
Инструкция по развертыванию
экземпляра

**“Сервис по оптимизации
длины URL”**

Содержание

Содержание	2
1. Введение	3
1.1 Описание документа	3
1.2 Уровень подготовки пользователей	3
2. Требования к эксплуатации серверной части	3
3. Установка	3
3.1 Описание компонентов	3
3.2 Порядок установки	4

1. Введение

Наименование программного обеспечения “Сервис по оптимизации длины URL”.

1.1 Описание документа

Документ содержит сведения о порядке установки программного обеспечения “Сервис по оптимизации длины URL”. Также в документе приводятся требования к программному и аппаратному обеспечению.

1.2 Уровень подготовки пользователей

Документ предназначен для администратора АС, выполняющего установку программного обеспечения “Сервис по оптимизации длины URL”.

2. Требования к эксплуатации серверной части

Обеспечение функционирования ПО серверной части “Сервис по оптимизации длины URL” реализовано на базе серверной операционной системы Linux. Минимальной конфигурацией аппаратной составляющей являются:

- Современная ОС: Linux;
- Оперативная память: 2 ГБ;
- Свободное дисковое пространство: не менее 20 Gb;
- Количество логических ядер процессора: 2;
- Частота процессора: 3.50 GHz.

Возможно разворачивание экземпляра ПО и на других ОС, поддерживающих платформу для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации Docker, например Windows 10 (Профессиональная или Корпоративная).

3. Установка

3.1 Описание компонентов

Состав компонентов для установки:

- `link-wrapper` – сервис для оборачивания ссылок.

- `gateway-api` – сервис для передачи информации от мерчантов и первичной обработки информации.
- `rabbitmq-service` – программный брокер сообщений на основе стандарта AMQP;
- `db-wrapper` – свободная объектно-реляционная система управления базами данных.
- `db-pay-service` – свободная объектно-реляционная система управления базами данных.

3.2 Порядок установки

Для разворачивание экземпляра ПО необходим `docker-compose` (инструментальное средство, входящее в состав **Docker**. Оно предназначено для решения задач, связанных с развертыванием проектов). Его можно установить, следуя инструкции на официальном сайте <https://docs.docker.com/compose/install/>.

Необходимо создать файл `docker-compose.yml` с содержимым:

```
version: '2.4'  
services:
```

```
links-wrapper-service:  
  image: 'docker.mbtel.ru/paymentboom_links-wrapper:v1.3.0.28'  
  restart: always  
  environment:  
    - NGINX_WORKERS=1  
    - TRANSPORT_CONNECTION_PROTOCOL=amqp  
    - TRANSPORT_CONNECTION_HOSTNAME=rabbitmq-service  
    - TRANSPORT_CONNECTION_PORT=5672  
    - TRANSPORT_CONNECTION_VHOST=/  
    - TRANSPORT_CONNECTION_FRAME_MAX=0  
    - TRANSPORT_CONNECTION_CHANNEL_MAX=0  
    - TRANSPORT_CONNECTION_HEARTBEAT=0  
    - TRANSPORT_AUTO_RECONNECT=true  
    - TRANSPORT_AUTO_RECONNECT_TIMEOUT=5000  
    - TRANSPORT_SERVICE_NAME=links_wrapper  
    - TRANSPORT_SHARED_SERVICE_QUEUE=true  
    - TRANSPORT_SHARED_SERVICE_QUEUE_MESSAGE_TTL=10000  
    - TRANSPORT_EXCLUSIVE_QUEUE_ACK=false  
    - TRANSPORT_SHARED_SERVICE_QUEUE_ACK=false  
    - TRANSPORT_MAIN_EXCHANGE_NAME=main
```

