|  |
| --- |
| Logo_mail |
| МЕДИЦИНСКАЯ ИНФОРМАЦИОННАЯ СИСТЕМА |
|  |
| ПРОГРАММНЫЙ МОДУЛЬ |
| АРМ ЛАБОРАТОРИЯ  |
| РУКОВОДСТВО ПО УСТАНОВКЕ |
|  |

# Варианты дистрибутивов

Дистрибутив программного модуля состоит из 4 архивов, объединенных единой версией.

Состав дистрибутива:

* + "apps-linux" - дистрибутив desktop-приложения ЛИС для развертывания в среде операционных систем Linux
	+ "apps-windows" - дистрибутив desktop-приложения ЛИС для развертывания в среде операционных систем Windows
	+ "core-docker"- дистрибутив серверной части ЛИС для развертывания в среде Docker под управлением операционных систем Centos 7/8
	+ "core-windows"- дистрибутив серверной части ЛИС для развертывания в среде операционных систем Windows

# Серверная часть ЛИС

Корректная работа системы обеспечивается использованием следующего программного обеспечения:

* HTTP-сервер
Рекомендуется использование Nginx. В документации рассматривается использование HTTP-сервера Nginx.
* СУБД PostgreSQL (версия не ниже 11)

# Nginx

Основная функция Nginx - обеспечить маршрутизацию http-запросов между модулями системы по контексту.

Маршрутизация внутри приложения построена на использовании контекстов. Используются следующие значения:

* /backend-icm/ - проксирование запросов к модулю \_backend-icm. Требуется указание IP-адреса и порта, на котором запущен jar-файл приложения.
* /backend-main/ - проксирование запросов к модулю *backend-main*. Требуется указание IP-адреса и порта, на котором запущен jar-файл приложения.
* /backend-warehouse/ - проксирование запросов к модулю *backend-warehouse*. Требуется указание IP-адреса и порта, на котором запущен jar-файл приложения.
* /main/ - обработка запросов к модулю *ui-main*. Требуется указание пути к папке с содержимым dist для ui-main.
* /warehouse/ - обработка запросов к модулю *ui-warehouse*. Требуется указание пути к папке с содержимым dist для ui-warehouse.
* /docs/ - обработка запросов к модулю *docs*. Требуется указание пути к папке c содержимым для docs.

По умолчанию используются следующие порты:

* nginx - порт 8090
* backend-main - порт 8091
* backend-warehouse - порт 8092
* backend-icm - порт 8093

Порт nginx задается в конфигурационном файле <папка установки>\nginx\conf\nginx.conf.

Порты, на которых запускаются backend-модули, задаются в скрипте newlis-services-install.cmd, который используется для создания служб Windows.

Порты, на которые делается маршрутизация из nginx в backend-модули, задаются в конфигурационных файлах <папка установки>\nginx\conf\bregis-lis.conf.d\backend-\*.conf.

# PostgreSQL

Для работы системы требуется наличие установленной СУБД [PostgreSQL](https://www.enterprisedb.com/downloads/postgres-postgresql-downloads) 11 и выше.

Во время установки PostgreSQL все параметры нужно оставить по умолчанию, только изменить порт с 5432 на 5433. Пароль для пользователя postgres назначается любой, по усмотрению пользователя.

После установки PostgreSQL необходимо зайти в папку C:\Program Files\PostgreSQL\11\data и открыть файл pg\_hba.conf. В нём добавить правило на доступ из подсети, чтобы к БД можно было подключиться, и перезапустить службу PostgreSQL. Пример добавляемой строки в pg\_hba.conf:

# TYPE DATABASE USER ADDRESS METHOD

# IPv4 local connections:

host all all 127.0.0.1/32 md5

host all all 192.168.44.0/24 md5

где \* 192.168.44.0 - подсеть \* 24 - [маска подсети](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D0%BA%D0%B0_%D0%BF%D0%BE%D0%B4%D1%81%D0%B5%D1%82%D0%B8)

Для отключения ограничений по IP имеется возможность задать единственное правило:

 host all all 0.0.0.0/0 md5

Для запуска системы необходимо создание базы данных, пользователя-владельца базы данных (lis\_admin) и пользователя, от имени которого будет вестись работа приложений системы с базой данных (lis\_user). Используемая кодировка - UTF8.

