



инфоэнерго

ООО «АЛЬФА-ИНТЕГРАТОР-ИНФОЭНЕРГО»

**Информационная система
самообслуживания потребителей
"Личный кабинет потребителей энергоресурсов"
(ИС Личный кабинет)**

**ИНСТРУКЦИЯ АДМИНИСТРАТОРА
(ДЛЯ УСТАНОВКИ ПО)**

**Санкт-Петербург
2023**

Оглавление

Принятые термины и сокращения	3
1. Общие положения	3
2. Описание архитектуры ИС Личный кабинет	3
3. Установка и настройка системы	5
4. Настройка сервисов системы.	7

Принятые термины и сокращения

Термин	Определение
ИС Личный кабинет	Информационная система самообслуживания потребителей "Личный кабинет потребителей энергоресурсов"
ПО	Программное обеспечение
СУБД	Система управления базами данных
Rabbit MQ	Программный брокер сообщений на основе стандарта AMQP
AMQP	Протокол передачи сообщений

1. Общие положения

Настоящая Инструкция системного администратора разработана с целью:

- описания архитектуры системы и взаимосвязи её компонентов;
- описания процесса установки и настройки системы;
- определения порядка диагностирования проблем функционирования системы;
- упорядочения работы должностных лиц, связанной с диагностированием проблем функционирования системы.

В настоящем документе регламентируются действия при выполнении следующих мероприятий:

- установка и настройка системы;
- диагностирование проблем функционирования системы.

2. Описание архитектуры ИС Личный кабинет

ИС Личный кабинет представляет собой программно-аппаратный комплекс, состоящий из серверов приложений, сервера базы данных и установленного на них ПО.

Схема развертывания ИС Личный кабинет в инфраструктуре Заказчика представлена на Рис.1:

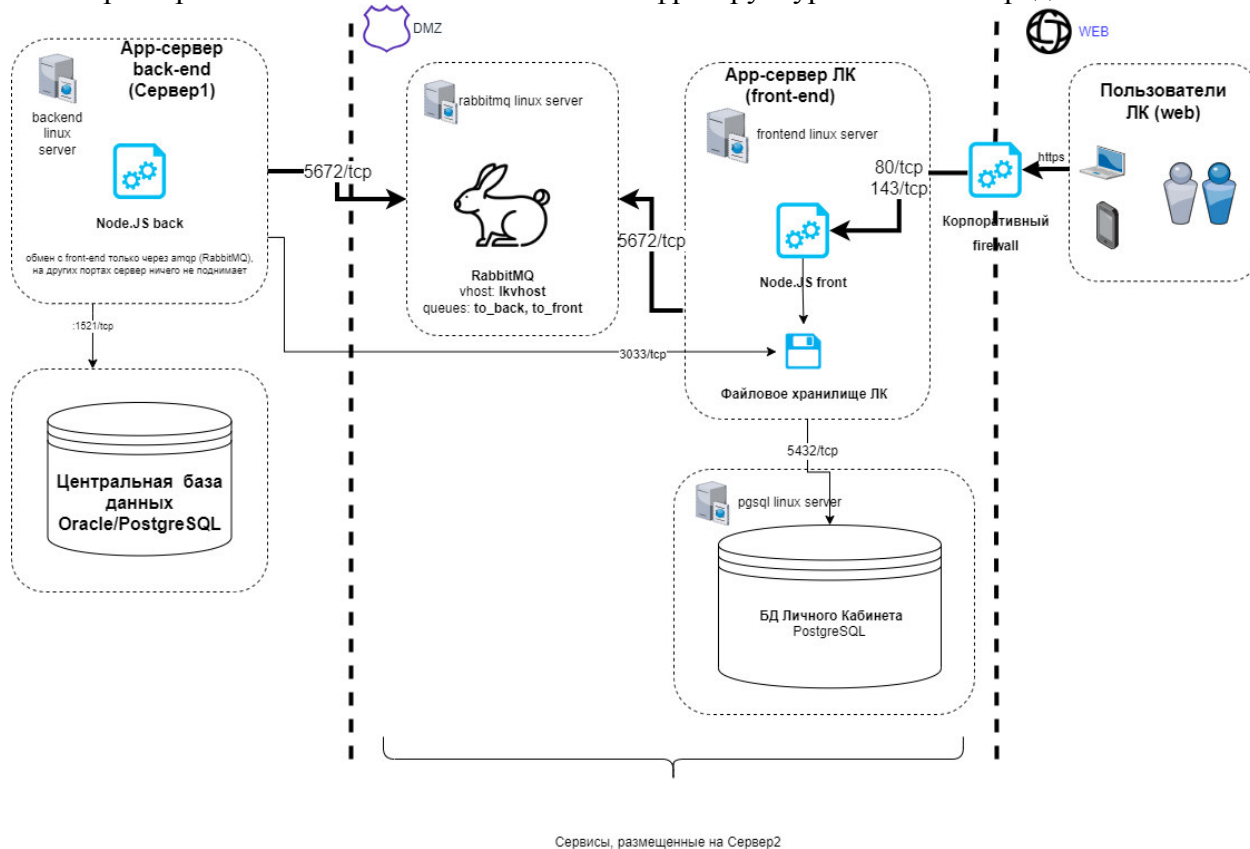


Рис 1. Схема развертывания ИС Личный кабинет

Система состоит из следующих программных модулей:

- Внутренний (back-end) модуль обеспечивает взаимосвязь с биллинговой системой
- Внешний модуль обеспечивает промежуточное хранение передаваемых из биллинговой системы и обратно данных, а также предоставление данных для вывода в веб-интерфейс клиента
- База данных (PostgreSQL), на которой развернуто промежуточное хранилище данных для отображения в личном кабинете
- Веб-интерфейс в браузере клиента – визуальная часть приложения. Веб-интерфейс загружается с внешнего сервера после соединения клиента (перехода на страницу ИС Личный кабинет) и выполняется в веб-браузере клиента.
- Шина данных (на основе RabbitMQ) отвечает за взаимодействие внешнего и внутреннего контура Системы.

Все модули, кроме внутреннего сервера размещаются на выделенном сервере, который, в целях безопасности, расположен в изолированной области локальной сети (DMZ) и не имеет доступа в локальную сеть. Обмен данными между модулями системы реализуется через шину данных по протоколу обмена сообщениями AMQP.

Сервера системы работают под управлением операционной системы РЕД ОС 7.3.3

Для работы системы используется следующее системное ПО:

СУБД PostgreSQL версии 14.9

RabbitMQ версии 3.11.20

Node.JS версии 18.14.2

3. Установка и настройка системы

3.1 Установка Rabbit MQ

Установка Rabbit MQ производится на одном из серверов системы, как правило front-end или сервере баз данных.

3.1.1 Проверить доступность модуля Rabbit MQ соответствующей версии в репозитории: **sudo dnf search rabbitmq-server -v**

3.1.2 Установить модуль Rabbit MQ из репозитория и запуск сервиса:

```
sudo dnf install rabbitmq-server  
sudo systemctl enable --now rabbitmq-server.service  
sudo rabbitmq-plugins enable rabbitmq_management
```

3.1.3 Проверить установленную версию Rabbit MQ командой: **sudo rabbitmqctl version**

3.1.4 Открыть порты для подключения к Rabbit MQ и удаленного управления (при необходимости) на firewall-е (в данном случае приложение на порту {5672,15672}/tcp):

- добавляем порт
sudo firewall-cmd --permanent --zone=public --add-port={5672,15672}/tcp
- перезапускаем firewall
sudo firewall-cmd --reload
- проверяем включение правила
sudo firewall-cmd --list-all

3.2 Установка и настройка PostgreSQL

Установка СУБД PostgreSQL производится на сервере баз данных или сервере front-end.

