

---

**Личный кабинет пациента.  
Инструкция по установке.**

**2023, Москва**

## Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
1.1	Назначение и функции комплекса.	4
1.2	Компоненты комплекса.	4
1.3	Термины и определения	4
<b>2</b>	<b>Требования для установки комплекса</b>	<b>5</b>
2.1	Требования к квалификации специалиста	5
2.2	Требования к аппаратному обеспечению	5
2.3	Необходимое ПО	5
2.4	Требования к каналам связи	6
2.5	Комплектность инсталляционного пакета	6
<b>3</b>	<b>Настройка системы</b>	<b>6</b>
3.1	Предварительное планирование инсталляции	6
3.2	Установка и конфигурация PostgreSQL	7
3.2.1	Установка PostgreSQL в ручном режиме	7
3.2.2	Конфигурация PostgreSQL	7
3.3	Установка и настройка redis cluster	8
3.3.1	Установка redis server	8
3.3.2	Настройка redis cluster	8
3.4	Установка и запуск системы	9
3.4.1	Загрузка образов	9
3.4.2	Подготовка конфигураций helm. Обязательные параметры.	10
3.4.3	Подготовка конфигураций helm. Интеграционные параметры.	10
3.4.4	Подготовить доступ к docker репозиторий	12
3.4.5	Установка helm-чартов	13
3.4.6	Проверка функционирования	13
3.4.7	Дополнительные опциональные настройки	13
<b>4</b>	<b>Описание интеграций личного кабинета пациента</b>	<b>14</b>
<b>5</b>	<b>Список сокращений</b>	<b>16</b>

# 1 Введение

Данный документ содержит порядок установки и настройки личного кабинета пациента (далее по тексту - «системы»).

## 1.1 Назначение и функции комплекса.

Система предназначена для упрощения взаимодействия между пациентом бизнесом компании посредством онлайн сервисов.

Основными целями применения системы являются:

- организация доступа в личный кабинет;
- организации взаимодействия пациента с онлайн сервисами;
- организации взаимодействия между пациентом и внешними сервисами, такими как банковские, госуслуги.

## 1.2 Компоненты комплекса.

- ЛКП Личный кабинет пациента
- ЛКАП Личный кабинет- административная панель.

## 1.3 Термины и определения

**НТТР** - HyperText Transfer Protocol — протокол передачи гипертекста. Протокол прикладного уровня передачи данных.

## 2 Требования для установки комплекса

Для установки и настройки системы существуют обязательные минимально необходимые требования.

### 2.1 Требования к квалификации специалиста

Настройку и установку всех компонентов и подсистем, входящих в систему должен выполнять технический специалист, имеющий соответствующую квалификацию по установке серверных систем.

Технический специалист должен отвечать следующим квалификационным требованиям:

- иметь опыт администрирования, Linux, Kubernetes, Docker;
- владеть знаниями в области сетевых технологий;
- иметь знания в области Java-технологий;
- владеть методами системного администрирования.

### 2.2 Требования к аппаратному обеспечению

#### Сервер приложений ЛКП

Аппаратные требования

- HDD: от 350GB 15K RPM (300 Гб под данные, 30 Гб системный диск).
- CPU: Intel® Xeon® Silver 4208 с частотой 2,4 ГГц
- RAM: от 60GB DDR4.

#### Сервер СУБД:

- HDD: от 350GB 15K RPM (300 Гб под данные, 30 Гб системный диск).
- CPU: Intel® Xeon® Silver 4208 с частотой 2,4 ГГц
- RAM: от 60GB DDR4.

### 2.3 Необходимое ПО

Следующее ПО необходимо установить перед основной установкой компонентов системы:

- OS Ubuntu 20.04.
- Apache Tomcat 9.
- Redis cluster 6.0.3.
- JDK: openjdk 11.
- PostgreSQL 14.
- Kubernetes 1.25 и выше.
- docker версии 1.2 и выше.

## 2.4 Требования к каналам связи

Способность систем к обмену данными должна учитывать возможность использования каналов со скоростью передачи не выше 100 Мбит/с.