Стандартные значения, используемые в системе:

POSTGRES\_HOST=127.0.0.1

POSTGRES\_PORT=5433

POSTGRES\_DB=bregis\_lis

POSTGRES\_USER=lis\_admin

POSTGRES\_PASSWORD=Bre91s\_Admin

POSTGRES\_SIMPLE\_USER=lis\_user

POSTGRES\_SIMPLE\_PASSWORD=Bconnect\_2

Пример скрипта создания пользователей и базы данных вручную (актуально для [развертывания в Windows](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B2-%D1%81%D1%80%D0%B5%D0%B4%D0%B5-Windows), в Docker все действия выполняются скриптом):

* Выполнять от имени суперадминистратора, подключенного к любой БД (на пустом PostgreSQL имеется БД postgres):
	+ [Создание пользователя-владельца БД](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B):

**CREATE** **ROLE** lis\_admin NOINHERIT LOGIN PASSWORD 'Bre91s\_Admin';

* + [Создание пользователя для работы приложений с БД](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B)

**CREATE** **ROLE** lis\_user NOINHERIT LOGIN PASSWORD 'Bconnect\_2';

* + [Создание БД](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B):

**CREATE** **DATABASE** ariadna\_lis

**WITH** **OWNER** = lis\_admin

**ENCODING** = 'UTF8'

LC\_COLLATE = 'C'

LC\_CTYPE = 'en\_US.utf8'

**TEMPLATE** = template0;

***Обратите внимание!*** При создании БД обязательно должно быть указано значение параметра **LC\_COLLATE = 'C'**

* Выполнять от имени суперадминистратора, подключенного к созданной БД (в примере ariadna \_lis):
	+ [Выдача прав для пользователя lis\_admin](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B):

GRANT CREATE

 **ON** **SCHEMA** **public** **TO** lis\_admin;

GRANT USAGE

 **ON** **SCHEMA** **public** **TO** lis\_admin;

GRANT CONNECT

 **ON** **DATABASE** ariadna \_lis **TO** lis\_admin;

GRANT CREATE

 **ON** **DATABASE** ariadna \_lis **TO** lis\_admin;

GRANT TEMP

 **ON** **DATABASE** ariadna \_lis **TO** lis\_admin;

* + Миграции и выдача прав пользователю lis\_user выполняются в рамках отдельных backend-модулей.

Рекомендуется включить логирование PostgreSQL.
Для PostgreSQL 11 заходим в папку C:\Program Files\PostgreSQL\11\data , открываем postgresql.conf и в нём ищём соответствующий параметр:

logging\_collector=on

По умолчанию postgresql настроен на поддержание 100 одновременных соединений, 3 из которых зарезервированы для Суперпользователей для выполнения административных работ с СУБД (см. [superuser\_reserved\_connections](https://www.postgresql.org/docs/11/runtime-config-connection.html)).

Модули системы настроены на использование определенного количества соединений. Для настройки по умолчанию в 100 соединений ситуации могут быть следующие:

* "В покое" (АРМы не запущены, в WEB никто не работает) активно минимум 35 соединений к БД.
Расчет: (1+2+10)+(1+10)+(1+10) = 35
* При малоинтенсивной работе (АРМы запущены, но не выполняют активной выборки данных, в WEB не ведется работа с инструментами, меняющими структуру БД) могут быть запущены до 31 АРМа.
Расчет: (100-3-35)/2 = 62/2 = 31
* При высокой интенсивности работы (в АРМах идет активный отбор данных, в WEB ведется работа с инструментами, меняющими структуру БД) могут быть запущены до 12 АРМов.
Расчет: (100-3-38)/5 = 59/5 = ~12

# Установка в среде Windows

Создание служб при развертывании в среде Windows осуществляется утилитой nssm.exe.
При развертывании создаются следующие службы:

* NEWLIS-nginx (используется nginx версии 1.15.12)
* NEWLIS-backend-icm
* NEWLIS-backend-main
* NEWLIS-backend-warehouse

PostgreSQL 11 должен быть установлен и настроен отдельно (см. раздел [PostgreSQL](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%22%20%5Cl%20%22PostgreSQL)). Значения по умолчанию для подключения к PostgreSQL заданы в скрипте установки служб newlis-services-install.cmd. При необходимости значения могут быть изменены.
OpenJDK 11 должен быть установлен отдельно. По умолчанию скрипт установки службы использует команду "java.exe", она должна быть доступна [в переменных окружения](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B):

Панель управления - Система - Дополнительно - Переменные среды - Системные переменные - Ищем "Path" - Нажимаем изменить - Добавляем путь до java.exe, например, "C:\Program Files\Java\jdk-11\bin"

В противном случае в скрипте установки службы необходимо задать полный путь до java.exe из состава [OpenJDK 11](https://jdk.java.net/java-se-ri/11), изменив значение переменной JAVAEXE в файле newlis-services-install.cmd.