3.2.1 Проверить доступность модуля PostgreSQL соответствующей версии в репозитории:

```
sudo dnf search postgresql14 -v
```

3.2.2 Установить PostgreSQL из репозитория: **sudo dnf install postgresql14-server postgresql14-contrib**

3.2.3 Инициализировать базу данных: **sudo /usr/pgsql-14/bin/postgresql-14-setup initdb**

3.2.4 Запустить и включить в автозапуск службу: **sudo systemctl enable --now postgresql-14**

3.2.5 Проверить доступность и установленную версию СУБД: **sudo -u postgres psql -c "SELECT version();"**

3.2.6 Задать пароль суперпользователю СУБД postgres:

```
sudo -u postgres psql  
\password postgres  
\q
```

3.2.7 Настроить парольную аутентификацию в файле конфигурации pg_hba.conf (расположение по-умолчанию /var/lib/pgsql/14/data/pg_hba.conf в каталоге пользователя postgres):

```
# IPv4 local connections:
host all all 127.0.0.1/32 scram-sha-256
# IPv6 local connections:
host all all ::1/128 scram-sha-256
# IPv4 LAN connections:
host all all 192.168.1.19/32 scram-sha-256
```

В разделе LAN connections указать IP-адрес front-end сервера.

3.2.8 Настроить следующие параметры в файле конфигурации postgresql.conf (расположение по-умолчанию /var/lib/pgsql/14/data/postgresql.conf):

```
#Включаем прослушивание на внешних интерфейсах системы
listen_addresses = '*'
#Устанавливаем количество соединений
max_connections = 100
#Включаем уровень записи в WAL, необходимый для восстановления из резервной копии
wal_level = replica
#Включаем режим архивации
archive_mode = on
#Устанавливаем команду архивации, вместо /opt/postgres/wal_backup/ должен быть каталог, в котором будут размещаться резервные копии WAL
archive_command = 'test ! -f /opt/postgres/wal_backup/%f.gz && /usr/bin/gzip -c %p > /opt/postgres/wal_backup/%f.gz'
#Настраиваем политику удержания WAL
wal_keep_segments = 60
#Параметры локализации
lc_messages = 'en_US.UTF-8'
lc_monetary = 'ru_RU.UTF-8'
lc_numeric = 'ru_RU.UTF-8'
lc_time = 'ru_RU.UTF-8'
#Параметры отображения даты
datestyle = 'iso, mdy'
```

3.2.9 Перезапустить сервис PostgreSQL для применения новых параметров: **sudo systemctl restart postgresql-14**

3.2.10 Настройка резервного копирования

Создать скрипт резервного копирования. В данном примере каталог запуска PostgreSQL - /opt/postgresql, на сервере хранятся 2 последних базовых резервных копии (в каталоге db_backup) и WAL за последние 2-е суток (в каталоге wal_backup):

```
#!/bin/bash

PG_HOME=/opt/postgresql
export PG_HOME
mkdir $PG_HOME/pg_backup
/usr/bin/pg_basebackup -U postgres -D $PG_HOME/pg_backup -Ft -z -Xf

INDEX=$(date +"%u")

test -e $PG_HOME/db_backup/base.${INDEX}.tar.gz && rm $PG_HOME/db_backup/base.${INDEX}.tar.gz
cp $PG_HOME/pg_backup/base.tar.gz $PG_HOME/db_backup/base.${INDEX}.tar.gz

test -e $PG_HOME/db_backup/base.last.tar.gz && rm $PG_HOME/db_backup/base.last.tar.gz
ln $PG_HOME/db_backup/base.${INDEX}.tar.gz $PG_HOME/db_backup/base.last.tar.gz

rm -r $PG_HOME/pg_backup
test -e $PG_HOME/db_backup/base.${INDEX}.tar.gz && test -e $PG_HOME/db_backup/base.${date --date="-1 day" +"%u"}.tar.gz && find $PG_HOME/db_backup -type f -mtime +1 -exec rm {} \;
find $PG_HOME/wal_backup -type f -mtime +2 -exec rm {} \;
```

3.3 Сервисы под управлением Node.JS

3.3.1 Проверить доступность модуля среды выполнения Node.JS соответствующей версии в репозитории: **sudo dnf search nodejs -v |more**

Установка среды выполнения производится на всех серверах ИС Личный кабинет: **sudo dnf install nodejs**

