
 - ipam:
 - driver: default
 - config:
 - subnet: 100.64.0.0/24

- **nginx.conf:**

```
user nginx;
events {
    worker_connections 1000;
}
http {
    proxy_read_timeout 300;
    upstream tcupstream {
        server app:8888;
        server app2:8888;
    }
    server {
        listen 80;
        location / {
            client_max_body_size 200M;
            proxy_pass http://tcupstream;
        }
    }
}
```

- **haproxy.cfg**

```
global
    stats socket /var/run/api.sock user haproxy group haproxy mode 660 level
admin expose-fd listeners
    log stdout format raw local0 debug
defaults
    mode http
    timeout client 300s
    timeout connect 10s
    timeout server 300s
    timeout http-request 300s
    log global
    option httplog
```

```
log-format "%ci:%cp [%t] %ft %b/%s %Tw/%Tc/%Tt %B %ts  
%ac/%fc/%bc/%sc/%rc %sq/%bq"
```

frontend stats

```
bind *:8404
```

```
stats enable
```

```
stats uri /
```

```
stats refresh 10s
```

frontend myfrontend

```
bind :80
```

```
default_backend webservers
```

```
http-response set-log-level debug
```

```
http-request set-log-level debug
```

```
option httplog
```

backend webservers

```
server s1 nginx:80 check
```

```
http-response set-log-level debug
```

```
http-request set-log-level debug
```

- **config.yml:**

project:

```
name: Obit Compressor Project
```

```
debug: true
```

```
environment: production
```

http:

```
host: 0.0.0.0
```

```
port: ${HTTP_PORT|8081} # example environment variables
```

```
path: /
```

ocr_service:

```
host: http://ocr-proxy
```

```
port: 80
```

```
path_text: /image2text/
```

```
path_conf: /images2conf/
```

count_char: 15
confidence_percent: 50
max_width_recognition: 2500

status:
host: 0.0.0.0
port: 8000
livenessPath: /live
readinessPath: /ready
versionPath: /version

metrics:
host: 0.0.0.0
port: 8001
path: /metrics

jaeger:
enabled: false
service: "obit-compressor-project"
host: "jaeger"
port: ":6831"

compressor:
default_pipeline_file_image: "default-pipeline-image.json"
default_pipeline_file_pdf: "default-pipeline-pdf.json"
return_original_if_biggest: true
local_backup: false

pdf:
mfu_names: "xerox,canon,laserjet,deskjet,laser
mfp,epson,kyocera,ecosys,pantum,samsung,brother"
image:
background_white:
R: 230
G: 230

B: 230

filled_percent: 50

- **default-pipeline-image.json**

```
{
  "pipelines": [
    {
      "name": "q 55", "pipes": [
        { "fn": "quality", "params": { "quality": 55 } },
        { "fn": "pngQuality", "params": { "level": 2, "quantize": true } }
      ],
    },
    {
      "name": "q 45", "pipes": [
        { "fn": "quality", "params": { "quality": 45 } },
        { "fn": "pngQuality", "params": { "level": 2, "quantize": true } }
      ],
    },
    {
      "name": "q 35", "pipes": [
        { "fn": "quality", "params": { "quality": 35 } },
        { "fn": "pngQuality", "params": { "level": 3, "quantize": true } }
      ]
    }
  ]
}
```

- **default-pipeline-pdf.json**

```
{
  "pipelines": [
    {
      "name": "pdf images", "pipes": [
        { "fn": "relativeQuality", "params": { "qualityPercent": 80, "threshold": 50 } },
        { "fn": "pngQuality", "params": { "level": 3, "quantize": true } }
      ]
    },
    {
      "compress_image_into_doc": true,
      "auto_find_best_result": true
    }
  ]
}
```

- После создания всех запускаем сервис:
docker compose up -d
- Проверяем, что сервис корректно запустился
docker compose ps -a