## 2.5 Комплектность инсталляционного пакета

Инсталляционный пакет состоит из:

- Установка и конфигурация PostgreSQL;
- Установка и конфигурация Redis;
- Запуск и конфигурация сервисов;

## 3 Настройка системы

### 3.1 Предварительное планирование инсталляции

Приложение ЛК состоит из набора микросервисов, которые работают в кластере kubernetes.

Приложение поставляется в виде набора докер-образов. Перечень поставляемых образов:

- *admin-backend-v29924.tar.gz*;
- *database-updater-v29924.tar.gz*;
- *esb-subscriber-v29924.tar.gz*;
- *rest-api-v29924.tar.gz*;
- *rest-old-api-v29924.tar.gz*;
- *site-backend-v29924.tar.gz*;
- *site-frontend-v29924.tar.gz*.

Для работы приложения требуется БД postgres и кластер redis.

Предполагается, что кластер kubernetes имеет версию 1.25+, в нем установлен ingress-контроллер с ingressClass=nginx и к этому контроллеру привязано имя \*.lk3.my-ingress-domain.k8s.

Также должен существовать внутренний реестр образов (docker registry), расположенный по адресу docker-registry.mycompany.ru и должен быть доступ к нему на чтение и запись. В ходе работы с Docker образы отправляют в реестр и загружают из него. Адрес ссылки хранилища: «docker-registry.mycompany.ru»

Должен существовать сервис Active Directory для домена mycompany.ru по адресу ad-host1.

Обязательные условия, по настройке окружения, требующиеся для запуска образов сведены в таблицу 1.

*Таблица 1. Обязательные настройки окружения*

---

Параметр окружения	Примечание
mycompany.ru	Название локального домена.
ad-host1	Название хоста контроллера домена mycompany.ru.
postgreshost1	Название хоста сервера СУБД PostgreSQL.
6432	Порт для подключения к СУБД PostgreSQL.
«lk»	Название базы данных (Database) в СУБД PostgreSQL.
lk-user	Имя пользователя для подключения к СУБД PostgreSQL.
lk-password	Пароль для подключения к СУБД PostgreSQL.
redis1	Название хоста сервера redis.
6379	Порт для подключения к серверу redis.

## 3.2 Установка и конфигурация PostgreSQL

Руководство по установке расположено по адресу:

<https://www.postgresql.org/docs/current/install-procedure.html/>

### 3.2.1 Установка PostgreSQL в ручном режиме

Установка выполняется на Ubuntu версии "20.04". Сервер должен быть подключен к сети интернет.

Откройте консоль сервера, либо непосредственно, либо удаленно по сети. Все действия будут описываться из условия выполнения в консоли.

Выполните команду

```
~# sudo apt install postgresql
```

Инсталляция должна пройти корректно без ошибок.

### 3.2.2 Конфигурация PostgreSQL

Выполните несколько действий по настройке сервера:

- Переключитесь на суперпользователя (~# **sudo su** -).
- Перейдите в каталог /etc/postgresql/14/main/.
- Измените файл конфигурации postgresql.conf.
  - Добавьте в конец строку «**listen\_addresses = '\*'**».
- Перейдите в каталог /etc/postgresql/14/main/.
- Измените файл конфигурации pg\_hba.conf.
  - Добавьте в конец строку «**host all all 192.168.1.0/24 trust**», где 192.168.1.0/24 подсеть клиента базы данных.

Разрешите доступ к порту базы данных (~# **sudo ufw allow 5432/tcp**).

Перезапустите сервер базы данных (~# **sudo service postgresql restart**).

Запустите утилиту psql (~# **sudo -u postgres psql**) и создайте пользователя системы выполнив команды:

- Создаем пользователя (Postgres=> **CREATE ROLE lk-user WITH LOGIN CREATEDB CREATEROLE;**).
- Задайте пароль «lk-password» (Postgres=> **\password lk**) введя его после запроса.
- Выйдите из psql (Postgres=> **\q**).

