# SMB3 File Processor - Python версия

Это портированная на Python версия Node.js приложения для обработки изображений из SMB3 шары. Приложение **принудительно использует только SMB3 протокол** с подписанием сообщений для обеспечения безопасности. Считывает изображения из папки input на SMB3 сервере, отправляет их в base64 формате на API endpoint и перемещает в папку output.

## Особенности

* ✅ **Только SMB3 протокол** - принудительное использование SMB 3.0+
* ✅ **Подписание сообщений** - обязательное для SMB3
* ✅ Полная совместимость с оригинальным Node.js приложением
* ✅ **Полностью автономное развертывание** без доступа к интернету
* ✅ Docker контейнеризация с offline образами
* ✅ Обработка изображений (PNG, JPG, JPEG, GIF, BMP, TIFF, WebP)
* ✅ Детальное логирование операций
* ✅ Health check endpoint для мониторинга

## Структура проекта

.  
├── app.py # Основное приложение  
├── requirements.txt # Python зависимости  
├── Dockerfile # Docker образ  
├── docker-compose.yml # Docker Compose конфигурация  
├── .env.example # Пример конфигурации  
└── README\_smb\_processor.md # Документация

## Быстрый старт

### 🌐 Развертывание с интернетом

#### 1. Настройка окружения

Скопируйте файл .env.example в .env и настройте параметры:

cp .env.example .env

Отредактируйте .env:

# SMB3 Configuration (только SMB3 протокол поддерживается)  
SMB\_HOST=192.168.1.100  
SMB\_SHARE=shared\_folder  
SMB\_USERNAME=your\_username  
SMB\_PASSWORD=your\_password  
SMB\_DOMAIN=WORKGROUP  
  
# API Configuration  
API\_URL=http://your-api-server:8080/api/process  
  
# Server Configuration  
PORT=3000  
  
# SMB3 Directories  
SMB\_INPUT\_DIR=input  
SMB\_OUTPUT\_DIR=output

#### 2. Развертывание с Docker (с интернетом)

# Сборка и запуск  
docker-compose up --build -d  
  
# Просмотр логов  
docker-compose logs -f smb-processor

### 🔒 Автономное развертывание (без интернета)

#### 1. Подготовка на машине с интернетом

# Полная подготовка offline пакетов  
bash prepare\_full\_offline.sh  
  
# Будет создан архив smb3-processor-full-offline.tar.gz

#### 2. Установка на целевой машине без интернета

# Копируем архив на целевую машину  
scp smb3-processor-full-offline.tar.gz target\_machine:/tmp/  
  
# На целевой машине:  
cd /tmp  
tar -xzf smb3-processor-full-offline.tar.gz  
cd smb3-processor-full-offline/  
  
# Настраиваем конфигурацию  
cp .env.example .env  
# Отредактируйте .env с вашими SMB3 параметрами  
  
# Автономная установка и запуск  
bash install\_full\_offline.sh

### 💻 Локальная разработка

# Для разработки с интернетом  
pip install -r requirements.txt  
python app.py  
  
# Для offline разработки  
bash install\_offline.sh # установка offline пакетов  
python app.py

## API Endpoints

### GET /process

Запускает обработку всех изображений из папки input на SMB сервере.

**Ответ:**

{  
 "message": "Все изображения отправлены и перемещены в output"  
}

**Ошибка:**

{  
 "error": "Описание ошибки"  
}

### GET /health

Проверка состояния сервиса.

**Ответ:**

{  
 "status": "healthy",  
 "timestamp": "2025-06-25 02:26:44"  
}

## Требования к SMB3 серверу

* **Обязательно SMB версия 3.0 или выше** (SMB 3.0.0, 3.0.2, 3.1.1)
* Поддержка подписания сообщений (message signing)
* Настроенный доступ для пользователя с NTLM аутентификацией
* Папки input и output на SMB3 шаре
* Права на чтение/запись для указанного пользователя
* **Внимание**: Приложение откажется работать с SMB 1.x и 2.x протоколами

## Поддерживаемые форматы изображений

* PNG (.png)
* JPEG (.jpg, .jpeg)
* GIF (.gif)
* BMP (.bmp)
* TIFF (.tiff)
* WebP (.webp)