Name	Command	State	Ports
compress-with-ocr_compressor_1	./api-server	Up	0.0.0.0:8000->8000/tcp, 0.0.0.0:8001->8001/tcp, 0.0.0.0:8081->8081/tcp
compress-with-ocr_nginx_1	/docker-entrypoint.sh nginx ...	Up	0.0.0.0:8088->80/tcp
compress-with-ocr_ocr-proxy_1	/docker-entrypoint.sh hapr ...	Up	0.0.0.0:8480->80/tcp, 0.0.0.0:8404->8404/tcp
compress-with-ocr_tc1_1	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_10	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_11	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_12	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_13	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_14	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_15	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_16	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_17	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_18	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_19	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_2	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_20	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_3	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_4	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_5	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_6	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_7	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_8	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc1_9	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_1	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_10	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_11	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_12	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_13	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_14	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_15	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_16	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_17	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_18	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_19	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_2	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_20	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_3	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_4	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_5	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_6	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_7	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_8	python /app/main.py	Up	8888/tcp
compress-with-ocr_tc2_9	python /app/main.py	Up	8888/tcp

Сервис готов к работе и принимает запросы на <http://<ip адрес сервера>:8081/v1/upload>

2. Эксплуатация программного обеспечения “Модуль сжатия данных”

2.1. Общая информация

Приложение «Модуль сжатия данных» не предполагает пользовательского интерфейса. Всё взаимодействие с ним происходит посредством HTTP вызовов к ручкам, которые предоставляют API интерфейс к функциональности.

Ручка /v1/upload ожидает на вход HTTP POST запрос в формате multipart/form-data с полями:

Наименование поля	Тип	Обязательность	Описание
source	File (бинарный)	Да	Файл, который необходимо обработать
config	Строка (json)	Нет	Конфигурация обработки файла - описание последовательности обработки

Если поле `config` не передано, то будет использоваться конфигурация по умолчанию.

Конфигурация по умолчанию задаётся администратором и передаётся приложению при его старте.

2.2. Описание конфигурации обработки изображений

Конфигурация обработки файла представляет собой JSON файл, в котором описываются шаги, которые применяются к входному файлу.

Общая структура конфигурации:

- Pipelines - массив, содержащий конфигурации конвейеров обработки
 - Каждый элемент конфигурации представляет собой объект, состоящий из двух полей:
 - name* - наименование конфигурации
 - pipes* - массив, описывающий последовательность применения функций обработки к изображению
 - Элемент массива *pipes* представляет собой объект, состоящий из двух полей:
 - fn* - наименование функции
 - params* - параметры функции, для каждой функции параметры специфичны и должны соблюдаться исходя из справочной информации

Схема JSON конфигурации:

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "properties": {
    "pipelines": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": { "type": "string" },
          "pipes": {
            "type": "array",
            "items": {
              "type": "object",
              "properties": {
                "fn": {
                  "type": "string"
                },
                "params": {
                  "type": "object"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

    },
    "required": ["fn", "params"],
    "additionalProperties": false
  }
},
"required": ["name", "pipes"],
"additionalProperties": false
}
},
"required": ["pipelines"],
"additionalProperties": false
}

```

2.3. Описание работы и пример

Сервис считывает конфигурацию и определяет количество конвейеров обработки. Для каждого конвейера формируется отдельная копия исходного файла, которая прогоняется по персональному конвейеру, в результате мы получаем несколько вариантов одного и того же файла.

Пример конфигурации:

```

{
  "pipelines": [
    {
      "name": "variant 1",
      "pipes": [
        { "fn": "greyscale", "params": null },
        { "fn": "maxWidth", "params": { "maxWidth": 1024, "ignoreLarger": 2048 } },
        { "fn": "quality", "params": { "quality": 30 } }
      ]
    },
    {
      "name": "variant 2",
      "pipes": [
        { "fn": "smartdpi", "params": { "x": 72, "y": 72, "returnSize": true } },
        { "fn": "quality", "params": { "quality": 30 } }
      ]
    },
    {
      "name": "variant 3",
      "pipes": [
        { "fn": "quality", "params": { "quality": 50 } }
      ]
    }
  ]
}

```