Зайдите в **psql** под пользователем «lk» и создайте базу данных. Для этого выполните команды:

- Запустите **psql** (~# **psql -U lk -h localhost -W -d postgres**).
- Создайте базу данных командой (Postgres=>**CREATE DATABASE lk WITH ENCODING = 'UTF8' OWNER = lk-user;**).
- Выйдите из **psql** (Postgres=> **\q**).

Проверьте соединение с созданной БД (~# **psql -U lk-user -h localhost -W**).

После ввода пароля получите приглашение «lk=>».

На последнем этапе убедиться, что адрес хоста с сервером БД называется «**postgreshost1**».

### 3.3 Установка и настройка redis cluster

#### 3.3.1 Установка redis server

Установите репозиторий redis.

```
sudo add-apt-repository ppa:chris-lea/redis-server
```

Обновите все пакеты репозитория.

```
sudo apt-get update
```

Установите redis.

```
sudo apt-get install redis-server
```

Убедитесь в том, что сервер Redis установлен и запущен.

```
redis-benchmark -q -n 1000 -c 10 -P 5
```

Запустите для проверки тест.

```
redis-benchmark -t ping, Set, get -n 1000000 --csv
```

В результате получите на экране результаты тестов.

#### 3.3.2 Настройка redis cluster

В случае, если необходим кластер, то настройка кластера redis в минимальной типовой конфигурации описана по ссылке (см. <https://redis.io/docs/management/scaling/>).

Для завершения настройки требуется проверить и дополнить файл конфигурации **redis**.

Откройте на изменение файл конфигурации `/etc/redis/redis.conf`.

Найдите строку, которая начинается с «port». Должен быть указан порт 6379. Если это не так, то исправьте на стандартный: «port 6379».

Посмотрите ip адрес сетевого интерфейса (команда `ifconfig`).

Найдите в `redis.conf` строку «bind 127.0.0.1 ::1» и дополните ее адресом сетевого интерфейса. Например, адрес интерфейса 192.168.0.15, тогда строка должна выглядеть следующим образом: «bind 127.0.0.1 ::1 192.168.0.15».

Перезапускаем сервис командой «**systemctl restart redis-server**»

Проверьте или внесите запись в локальный DNS сервер. Имя сервера должно быть «**redis1**».

Проверка DNS: На любом компьютере в сети запустите команду «**ping redis1**». Должен вернуться адрес сетевого интерфейса.

## 3.4 Установка и запуск системы

### 3.4.1 Загрузка образов

Образы поставляются в виде `tar.gz` архивов, которые можно загрузить в локальный docker репозиторий.

Пример содержания директории поставки:

```
$ ls
```

```
admin-backend-v29924.tar.gz
```

```
database-updater-v29924.tar.gz
```

```
esb-subscriber-v29924.tar.gz
```

```
rest-api-v29924.tar.gz
```

```
rest-old-api-v29924.tar.gz
```

```
site-backend-v29924.tar.gz
```

```
site-frontend-v29924.tar.gz
```

В имени каждого файла образа содержится версия его сборки, например, `v29924` - означает сборку с номером 29924.

Назначение файлов образов:

- *admin-backend-v29924.tar.gz* – серверная часть ЛКПАП;
- *database-updater-v29924.tar.gz* – сервис обновления БД ЛКП.;
- *esb-subscriber-v29924.tar.gz* – сервис шинного подписчика;
- *rest-api-v29924.tar.gz* – сервис реализации API ЛКАП;
- *rest-old-api-v29924.tar.gz* – сервис реализации API ЛКП (Старая версия);
- *site-backend-v29924.tar.gz* – серверная часть ЛКП;

- *site-frontend-v29924.tar.gz* – frontend ЛКП.

Чтобы загрузить каждый образ, выполнить для каждого файла команду `docker load -i <имя файла>`, например,

```
docker load -i admin-backend-v29924.tar.gz
```

Данную процедуру требуется выполнить для каждого из образов.