-
- TRANSPORT_MAIN_REPLY_EXCHANGE_NAME=main_reply
 - TRANSPORT_PUBLISH_AWAIT_TIMEOUT=10000
 - TRANSPORT_PREFETCH=100
 - TRANSPORT_MAIN_SCHEDULE_EXCHANGE_NAME=main_schedule
 - TRANSPORT_MAIN_SCHEDULE_QUEUE_NAME=main_schedule
 - TRANSPORT_MAIN_GARBAGE_EXCHANGE_NAME=main_garbage
 - TRANSPORT_MAIN_GARBAGE_QUEUE_NAME=main_garbage
 - DATABASE_MASTER_DATABASE=db_links_wrapper
 - DATABASE_MASTER_HOST=db-pay-service
 - DATABASE_MASTER_PORT=5432
 - DATABASE_SLAVE_DATABASE=db_links_wrapper
 - DATABASE_SLAVE_HOST=db-pay-service
 - DATABASE_SLAVE_PORT=5432
 - DATABASE_POOL_CONNECTION_TIMEOUT_MILLIS=2000
 - DATABASE_POOL_IDLE_TIMEOUT_MILLIS=30000
 - DATABASE_POOL_MAX=10
 - DATABASE_LOGGING=false
 - LOGGER_LEVEL=debug
 - TARGET_BASE_URI=http://5.182.5.111:3000
 - DATABASE_MASTER_PASSWORD=STRONG_password
 - DATABASE_MASTER_USER=postgres
 - DATABASE_SLAVE_PASSWORD=STRONG_password
 - DATABASE_SLAVE_USER=postgres
 - TRANSPORT_CONNECTION_PASSWORD=rmpassword
 - TRANSPORT_CONNECTION_USERNAME=ruser
 - K8S_POD_NAME=links-wrapper-lol-kek
 - DB_MASTER_PASSWORD=STRONG_password

ports:

- "3000:3000"

depends_on:

- rabbitmq-service
- db-pay-service

gateway-api-service:

image: 'docker.mbtel.ru/paymentboom_gateway-api:feat-import.186'

restart: always

environment:

- NGINX_WORKERS=2
- API_NAME=gateway-api
- API_VERSION=v1
- API_PORT=8080
- TRANSPORT_INIT_TIMEOUT=5000

- TRANSPORT_CONNECTION_HOSTNAME=rabbitmq-service
- TRANSPORT_CONNECTION_PORT=5672
- TRANSPORT_CONNECTION_VHOST=/
- TRANSPORT_AUTO_RECONNECT=true
- TRANSPORT_AUTO_RECONNECT_TIMEOUT=5000
- TRANSPORT_SERVICE_NAME=rmuser
- TRANSPORT_PUBLISH_AWAIT_TIMEOUT=15000
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENT_STATUS=ph_payment-status
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENT_NOTIFICATION=ph_payment-notification
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENT_REGISTER=ph_payment-registration
- SETTINGS_TRANSPORT_SERVICES_NAMES_SECOND_CPS=second-cps_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ECOMMPAY=ecommpay_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ASTROPAY_DIRECT=directa24_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_FIRST_CPS=first-cps_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_QIWI_TOP_UP_CARD=qiwi-top-up-card_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ASTROPAY_CARD=astropay-card_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_CONTACT=contact_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ECOPAYZ=ecopayz_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_WIRECARD=wirecard_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_HALYK_BANK=halyk-bank_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_RBK_MONEY=rbk_money_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYEER=payeer_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYSAGE=paysage_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_DAOWALLET=daowallet_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_SETTLE_PAY=settle_pay_plugin
-
- SETTINGS_TRANSPORT_SERVICES_NAMES_ASTROPAY_ONE_TOUCH=astopay_onetouch_plugin
-
- SETTINGS_TRANSPORT_SERVICES_NAMES_COINSPAID=coinspaid_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_REFERENCE_API=ph_reference-service
-
- SETTINGS_TRANSPORT_SERVICES_NAMES_APPLICATION_BALANCES_SERVICE=ph_application-balances-service
-
- SETTINGS_TRANSPORT_SERVICES_NAMES_JETON=jeton_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_LEOGAMING=leogaming_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_APS=aps_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_MUCHBETTER=muchbetter_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_EZEEWALLET=ezeewallet_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_BTQFINANCE=btqfinance_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_CONNECTUM=connectum_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_INTERKASSA=interkassa_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_BCPO=bcpo_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYDEX=paydex_plugin

