**ИНСТРУКЦИЯ ПО**

**УСТАНОВКЕ СИСТЕМЫ НА СЕРВЕР**

(Alt Linux 8, PostgreSQL)

**Инструкция администратора
 Версия 1**

Нижневартовск 2024

Оглавление

[1. Установка Программного комплекса «Единая медицинская информационная система» на сервер (Alt Linux 8, PostgreSQL) 3](#_Toc169695410)

[1.1. Рекомендуемые требования к серверному оборудованию 3](#_Toc169695411)

[1.2. Последовательность установки 3](#_Toc169695412)

[2. Запуск программы на рабочем месте 16](#_Toc169695413)

[2.1. Клиент системные требования 16](#_Toc169695414)

[3. Учетные данные для входа 16](#_Toc169695415)

[4. Контакты технических специалистов 16](#_Toc169695416)

# **Установка Программного комплекса «Единая медицинская информационная система» на сервер (Alt Linux 8, PostgreSQL)**

# **Рекомендуемые требования к серверному оборудованию**

1. Сервер приложений: процессор 8 ядра, оперативная память 16 Gb, два диска для приложения 300 Gb и для файлов 400 Gb для МО (700 Gb для региона);
2. Сервер БД : процессор 8 ядра, оперативная память 16 Gb, диск для хранения данных 300 Gb для МО (500 Gb для региона);
3. Диски для резервного копирования: 800 Gb для базы данных и 1 Tb

# **Последовательность установки**

1. Установка и настройка операционной системы Alt Linux 8;
2. Установка и настройка сервера баз данных Postgresql;
3. Установка docker;
4. Установка dokku;
5. Установка и настройка приложения ПК «ЕМИС»

Далее следует описание указанных шагов подробно:

1. **Установка и настройка операционной системы Alt Linux 8.**
	1. Устанавливаем операционную систему Alt Linux 8 следуя инструкциям производителя [( https://docs.altlinux.org/ru-RU/archive/3.0/html-single/compact/install/index.html)](http://www.astralinux.ru/).
2. **Установка и настройка сервера баз данных Postgresql.**
	1. Установить PostgreSQL:

*sudo apt-get install postgresql15-server postgresql15-contrib redis*

* 1. Запускаем сервис PostgreSQL:

*sudo systemctl start postgresql*

* 1. Следуя инструкциям производителя PostgreSQL создать новую базу данных ***zdraw*** и пользователя ***postgres*** с полным доступом к базе ***zdraw,*** задать пароль пользователю.

*sudo -u postgres psql
ALTER USER postgres WITH PASSWORD '* *zdraw';
CREATE SCHEMA core OWNER postgres;
CREATE USER zdraw WITH PASSWORD '* *zdraw';
ALTER USER zdraw WITH SUPERUSER;
ALTER DATABASE zdraw OWNER TO zdraw;*

1. **Установка Docker**

Установка следуя инструкциям <https://www.altlinux.org/Docker>

*sudo apt-get update
sudo apt-get install docker-engine
sudo usermod $USER -aG docker
sudo systemctl enable --now docker
sudo systemctl restart docker*

1. **Установка Dokku**
	1. В файле /root/dokku/Makefile закомментировать строки:

*#ifndef CI
# wget -nv -O - https://get.docker.com/ | sh
#ifdef DOCKER\_VERSION
# apt-get -qq -y docker-engine=${DOCKER\_VERSION} || (apt-cache madison docker-engine ; exit 1)
#endif
# sleep 2 # give docker a moment i guess
#endif*

* 1. Команда apt-get в ALT Server не знает ключа "--no-install-recommends", поэтому перед запуском установки dokku необходимо удалить все упоминания данного ключа в файле */root/dokku/Makefile*
	2. Установить Dokku:

*sudo -u root bash
sudo apt-get update
sudo apt-get install wget nginx make jq bind-utils
sudo mkdir /etc/sudoers.d/dokku-nginx
sudo mkdir /etc/sudoers.d/dokku-storage
sudo useradd -g dokku dokku
sudo adduser syslog
sudo groupadd dokku
sudo usermod -aG dokku dokku*