### 3.4.2 Подготовка конфигураций helm. Обязательные параметры.

Подготовка конфигураций выполняется путем внесения изменения в следующие файлы:

- «~/charts/admin-backend/values-added.yaml»
- «~/charts/database-updater/values-added.yaml»
- «~/charts/esb-subscriber/values-added.yaml»
- «~/charts/rest-api/values-added.yaml»
- «~/charts/rest-old-api/values-added.yaml»
- «~/charts/site-backend/values-added.yaml»
- «~/charts/site-frontend/values-added.yaml»

Значения всех нижеуказанных переменных окружения требуется проверить и изменить на текущие значения, если требуется.

LK3\_DATABASE\_URL: jdbc:postgresql://postgreshost1:6432/lk?charSet=UTF-8&prepareThreshold=0

Строка соединения с БД ЛКП, где:

- `postgreshost1` - Название хоста сервера БД (возможно использование IP адреса).
- `6432` – Порт подключения.
- `Lk` – Название БД к которой выполняется подключение.
- `UTF-8` – Кодировка БД к которой выполняется подключение.

LK3\_DATABASE\_USERNAME: lk-user

Имя пользователя для подключения к БД

LK3\_DATABASE\_PASSWORD: lk-password

Пароль пользователя для подключения к БД

LK3\_REDIS\_NODES: redis1:6379

Настройка подключения к redis. Формат «хост: порт».

### 3.4.3 Подготовка конфигураций helm. Интеграционные параметры.

Данные переменные окружения описывают интеграцию с другими системами, но используемыми личным кабинетом пациента.

В случае если не указать адрес, либо неправильно указать адрес, либо адрес недоступен, то взаимодействие с данной системой невозможно, и некоторый функционал личного кабинета пациента будет недоступен.

LK3\_INTEGRATION\_ARMPS\_CART\_API\_URL: [https://armps\\_cart\\_api\\_url](https://armps_cart_api_url)

Адрес API АРМПС. Используется для интеграции с АРМПС.

Ограничения в случае недоступности:

- Невозможно оформить заказ.

LK3\_INTEGRATION\_ELECTRONIC\_ARCHIVE\_URL: [https://electronic\\_archive\\_url](https://electronic_archive_url)

Адрес API сервиса «Электронный Архив». Предназначен для хранения электронных документов пациента.

Ограничения в случае недоступности:

- Невозможно просмотреть или скачать документы пациента.

LK3\_INTEGRATION\_EMAIL\_SERVICE\_URL: [https://email\\_service\\_url](https://email_service_url)

Адрес API сервиса отправки почты. Используется для отправки уведомлений пользователям.

Ограничения в случае недоступности:

- Невозможно отправить уведомление пользователю.

LK3\_INTEGRATION\_ESB\_OUTGOIN\_URL: [https://esb\\_outgoin\\_url](https://esb_outgoin_url)

Адрес для взаимодействия с интеграционной шиной ESB. Используется для синхронизации данных приложения.

Ограничения в случае недоступности:

- Нет обновления данных.

LK3\_INTEGRATION\_CONTACT\_HISTORY\_URL: [https://contact\\_history\\_url](https://contact_history_url)

Адрес API истории контакта. Используется для интеграции с историей контакта.

Ограничения в случае недоступности:

- Недоступны все заказы (заказы, оформленные в АРМПС) в разделе «Заказы».

LK3\_INTEGRATION\_BANK\_SERVICE\_BASE\_URL: [https://bank\\_service\\_base\\_url](https://bank_service_base_url)

Адрес API банковского сервиса. Используется для онлайн оплаты заказа.

Ограничения в случае недоступности:

- Недоступна онлайн оплата заказа.