Для удобства развертывания системы в среде Windows формируется архив newlis-...-core-windows.zip, содержащий скомпилированные модули системы и вспомогательные конфигурационные файлы [Структура архива](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B):

├── distr

│ ├── backend-icm.jar

│ ├── backend-main.jar

│ ├── backend-warehouse.jar

│ ├── docs

│ │ └── /\*статические ресурсы docs\*/

│ ├── ui-main

│ │ └── /\*статические ресурсы ui-main\*/

│ ├── ui-warehouse

│ │ └── /\*статические ресурсы ui-warehouse\*/

│ └── windows

│ ├── newlis-services-install.cmd

│ ├── newlis-services-uninstall.cmd

│ ├── nginx

│ │ └── conf

│ │ ├── bregis\_lis.conf.d

│ │ │ ├── backend-icm.conf

│ │ │ ├── backend-main.conf

│ │ │ ├── backend-warehouse.conf

│ │ │ ├── docs.conf

│ │ │ ├── ui-main.conf

│ │ │ └── ui-warehouse.conf

│ │ └── nginx.conf

│ ├── nginx.zip

│ ├── nssm.exe

│ ├── README.md

│ └── unzip.exe

├── install.cmd

└── INSTALL.md

Для развертывания необходимо выполнить следующие действия:

* скачать архив **newlis-...-core-windows.zip** нужной версии;
* распаковать архив в пустую локальную папку (рекомендуется C:\NEWLIS\);
* запустить install.cmd

cd C:\NEWLIS

start install.cmd

* установить службы (при необходимости предварительно можно поменять параметры, используемые при создании службы - SECRETKEY для авторизации, параметры подключения к PostgreSQL, переопределить другие параметры, задаваемые для каждого backend-модуля в [appliсation.conf](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B))

cd C:\NEWLIS

start newlis-services-install.cmd

Структура рабочей папки имеет следующий вид:

./

├── backend-icm <-- backend-icm.jar, логи backend-icm, конфигурационные файлы при необходимости

├── backend-main <-- backend-main.jar, логи backend-main, конфигурационные файлы при необходимости

├── backend-warehouse <-- backend-warehouse.jar, логи backend-warehouse, конфигурационные файлы при необходимости

├── logs <-- папка содержит символические ссылки до оригинальных папок с логами backend-main, backend-warehouse, nginx

├── nginx <-- рабочая папка nginx (nginx.exe, библиотеки, конфигурационные файлы, логи)

├── static

│ ├── docs <-- пользовательская документация

│ ├── main <-- скомпилированный ui-main

│ └── warehouse <-- скомпилированный ui-warehouse

├── newlis-services-install.cmd <-- скрипт установки служб

├── newlis-services-uninstall.cmd <-- скрипт удаления служб

└── nssm.exe <-- приложения для управления службами Windows

Запуск и остановка служб осуществляется стандартными средствами Windows.

# Установка в среде Docker

Развертывание Docker-контейнерами требует наличия установленных docker и docker-compose.
Для удобства развертывания системы docker-контейнерами формируется архив newlis-...-core-docker.zip, содержащий скомпилированные модули системы и вспомогательные конфигурационные файлы. [Структура архива](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B):

├── distr

│ ├── backend-icm.jar

│ ├── backend-main.jar

│ ├── backend-warehouse.jar

│ ├── docker

│ │ ├── docker-compose.env

│ │ ├── docker-compose.yml

│ │ ├── nginx

│ │ │ └── conf.d

│ │ │ ├── bregis\_lis.conf.d

│ │ │ │ ├── backend-icm.conf

│ │ │ │ ├── backend-main.conf

│ │ │ │ ├── backend-warehouse.conf

│ │ │ │ ├── docs.conf

│ │ │ │ ├── ui-main.conf

│ │ │ │ └── ui-warehouse.conf

│ │ │ └── default.conf

│ │ └── README.md

│ ├── docs

│ │ └── /\*статические ресурсы docs\*/

│ ├── ui-main

│ │ └── /\*статические ресурсы ui-main\*/

│ └── ui-warehouse

│ └── /\*статические ресурсы ui-warehouse\*/

├── INSTALL.md

└── install.sh

Для развертывания необходимо выполнить следующие действия:

* скачать архив **newlis-...-core-docker.zip** нужной версии;
* распаковать архив в пустую локальную папку;
* запустить install.sh