* 1. Указать IP

*sudo mkdir /home/dokku
sudo echo $(hostname -i) > /home/dokku/VHOST*

* 1. Установить dokku

*sudo make install -C /root/dokku*

* 1. Установить плагины

*sudo dokku plugin:install*

* 1. Дать разрешение на папку

*sudo chown dokku:dokku -R /home/dokku*

* 1. Добавить пользователя в группу

*sudo usermod -aG docker $USER
sudo usermod -aG docker dokku
sudo chown dokku:dokku -R /var/lib/dokku/*

* 1. Установить плагины для dokku

*sudo dokku plugin:install* [*https://github.com/dokku/dokku-postgres.git*](https://github.com/dokku/dokku-postgres.git) *sudo dokku plugin:install* [*https://github.com/dokku/dokku-rabbitmq.git*](https://github.com/dokku/dokku-rabbitmq.git) *sudo dokku plugin:install* [*https://github.com/dokku/dokku-redis.git*](https://github.com/dokku/dokku-redis.git) *sudo dokku plugin:install*

1. **Установка и настройка приложения**
	1. **Скачать установщик**
	2. **Core App**
		1. **Скачиваем образ проект:**
		2. *docker load -i core\_1.162.2-1.40.1\_\_bf2.tar*
		3. **Создаем проект:**
		4. *dokku apps:create core*
		5. *dokku redis:create core*
		6. *dokku redis:link core core*
		7. **Добавить проброс папки:**
		8. *dokku storage:mount core /home/dokku/core/static:/app/static*
		9. **Настройка и подключение к БД:**
		10. *dokku config:set --no-restart core DATABASE\_URL='postgres://zdraw:zdraw@localhost5432/zdraw'*
		11. **Установить ENV:**
		12. *dokku config:set --no-restart core CORE\_LOCAL\_PORT=3000*
		13. *dokku config:set --no-restart core CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http:// 192.168.2.45:3000*
		14. *dokku config:set --no-restart core DOKKU\_APP\_RESTORE=1*
		15. *dokku config:set --no-restart core DOKKU\_APP\_TYPE=dockerfile*
		16. *dokku config:set --no-restart core DOKKU\_CHECKS\_DISABLED=\_all\_*
		17. *dokku config:set --no-restart core DOKKU\_PROXY\_PORT=1112*
		18. *dokku config:set --no-restart core DOKKU\_PROXY\_PORT\_MAP=http:3000:5000*
		19. *dokku config:set --no-restart core HOST\_NAME=Test*
		20. *dokku config:set --no-restart core LOCAL\_ADDRESS=192.168.2.45*
		21. *dokku config:set --no-restart core REGION\_ID=92*
		22. *dokku config:set --no-restart core TIME\_ZONE=Asia/Yekaterinburg*
		23. *dokku config:set --no-restart core TZ=+0500*
		24. **Устанавливаем и пересобираем:**
		25. *dokku git:from-image core docker.pkzdrav.ru/zdraw/core-backend:1.164.0-1.40.1\_\_bf2*
		26. *dokku ps:rebuild core*
	3. **Lis App**
		1. **Скачиваем образ проект:**
		2. *docker load -i lis\_1.174.0-1.85.0\_\_bf2.tar*
		3. **Создаем проект:**
		4. *dokku redis:create lis*
		5. *dokku apps:create lis*
		6. *dokku redis:link lis lis*
		7. **Добавить проброс папки:**
		8. *dokku storage:mount lis /home/dokku/lis/static:/app/static*
		9. **Настройка и подключение к БД:**
		10. *dokku config:set --no-restart lis DATABASE\_URL='postgres://zdraw:zdraw@localhost:5432/zdraw'*
		11. **Установить ENV:**
		12. *dokku config:set --no-restart lis APP\_URL\_PREFIX=lis*
		13. *dokku config:set --no-restart lis CORE\_LOCAL\_PORT=3000*
		14. *dokku config:set --no-restart lis CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http://10.10.247.5:3000*
		15. *dokku config:set --no-restart lis DOKKU\_APP\_RESTORE=1*
		16. *dokku config:set --no-restart lis DOKKU\_APP\_TYPE=dockerfile*
		17. *dokku config:set --no-restart lis DOKKU\_CHECKS\_DISABLED=*