*LK3\_INTEGRATION\_GBQ\_PROJECT\_NAME: "*

*LK3\_INTEGRATION\_GBQ\_DATASET\_NAME: "*

*LK3\_INTEGRATION\_GBQ\_TABLE\_NAME: "*

*LK3\_INTEGRATION\_GBQ\_TOKEN\_PROJECT\_ID: invitro-bi-owox*

*LK3\_INTEGRATION\_GBQ\_TOKEN\_PRIVATE\_KEY\_ID: '\*\*\*\*'*

`LK3_INTEGRATION_GBQ_TOKEN_PRIVATE_KEY_BASE64: '****'`  
`LK3_INTEGRATION_GBQ_TOKEN_CLIENT_EMAIL: '****'`  
`LK3_INTEGRATION_GBQ_TOKEN_CLIENT_ID: '****'`

Настройка интеграции с Google Big Query. Используется для сбора статистики.

Ограничения в случае недоступности:

- Не будет собираться статистика.

`LK3_INTEGRATION_LOYALTY_URL: https://loyalty_url`

Адрес API лояльности. Используется для взаимодействия с системой лояльности.

Ограничения в случае недоступности:

- Недоступно подключение или отключение программ лояльности.
- Недоступна информация по состоянию бонусного счета.
- Недоступна информация по состоянию подключенных программ лояльности.
- Недоступно получение бонусов и скидков при оформлении заказа.

`LK3_INTEGRATION_PHONE_URL: https://phone\_url`

Адрес API сервиса отправки СМС. Используется для отправки уведомлений пользователям.

Ограничения в случае недоступности:

- Невозможно отправить уведомление пользователю.

`LK3_INTEGRATION_STORAGE_RESULT_URL: https://storage_result_url`

Адрес API сервиса «Хранилище результатов». Предназначен для хранения результатов.

Ограничения в случае недоступности:

- Невозможно просмотреть или скачать результаты.
- Невозможно просмотреть динамику результатов.

`LK3_INTEGRATION_TRANSLATE_URL: https://translate\_url`

Адрес API сервиса переводов. Предназначен для перевода результатов анализа.

Ограничения в случае недоступности:

- Невозможно перевести результаты анализов.

#### **3.4.4 Подготовить доступ к docker репозиторий**

Предполагается, что микросервисы будут установлены в пространство имен «invitro-lk» и в этом окружении есть секретный ключ для доступа к docker registry с именем global-docker-registry-secret

### 3.4.5 Установка helm-чартов

Выполняется с помощью `helm upgrade`. В каждой команде установить `image.tag` в номер версии образа, загруженного docker-репозиторий.

Команды Установки:

- `helm upgrade --set imagePullSecrets[0].name=global-docker-registry-secret --set image.tag=29924 --install --values values.yaml --values values-added.yaml --debug database-updater charts/database-updater --namespace=invitro-lk --atomic --timeout 10m`
- `helm upgrade --set ingress.hostname=admin-backend.lk.my-ingress-domain.k8s --set imagePullSecrets[0].name=global-docker-registry-secret --set image.tag=29924 --install --values values.yaml --values values-added.yaml --debug admin-backend charts/admin-backend --namespace=invitro-lk --atomic --timeout 10m`
- `helm upgrade --set ingress.hostname=esb-subscriber.lk.my-ingress-domain.k8s --set imagePullSecrets[0].name=global-docker-registry-secret --set image.tag=29924 --install --values values.yaml --values values-added.yaml --debug esb-subscriber charts/esb-subscriber --namespace=invitro-lk --atomic --timeout 10m`
- `helm upgrade --set ingress.hostname=rest-api.lk.my-ingress-domain.k8s --set imagePullSecrets[0].name=global-docker-registry-secret --set image.tag=29924 --install --values values.yaml --values values-added.yaml --debug rest-api charts/rest-api --namespace=invitro-lk --atomic --timeout 10m`
- `helm upgrade --set ingress.hostname=rest-old-api.lk.my-ingress-domain.k8s --set imagePullSecrets[0].name=global-docker-registry-secret --set image.tag=29924 --install --values values.yaml --values values-added.yaml --debug rest-old-api charts/rest-old-api --namespace=invitro-lk --atomic --timeout 10m`
- `helm upgrade --set ingress.hostname=site-backend.lk.my-ingress-domain.k8s --set imagePullSecrets[0].name=global-docker-registry-secret --set image.tag=29924 --install --values values.yaml --values values-added.yaml --debug site-backend charts/site-backend --namespace=invitro-lk --atomic --timeout 10m`
- `helm upgrade --set ingress.hostname=site-frontend.lk.my-ingress-domain.k8s --set imagePullSecrets[0].name=global-docker-registry-secret --set image.tag=29924 --install --values values.yaml --values values-added.yaml --debug site-frontend charts/site-frontend --namespace=invitro-lk --atomic --timeout 10m`