```

    },
    {
      "name": "variant 4",
      "pipes": [
        { "fn": "quality", "params": { "quality": 10 } }
      ]
    }
  ]
}

```

В данном примере будет создано 4 конвейера обработки и будет получено 4 варианта обработанного исходного файла, из которых и будет выбираться наилучший вариант.

Рассмотрим одну из конфигураций конвейеров подробнее:

```

"name": "variant 1",
  "pipes": [
    { "fn": "greyscale", "params": null },
    { "fn": "maxWidth", "params": { "maxWidth": 1024, "ignoreLarger": 2048 } },
    { "fn": "quality", "params": { "quality": 30 } }
  ]

```

В данном варианте изображение будет обработано следующими шагами:

1. Изображение приводится к градациям серого
2. Если изображение меньше 2048 пикселей по ширине, то оно будет отмасштабировано по ширине до 1024 пикселей (высота вычисляется автоматически)
3. Изображению будет установлено JPEG сжатие 30

2.4. Справочник функций обработки изображений

В данном справочнике предоставляется список функций и описание их параметров.

quality	Числовое значение от 1 до 100
---------	-------------------------------

- **quality**
Для изображения не в формате PNG установление значения сжатия данных.

Параметры:

Пример:

```
{ "fn": "quality", "params": { "quality": 30 } }
```

Установит значение сжатия в 30

- **relativeQuality**
Для изображения не в формате PNG, если его текущее значение параметра сжатия выше указанного, установить уровень сжатия в n процентов от текущего.

Параметры:

qualityPercent	Уровень сжатия в % от текущего значения
threshold	Порог срабатывания функции

Пример:

```
{ "fn": "relativeQuality", "params": { "qualityPercent": 70, "threshold": 50 } }
```

Возьмём текущее значение сжатия за N, если $N > 50$, то изображению будет установлено сжатие $(N/100)*70$. Для примера, если текущее значение N равно 40, то изображение не будет изменено, а если значение N равно 80, то ему будет установлено значение сжатия 56 $((80 / 100) * 70 = 56)$.

- **pngQuality**

Функция устанавливает уровень сжатия и необходимость квантизации для изображений в формате PNG.

Параметры:

level	Уровень обработки PNG изображения.
quantize	Квантизировать изображение (оптимизация палитры) или упростить изображение через уменьшение количества используемых оттенков.

Уровень	Квантируем	Упрощаем
1	палитра - 512 цветов	Глубина - 8bit, групп оттенков - 60
2	палитра - 256 цветов	Глубина - 5bit, групп оттенков - 50
3	палитра - 128 цветов	Глубина - 4bit, групп оттенков - 35
4	палитра - 3 цвета	Глубина - 2bit, групп оттенков - 10

Пример:

```
{ "fn": "pngQuality", "params": { "level": 3, "quantize": true } }
```

- **dpi**

Изменение DPI (количество точек на дюйм) изображения

Параметры:

x	Целевое количество точек по оси X
---	-----------------------------------

y	Целевое количество точек по оси Y
returnSize	При изменении dpi у изображения меняется его размер в пикселях по ширине/высоте. Данный параметр позволяет вернуть изображению его изначальный размер.

Пример:

```
{ "fn": "dpi", "params": { "x": 96, "y": 96, "returnSize": true } }
```

Установит размер в 96 dpi и изображение будет возвращено в изначальном размере в пикселях по ширине/высоте.

- **smartdpi**

Функция аналогична **dpi**, только дополнительно проверяет, не будет ли изменение dpi слишком резким. Если изменение dpi слишком резкое (больше чем на 50%) функция не активируется.

Пример:

```
{ "fn": "smartdpi", "params": { "x": 96, "y": 96, "returnSize": true } }
```

Установит размер в 96 dpi и изображение будет возвращено в изначальном размере в пикселях по ширине/высоте, только в случае, если изначальный dpi меньше 192.

- **greyscale**

Преобразует изображение из цветного к градациям серого.

Параметры: нет

Пример:

```
{ "fn": "greyscale", "params": { } }
```

Установит размер в 96 dpi и изображение будет возвращено в изначальном размере в пикселях по ширине/высоте.

- **imageFormat**

Изменение формата изображения

Параметры:

format	В какой формат сконвертировать исходное изображение, например JPEG или PNG
--------	--

Пример:

```
{ "fn": "imageFormat", "params": { "format": "PNG" } }
```

Преобразует исходное изображение в формат PNG.

- **maxWidth**

Изменить размер изображения, если его ширина меньше указанного значения

Параметры:

maxWidth	Целевое значение ширины изображения в px, высота высчитывается автоматически
ignoreLarger	Значение в px ширины изображения, которое не изменяем

Пример:

```
{ "fn": "maxWidth", "params": { "maxWidth": 1024, "ignoreLarger": 2048 } }
```

Если изображение по ширине меньше 2048 пикселей, то установить его ширину в 1024 пикселя.

- **brightnessContrast**

Установка яркости и контраста изображения

Параметры:

brightness	Значение яркости (от -100 до 100)
contrast	Значение контраста (от -100 до 100)

Пример:

```
{ "fn": "brightnessContrast", "params": { "brightnes": -50, "contrast": 20 } }
```

Уменьшит яркость изображения на -50, увеличить контраст до 20

- **normalize**

Повышает контрастность цветного изображения путем регулировки цвета пикселей для охвата всего диапазона доступных цветов.

Параметры: нет

- **enhance**

Применяет цифровой фильтр, улучшающий качество зашумленного изображения.

Параметры: нет

- **despeckle**

Уменьшает зернистость на изображении, сохраняя при этом края исходного изображения.

2.5. Обработка PDF

Конфигурация обработки PDF файлов аналогична обработки изображений, но есть дополнительный параметр:

compress_image_into_doc	На многостраничных (30+) текстовых документах может работать немного дольше. Все изображения в документе извлекаются как есть, конвертируется в разные форматы для определения наилучшего сжатия, далее проходят обработку в соответствии с переданным конфигом и обработанное изображение заменяет оригинальное внутри документа. После сохраняются свойства документа (title, creator, etc.), документ проходит оптимизацию и отдаётся результат.
auto_find_best_result	При значении true: В разных документах, в зависимости от compress_image_into_doc результат сжатия может быть разным. При включённом <i>auto_find_best_result=true</i> документ проходит все (описанные ниже) варианты сжатия и возвращает наименьший по размеру результат. После сжатия сохраняются свойства документа (title, creator, etc.), документ проходит оптимизацию и отдаётся результат. Т.к. сжатие документа запускаются параллельно, потребляет немного больше памяти. При значении false: Сначала в документе ищутся признаки сканированного документа: это текст, ссылки, формы, признак МФУ. Если в документе присутствует текст, ссылки, формы и т.п., то документ идёт в обработку в GS, где происходит сжатие документа и изображений в нём. Если текст не был найден, то считаем, что это скан, страницы извлекаются как единое изображение, затем: <u>! этот пункт пропускается из-за утечек памяти при получении гистограммы в imagemagick.</u> Считаем процент заполнения страницы изображения. При подсчёте белый (не считаемый) фон берём из <i>config.yml:pdf.image.background_white</i>. Если заполняемость выше <i>config.yml:pdf.image.filled_percent</i>, считаем, что они "грязные" (например, таблицы с покраской цвета) и такие изображения не подлежат обработке. Далее, если изображение подходит, идёт обработка в соответствии с переданным конфигом. После страница изображения конвертируется в несколько форматов для определения наилучшего сжатия. После сохраняются свойства документа (title, creator, etc.), документ проходит оптимизацию и отдаётся результат.

Пример конфигурации:

```
{
  "pipelines": [
    {"name": "pdf images", "pipes": [
      {"fn": "relativeQuality", "params": { "qualityPercent": 70, "threshold": 50 }},
```

```
    {"fn": "pngQuality", "params": { "level": 3, "quantize": true}}  
  ]}  
],  
"compress_image_into_doc": false,  
"auto_find_best_result": true  
}
```