		18. *dokku config:set --no-restart lis DOKKU\_PROXY\_PORT\_MAP=http:3003:5000*
		19. *dokku config:set --no-restart lis REGION\_ID=92*
		20. *dokku config:set --no-restart lis TIME\_ZONE=Asia/Yekaterinburg*
		21. *dokku config:set --no-restart lis TZ=+0500*
		22. *dokku config:set --no-restart lis HOST\_NAME=Test*
		23. *dokku config:set --no-restart lis LOCAL\_ADDRESS=192.168.2.45*
		24. **Устанавливаем и пересобираем:**
		25. *dokku git:from-image lis docker.pkzdrav.ru/zdraw/lis-backend:1.174.0-1.86.0\_\_bf*
		26. *dokku ps:rebuild lis*
	4. **Nsi App**
		1. **Скачиваем образ проект:**
		2. *docker load -i nsi-backend\_1.29.3\_\_bf2.tar*
		3. **Создаем проект:**
		4. *dokku apps:create nsi*
		5. **Добавить проброс папки:**
		6. *dokku storage:mount nsi /home/dokku/nsi/static:/app/static*
		7. **Настройка и подключение к БД:**
		8. *dokku config:set --no-restart nsi DATABASE\_URL='postgres://zdraw:zdraw@localhost:5432/zdraw'*
		9. **Установить ENV:**
		10. *dokku config:set --no-restart nsi APP\_URL\_PREFIX=nsi*
		11. *dokku config:set --no-restart nsi CORE\_LOCAL\_PORT=3000*
		12. *dokku config:set --no-restart nsi CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http://10.10.247.5:3000*
		13. *dokku config:set --no-restart nsi DOKKU\_APP\_RESTORE=1*
		14. *dokku config:set --no-restart nsi DOKKU\_APP\_TYPE=dockerfile*
		15. *dokku config:set --no-restart nsi DOKKU\_CHECKS\_DISABLED=\_all\_*
		16. *dokku config:set --no-restart nsi DOKKU\_PROXY\_PORT\_MAP=http:3002:5000*
		17. *dokku config:set --no-restart nsi REGION\_ID=92*
		18. *dokku config:set --no-restart nsi TIME\_ZONE=Asia/Yekaterinburg*
		19. *dokku config:set --no-restart nsi TZ=+0500*
		20. *dokku config:set --no-restart nsi HOST\_NAME=Test*
		21. *dokku config:set --no-restart nsi LOCAL\_ADDRESS=192.168.2.45*
		22. **Устанавливаем и пересобираем:**
		23. *dokku git:from-image nsi docker.pkzdrav.ru/zdraw/nsi-backend:1.29.2\_\_bf*
		24. *dokku ps:rebuild nsi*
	5. **Templates App**
		1. **Скачиваем образ проект:**
		2. *docker load -i templates-backend\_1.40.0\_\_bf2.tar*
		3. **Создаем проект:**
		4. *dokku apps:create templates*
		5. **Добавить проброс папки:**
		6. dokku storage:mount templates /home/dokku/templates/static:/app/static
		7. **Настройка и подключение к БД:**
		8. *dokku config:set --no-restart templates DATABASE\_URL='postgres://zdraw:zdraw@localhost:5432/zdraw'*
		9. **Установить ENV:**
		10. dokku config:set --no-restart templates APP\_URL\_PREFIX=templates
		11. dokku config:set --no-restart templates CORE\_LOCAL\_PORT=3000
		12. dokku config:set --no-restart templates CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http://10.10.247.5:3000
		13. dokku config:set --no-restart templates DOKKU\_APP\_RESTORE=1
		14. dokku config:set --no-restart templates DOKKU\_APP\_TYPE=dockerfile
		15. dokku config:set --no-restart templates DOKKU\_CHECKS\_DISABLED=\_all\_
		16. dokku config:set --no-restart templates DOKKU\_PROXY\_PORT\_MAP=http:3004:5000
		17. dokku config:set --no-restart templates LOCAL\_ADDRESS=192.168.2.45
		18. dokku config:set --no-restart templates REGION\_ID=92
		19. dokku config:set --no-restart templates TIME\_ZONE=Asia/Yekaterinburg
		20. dokku config:set --no-restart templates TZ=+0500
		21. dokku config:set --no-restart templates HOST\_NAME=Test
		22. dokku config:set --no-restart templates LOCAL\_ADDRESS=10.2.155.38