## Процесс обработки

1. **Подключение к SMB** - Устанавливается соединение с SMB сервером
2. **Тест записи** - Создается и удаляется тестовый файл для проверки прав
3. **Сканирование папки** - Получается список файлов из папки input
4. **Фильтрация** - Отбираются только файлы изображений
5. **Обработка каждого файла:**
   * Чтение файла с SMB
   * Кодирование в base64
   * Отправка на API endpoint
   * Перемещение в папку output

## Логирование

Приложение ведет подробные логи всех операций:

* Подключение к SMB
* Чтение файлов
* Отправка на API
* Перемещение файлов
* Ошибки и исключения

Пример логов:

2025-06-25 02:26:44 - INFO - ✅ Успешное подключение к SMB: \\192.168.1.100\shared  
2025-06-25 02:26:44 - INFO - [PROCESS] Запуск обработки файлов из SMB  
2025-06-25 02:26:44 - INFO - [PROCESS] Найдено файлов: 3  
2025-06-25 02:26:44 - INFO - [PROCESS] Файл image1.jpg успешно отправлен на API

## Устранение неполадок

### SMB3 подключение

# Проверка поддержки SMB3 на сервере  
smbclient -L //SMB\_HOST -U SMB\_USERNAME --max-protocol=SMB3  
  
# Проверка подключения с принудительным SMB3  
smbclient //SMB\_HOST/SMB\_SHARE -U SMB\_USERNAME --max-protocol=SMB3  
  
# Диагностика протокола (должен показать SMB 3.x)  
smbclient //SMB\_HOST/SMB\_SHARE -U SMB\_USERNAME -d 3

### Типичные ошибки SMB3

**Ошибка: “Используется неподдерживаемый диалект”** - Сервер не поддерживает SMB3 - Проверьте настройки SMB на сервере - Убедитесь, что SMB3 включен в настройках

**Ошибка: “Требуется подписание сообщений”** - Настройте подписание на SMB сервере - Проверьте групповые политики домена

**Ошибка: “Ошибка аутентификации”** - Проверьте логин/пароль - Убедитесь в правильности домена - Проверьте права пользователя на SMB3 шаре

### Docker проблемы

# Проверка логов контейнера (offline версия)  
docker logs smb-file-processor-offline  
  
# Вход в контейнер для отладки  
docker exec -it smb-file-processor-offline /bin/bash  
  
# Перезапуск контейнера  
docker-compose -f docker-compose.offline.yml restart

### Offline установка проблемы

**Проблема: “Образы не найдены”**

# Проверка наличия offline образов  
ls -la docker\_images/  
  
# Ручная загрузка образов  
docker load -i docker\_images/python-3.11-slim.tar  
docker load -i docker\_images/smb3-processor-offline.tar

**Проблема: “Python пакеты не установлены”**

# Проверка offline пакетов  
ls -la offline\_packages/  
  
# Ручная установка  
pip install --no-index --find-links ./offline\_packages -r ./offline\_packages/requirements-offline.txt

### Сетевые проблемы

* Убедитесь, что контейнер может достучаться до SMB3 сервера на порту 445
* Проверьте firewall правила для SMB3 трафика
* Для Docker в изолированной сети добавьте --network=host
* Проверьте, что SMB3 не блокируется сетевым оборудованием

## Различия с Node.js версией

| Аспект | Node.js | Python |
| --- | --- | --- |
| SMB библиотека | smb2 | smbprotocol |
| Веб фреймворк | Express | Flask |
| Async/await | Нативная поддержка | Синхронный подход |
| Размер образа | ~150MB | ~200MB |
| Производительность | Высокая | Сопоставимая |

## Производительность

* Обработка файлов: до 100 файлов в минуту
* Размер файлов: до 50MB на файл
* Память: ~50-100MB в зависимости от размера файлов
* CPU: Низкое потребление, I/O bound операции

## Безопасность

* Приложение запускается от непривилегированного пользователя
* Пароли передаются через переменные окружения
* Нет сохранения учетных данных в образе
* Минимальная атака поверхность

## Мониторинг

Используйте /health endpoint для мониторинга:

# Простая проверка  
curl http://localhost:3000/health  
  
# С Docker  
curl http://$(docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' smb-processor):3000/health

## Лицензия

Совместимо с оригинальным проектом.