Проверяем версию установленной среды выполнения: **node -v**

3.3.2 Запуск и мониторинг сервисов

Для поддержания непрерывной работы сервисов на Node.JS необходимо установить менеджер процессов pm2: **npm install pm2@latest -g**

Проверяем версию установленного pm2: **pm2 -v**

Затем устанавливаем модуль ротации журналов для pm2 и настраиваем его:

pm2 install pm2-logrotate

pm2 set pm2-logrotate:max_size 10M

pm2 set pm2-logrotate:compress true

Первый запуск сервиса производится командой **pm2 start ecosystem.config.js** выполненной из каталога, в котором размещён сервис от имени пользователя, который будет являться владельцем сервисов (например webuser).

В дальнейшем:

Запуск сервиса: **pm2 start <appname>**

Перезапуск сервиса: **pm2 restart <appname>**

Остановка сервиса: **pm2 stop <appname>**

Просмотр логов сервиса: **pm2 logs <appname>**

Где **<appname>** - имя приложения, описанной в файле конфигурации **ecosystem.config.js**

Настройка автозапуска сервиса после перезагрузки сервиса:

pm2 save – сохраняем список сервисов для перезапуска

pm2 startup – включаем автозапуск, для завершения настройки необходимо будет выполнить команду, полученную в результате запуска.

4. Настройка сервисов системы.

4.1 Файл конфигурации ecosystem.config.js

Файл конфигурации **ecosystem.config.js** содержит необходимые настройки для управления сервисом и его подключения к серверам Rabbit MQ, серверам баз данных.

4.1.1 Параметры конфигурации для front-end сервера.

name: 'cabinet.front', – имя приложения

'CABINET_PORT': 7100, – порт для запуска приложения

'CABINET_HOST': 'http://192.168.1.1', - IP-адрес сервера, где запущен сервис

'RABBIT_URL': 'amqp://username:password@hostname:5672/vhost', - строка подключения к Rabbit MQ

'RABBIT_QUEUE_OUTCOME': 'to_int', - очередь исходящих сообщений Rabbit MQ

'RABBIT_QUEUE_INCOME': 'to_ext', очередь входящих сообщений Rabbit MQ

'HTTP_PROXY': 'http://192.168.1.100:3128', - адрес прокси-сервиса (при необходимости)

'HTTPS_PROXY': 'http://192.168.1.100:3128', - адрес прокси-сервиса (при необходимости)
'EXTERNAL_URL': 'https://lk.domain.ru', - адрес, по которому Личный кабинет будет доступен из сети Интернет

'EXTERNAL_PORT': 443 - порт, по которому Личный кабинет будет доступен из сети Интернет

4.1.2 Параметры конфигурации для back-end сервера.

name: 'cabinet.back', – имя приложения

'DB_ORACLE_CONN_STR': 'localhost:1521/lkora', - строка подключения к БД Oracle (если используется)

'DB_ORACLE_USER': 'USER_LKK', - пользователь для подключения к БД Oracle (если используется)

'DB_ORACLE_PASSWORD': 'PASSWORD', - пароль пользователя подключения к БД Oracle (если используется)

'RABBIT_URL': 'amqp://username:password@hostname:5672/vhost', - строка подключения к Rabbit MQ

'RABBIT_QUEUE_OUTCOME': 'to_ext', - очередь исходящих сообщений Rabbit MQ

'RABBIT_QUEUE_INCOME': 'to_int', очередь входящих сообщений Rabbit MQ

4.1.3 Применение настроек

После изменений файла конфигурации необходимо пересоздать сервис командами:

pm2 delete – удаляем сервис

pm2 start ecosystem.config.js - выполнить из каталога, в котором размещён сервис.

4.2 Настройка firewall для front-end сервера

Открыть порты для подключения к сервисам на firewall-е (в данном случае приложение на порту 7100/tcp):

- добавляем порт

sudo firewall-cmd --permanent --zone=public --add-port=7100/tcp

- перезапускаем firewall

sudo firewall-cmd --reload

- проверяем включение правила

sudo firewall-cmd --list-all