		23. **Устанавливаем и пересобираем:**
		24. *dokku git:from-image templates docker.pkzdrav.ru/zdraw/templates-backend:1.16.0\_\_bf*
		25. *dokku ps:rebuild templates*
	6. **Torlis App**
		1. **Скачиваем образ проект:**
		2. *docker load -i torlis\_1.82.0\_\_bf2.tar*
		3. **Создаем проект:**
		4. *dokku redis:create torlis*
		5. *dokku apps:create torlis*
		6. *dokku redis:link torlis torlis*
		7. **Добавить проброс папки:**
		8. dokku storage:mount torlis /home/dokku/torlis/static:/app/static
		9. **Настройка и подключение к бд:**
		10. *CREATE USER torlis WITH PASSWORD 'r99VbNnmQ';*
		11. *ALTER USER torlis WITH SUPERUSER;*
		12. *CREATE DATABASE torlis;*
		13. *ALTER DATABASE torlis OWNER TO torlis;*
		14. dokku config:set --no-restart torlis DATABASE\_URL='postgres://torlis:r99VbNnmQ@localhost:5432/torlis*’*
		15. **Установить ENV:**
		16. dokku config:set --no-restart torlis APP\_URL\_PREFIX=torlis
		17. dokku config:set --no-restart torlis CORE\_LOCAL\_PORT=3000
		18. dokku config:set --no-restart torlis CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http://10.10.247.5:3000
		19. dokku config:set --no-restart torlis DOKKU\_APP\_RESTORE=1
		20. dokku config:set --no-restart torlis DOKKU\_APP\_TYPE=dockerfile
		21. dokku config:set --no-restart torlis DOKKU\_CHECKS\_DISABLED=\_all\_
		22. dokku config:set --no-restart torlis DOKKU\_PROXY\_PORT\_MAP=http:3008:5000
		23. dokku config:set --no-restart torlis REGION\_ID=92
		24. dokku config:set --no-restart torlis TIME\_ZONE=Asia/Yekaterinburg
		25. dokku config:set --no-restart torlis TZ=+0500
		26. dokku config:set --no-restart torlis HOST\_NAME=Test
		27. dokku config:set --no-restart torlis LOCAL\_ADDRESS=192.168.2.45
		28. **Устанавливаем и пересобираем:**
		29. *dokku git:from-image torlis docker.pkzdrav.ru/zdraw/torlis:1.82.0\_\_bf*
		30. *dokku ps:rebuild torlis*
	7. **Pdf App**
		1. **Скачиваем образ проект:**
		2. *docker load -i pdf-app\_1.0.0.tar*
		3. **Создаем проект:**
		4. *dokku redis:create pdf-app*
		5. *dokku apps:create pdf-app*
		6. *dokku redis:link pdf-app pdf-app*
		7. **Добавить проброс папки:**
		8. *dokku storage:mount pdf-app /home/dokku/pdf-app/static:/app/static*
		9. *dokku storage:mount pdf-app /komtek/file-service/files:/mnt/emk*
		10. **Настройка и подключение к бд:**
		11. *CREATE USER pdf WITH PASSWORD 'r99VbNnmQ';*
		12. *ALTER USER pdf WITH SUPERUSER;*
		13. *ALTER DATABASE pdf OWNER TO pdf;*
		14. dokku config:set pdf-app DATABASE\_URL='postgres://pdf:r99VbNnmQ@localhostp:5432/pdf'
		15. ***Установить ENV:***
		16. dokku config:set --no-restart pdf-app APP\_URL\_PREFIX=fileservice
		17. dokku config:set --no-restart pdf-app CORE\_LOCAL\_PORT=3000
		18. dokku config:set --no-restart pdf-app CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http://10.10.247.5:3000
		19. dokku config:set --no-restart pdf-app DEBUG=False
		20. dokku config:set --no-restart pdf-app DOKKU\_APP\_RESTORE=1
		21. dokku config:set --no-restart pdf-app DOKKU\_APP\_TYPE=dockerfile
		22. dokku config:set --no-restart pdf-app DOKKU\_CHECKS\_DISABLED=\_all\_
		23. dokku config:set --no-restart pdf-app DOKKU\_DOCKERFILE\_START\_CMD=gunicorn pdf\_app.wsgi --timeout 600
		24. dokku config:set --no-restart pdf-app DOKKU\_PROXY\_PORT\_MAP='http:3007:5000 http:8039:5000'
		25. dokku config:set --no-restart pdf-app REGION\_ID=92