### 3.4.6 Проверка функционирования

Зайдите браузером по адресу, указанному в поле `host ingress` сервиса `site-frontend`, например: <http://site-frontend.lk.my-ingress-domain.k8s>.

### 3.4.7 Дополнительные опциональные настройки

Для удобства использования можно установить `nginx`, который будет перенаправлять запросы к приложению ЛК. Пример конфигурации - в каталоге `nginx`.

## 4 Описание интеграций личного кабинета пациента

Личный кабинет пациента это одна из частей набора систем компании. Взаимодействие с другими системами осуществляется либо посредством интеграционного API, либо посредством шины ESB.

Личный кабинет пациента не хранит у себя данные, которые не обрабатывает. Подобные данные в случае необходимости запрашиваются у систем владельцев.

Перечень систем, не входящих в комплект поставки, с которыми осуществляется прямая интеграция личного кабинета пациента:

- АРМПС;
- «Электронный Архив»;
- «Сервис отправки почты»;
- ESB интеграционная шина;
- «Сервис истории контакта»;
- «Банковский сервиса»;
- Google Big Query (сбор и анализ статистики);
- «Система лояльности»;
- «Сервиса отправки СМС»;
- «Хранилище результатов»;
- «Сервиса переводов».

Перечень последствий из-за отсутствия интеграции в рамках личного кабинета пациента.

- АРМПС. Последствия отсутствия интеграции:
  - невозможно оформить заказ;
- «Электронный Архив». Последствия отсутствия интеграции:
  - невозможно просмотреть или скачать документы пациента;
- «Сервис отправки почты». Последствия отсутствия интеграции:
  - невозможно отправить уведомление пользователю;
- ESB интеграционная шина. Последствия отсутствия интеграции:
  - недоступно обновление справочников;
- «Сервис истории контакта». Последствия отсутствия интеграции:
  - недоступны все заказы (заказы, оформленные в АРМПС) в разделе «Заказы»;
- «Банковский сервиса». Последствия отсутствия интеграции:
  - недоступна онлайн оплата заказа;
- Google Big Query (сбор и анализ статистики). Последствия отсутствия интеграции:
  - не будет собираться статистика;
- «Система лояльности». Последствия отсутствия интеграции:
  - недоступно подключение или отключение программ лояльности;
  - недоступна информация по состоянию бонусного счета;
  - недоступна информация по состоянию подключенных программ лояльности;
  - недоступно получение бонусов и скидок при оформлении заказа;
- «Сервиса отправки СМС». Последствия отсутствия интеграции:
  - невозможно отправить уведомление пользователю;
- «Хранилище результатов». Последствия отсутствия интеграции:
  - невозможно просмотреть или скачать результаты;

- невозможно просмотреть динамику результатов;
- «Сервиса переводов». Последствия отсутствия интеграции:
- невозможно перевести результаты анализов.

## 5 Список сокращений

Таблица 2. Список сокращений

<b>Сокращение</b>	<b>Описание</b>
<b>ЛКП</b>	Личный кабинет пациента.
<b>ЛКАП</b>	Личный кабинет- административная панель
<b>БД</b>	База данных
<b>СУБД</b>	Система управления базами данных
<b>ESB</b>	ESB (enterprise service bus)- интеграционная или сервисная шина. Программное обеспечение, благодаря которому возможен обмен данными между разными информационными системами предприятия.
<b>API</b>	API –Application Programming Interface, или программный интерфейс приложения.
<b>АРМПС</b>	Автоматизированное рабочее место процедурной сестры, основное программное обеспечение, с которым работает сотрудник медицинского офиса при оформлении заказов.