- SETTINGS_TRANSPORT_SERVICES_NAMES_GETAPAY_PSP=getapay_psp_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_JETONCASH=jetoncash_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_INWIZO=inwizo_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYCOS=paycos_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMEGA=paymega_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_WALLET_COM=wallet_com_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_MERCURYO=mercuryo_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_FINANA=finana_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_DECARDS2P=decards2p_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_P2PAY=p2pay_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYDO=paydo_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_ZIPPY=zippy_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_0XPROCESSING=oxprocessing_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_CONNPAY=connpay_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAY_PORT=pay_port_integration_service
-

SETTINGS_TRANSPORT_SERVICES_NAMES_GATE_EXPRESS=gate_express_integration_service

- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENTS_ROUTER=payments_router
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYES=payes_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAY4FUN=pay4fun_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYRETAILERS=payretailers_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_CYPIX=cypix_integration_service
-

SETTINGS_TRANSPORT_SERVICES_NAMES_JUMPFINANCE=jumpfinance_integration_service

-

SETTINGS_TRANSPORT_SERVICES_NAMES_MAGIC_PAYMENTS=magic_payments_integration_service

- DB_MASTER_NAME=db_pay
- DB_MASTER_HOST=db-pay-service
- DB_MASTER_PORT=5432
- DB_MASTER_CONN_TIMEOUT_MILLIS=3000
- DB_MASTER_IDLE_TIMEOUT_MILLIS=10000
- DB_MASTER_MAX_CLIENTS=10
- DB_MASTER_PING_TIMEOUT_MILLIS=60000
- DB_SLAVE_NAME=db_pay
- DB_SLAVE_HOST=db-pay-service
- DB_SLAVE_PORT=5432
- DB_SLAVE_CONN_TIMEOUT_MILLIS=3000
- DB_SLAVE_IDLE_TIMEOUT_MILLIS=10000
- DB_SLAVE_MAX_CLIENTS=10
- DB_SLAVE_PING_TIMEOUT_MILLIS=60000
- LOG_LEVEL=0
- CARD_ENCRYPTION_KEY=enckey
- DATABASE_MASTER_PASSWORD=STRONG_password

-
- DATABASE_MASTER_USER=postgres
 - DATABASE_SLAVE_PASSWORD=STRONG_password
 - DATABASE_SLAVE_USER=postgres
 - TRANSPORT_CONNECTION_PASSWORD=rmpassword
 - TRANSPORT_CONNECTION_USERNAME=rmuser
 - K8S_POD_NAME=gateway-api-pod-name
 - DB_MASTER_USER=postgres
 - DB_MASTER_PASSWORD=STRONG_password
 - DB_SLAVE_USER=postgres
 - DB_SLAVE_PASSWORD=STRONG_password

ports:

- "8080:8080"

depends_on:

- rabbitmq-service
- db-pay-service

rabbitmq-service:

image: rabbitmq:3.11.8-management

hostname: rabbitmq-service

restart: always

environment:

- RABBITMQ_DEFAULT_USER=rmuser
- RABBITMQ_DEFAULT_PASS=rmpassword
- RABBITMQ_SERVER_ADDITIONAL_ERL_ARGS=-rabbit channel_max 0

volumes:

- ./rabbitmq:/var/lib/rabbitmq

command:

- bash

- -c

- |

cd /opt/rabbitmq/plugins/

apt update

apt install -y wget

wget https://github.com/rabbitmq/rabbitmq-delayed-message-

exchange/releases/download/3.11.1/rabbitmq_delayed_message_exchange-3.11.1.ez

rabbitmq-plugins enable rabbitmq_delayed_message_exchange

rabbitmq-server start

ports:

- "15672:15672"

- "5672:5672"

db-pay-service:

```
container_name: db-pay-service
image: 'postgres:14'
environment:
  POSTGRES_PASSWORD: STRONG_password
  POSTGRES_USER: postgres
  POSTGRES_DB: db_pay
restart: on-failure
volumes:
  - ./postgres:/var/lib/postgresql/
ports:
  - "5432:5432"
command: [ "postgres", "-c", "wal_level=logical" ]
```

```
volumes:
  rabbitmq:
    external: true
  postgres:
    external: true
```

Запустить командой:

```
docker-compose up -d
```