/bin/bash ./install.sh

Помимо построения структуры рабочей директории скрипт запускает контейнер postgresql и создает в нем дополнительного пользователя lis\_user, который необходим для работы системы (см. [PostgreSQL](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%22%20%5Cl%20%22PostgreSQL)).
запустить docker-контейнеры

sudo docker-compose up -d

Структура рабочей папки имеет следующий вид:

./

├── backend-icm <-- backend-icm.jar, логи backend-icm, конфигурационные файлы при необходимости

├── backend-main <-- backend-main.jar, логи backend-main, конфигурационные файлы при необходимости

├── backend-warehouse <-- backend-warehouse.jar, логи backend-warehouse, конфигурационные файлы при необходимости

├── logs <-- папка содержит символические ссылки до оригинальных папок с логами backend-main, backend-warehouse, backend-icm, nginx, postgresql

├── nginx <-- конфигурационные файлы и логи nginx

├── postgres <-- кластер БД PostgreSQL и его конфиги

├── static

│ ├── docs <-- пользовательская документация

│ ├── main <-- скомпилированный ui-main

│ └── warehouse <-- скомпилированный ui-warehouse

├── docker-compose.env <-- переменные окружения docker

└── docker-compose.yml <-- конфигурационный файл docker-compose

Docker-compose использует образы контейнеров из официальных репозиториев (postgres:11.2, nginx:latest, openjdk:11) и не требует создания собственных.

# Обновление в среде Windows

Для обновления системы необходимо выполнить следующие процедуры:

* создать резервную копию базы данных стандартными средствами PostgreSQL;
* скачать архив с дистрибутивом;
* распаковать его в папку системы (например, C:\NEWLIS\);
* выполнить скрипт install.cmd (см. установку) - скрипт перед выполнением остановит службы, после выполнения запустит службы, если они были установлены ранее со стандартным названием (в случае отклонений от стандартных значений следует внести изменения в install.cmd);
* удалить архив.

# Обновление в среде Docker

Для обновления приложения необходимо выполнить следующие процедуры:

* создать резервную копию базы данных стандартными средствами PostgreSQL;
* удалить docker-контейнеры:

sudo docker-compose down

* скачать архив с дистрибутивом, распаковать его, выполнить скрипт install.sh (см. установку), удалить архив;
* запустить docker-контейнеры:

sudo docker-compose up -d

# Пользовательские desktop-приложения ЛИС

# Установка в среде Windows

Необходимо скачать дистрибутив **newlis-...-apps-windows.zip** нужной версии, распаковать его в удобное место, произвести запуск АРМов:

* ArmAdmin - <папка установки>\bin\ArmAdmin.bat
* ArmContent - <папка установки>\bin\ArmContent.bat
* ArmEconom - <папка установки>\bin\ArmEconom.bat
* ArmLab - <папка установки>\bin\ArmLab.bat
* ArmQC - <папка установки>\bin\ArmQC.bat
* ArmSorter - <папка установки>\bin\ArmSorter.bat

После запуска любого из АРМов необходимо произвести настройку подключения к БД и вести привычную работу в АРМах.
Дистрибутив содержит все необходимое для работы АРМов.

# Установка в среде Linux

Необходимо скачать дистрибутив **newlis-...-apps-linux.zip** нужной версии, распаковать его в удобное место, произвести запуск АРМов:

* ArmAdmin - <папка установки>/bin/ArmAdmin.sh
* ArmContent - <папка установки>/bin/ArmContent.sh
* ArmEconom - <папка установки>/bin/ArmEconom.sh
* ArmLab - <папка установки>/bin/ArmLab.sh
* ArmQC - <папка установки>/bin/ArmQC.sh
* ArmSorter - <папка установки>/bin/ArmSorter.sh

После запуска любого из АРМов необходимо произвести настройку подключения к БД и вести привычную работу в АРМах.
Дистрибутив содержит все необходимое для работы АРМов.

# Обновление в среде Windows

Необходимо:

* при необходимости сделать копию конфига <папка установки>\bin\connection.xml
* удалить папку установки
* произвести установку согласно [инструкции по установке](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B#%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B2-%D1%81%D1%80%D0%B5%D0%B4%D0%B5-Windows-2)

# Обновление в среде Linux

Необходимо:

* при необходимости сделать копию конфига <папка установки>/bin/connection.xml
* удалить папку установки
* произвести установку согласно [инструкции по установке](https://redmine.bregis.ru/projects/ariadna-postgres/wiki/%D0%A0%D0%B0%D0%B7%D0%B2%D0%B5%D1%80%D1%82%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%22%20%5Cl%20%22%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B2-%D1%81%D1%80%D0%B5%D0%B4%D0%B5-Linux)