		26. dokku config:set --no-restart pdf-app TIME\_ZONE=Asia/Yekaterinburg
		27. dokku config:set --no-restart pdf-app TZ=+0500
		28. dokku config:set --no-restart pdf-app HOST\_NAME=Test
		29. dokku config:set --no-restart pdf-app LOCAL\_ADDRESS=192.168.2.45
		30. **Устанавливаем и пересобираем:**
		31. dokku git:from-image *pdf-app* docker.pkzdrav.ru/zdraw/ *pdf-app*:1.8.0\_\_bf
		32. *dokku ps:scale pdf-app web=1*
		33. dokku ps:rebuild *pdf-app*
	8. **Reception App**
		1. **Скачиваем образ проект:**
		2. *docker load -i reception\_1.83.0-1.72.10\_\_bf2.tar*
		3. **Создаем проект:**
		4. *dokku redis:create reception*
		5. *dokku apps:create reception*
		6. *dokku redis:link reception reception*
		7. **Добавить проброс папки:**
		8. *dokku storage:mount reception /home/dokku/reception/static:/app/static*
		9. **Настройка и подключение к бд:**
		10. dokku config:set *reception* DATABASE\_URL='postgres://zdraw:zdraw@localhostp:5432/pzdraw
		11. ***Установить ENV:***
		12. dokku config:set --no-restart reception APP\_URL\_PREFIX=*reception*
		13. dokku config:set --no-restart reception CORE\_LOCAL\_PORT=3000
		14. dokku config:set --no-restart reception CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http://10.10.247.5:3000
		15. dokku config:set --no-restart reception DEBUG=False
		16. dokku config:set --no-restart reception DOKKU\_APP\_RESTORE=1
		17. dokku config:set --no-restart reception DOKKU\_APP\_TYPE=dockerfile
		18. dokku config:set --no-restart reception DOKKU\_CHECKS\_DISABLED=\_all\_
		19. dokku config:set --no-restart reception
		20. dokku config:set --no-restart reception REGION\_ID=92
		21. dokku config:set --no-restart reception TIME\_ZONE=Asia/Yekaterinburg
		22. dokku config:set --no-restart reception TZ=+0500
		23. dokku config:set --no-restart reception HOST\_NAME=Test
		24. dokku config:set --no-restart reception LOCAL\_ADDRESS=192.168.2.45
		25. **Устанавливаем и пересобираем:**
		26. dokku git:from-image *reception* docker.pkzdrav.ru/zdraw/reception:1.83.0-1.72.10\_\_bf2
		27. dokku ps:rebuild *reception*
	9. **Apteka App**
		1. **Скачиваем образ проект:**
		2. *docker load -i apteka\_1.262.0-1.16.18\_\_bf2.tar*
		3. **Создаем проект:**
		4. *dokku redis:create apteka*
		5. *dokku apps:create apteka*
		6. *dokku redis:link apteka apteka*
		7. **Добавить проброс папки:**
		8. *dokku storage:mount apteka /home/dokku/apteka/static:/app/static*
		9. **Настройка и подключение к бд:**
		10. dokku config:set *apteka* DATABASE\_URL='postgres://zdraw:zdraw@localhostp:5432/pzdraw
		11. ***Установить ENV:***
		12. dokku config:set --no-restart apteka APP\_URL\_PREFIX=apteka
		13. dokku config:set --no-restart apteka CORE\_LOCAL\_PORT=3000
		14. dokku config:set --no-restart apteka CSRF\_TRUSTED\_ORIGINS=http://localhost,http://localhost:3000,http://10.10.247.5:3000
		15. dokku config:set --no-restart apteka DEBUG=False
		16. dokku config:set --no-restart apteka DOKKU\_APP\_RESTORE=1
		17. dokku config:set --no-restart apteka DOKKU\_APP\_TYPE=dockerfile
		18. dokku config:set --no-restart apteka DOKKU\_CHECKS\_DISABLED=\_all\_
		19. dokku config:set --no-restart apteka REGION\_ID=92
		20. dokku config:set --no-restart apteka TIME\_ZONE=Asia/Yekaterinburg
		21. dokku config:set --no-restart apteka TZ=+0500
		22. dokku config:set --no-restart apteka HOST\_NAME=Test
		23. dokku config:set --no-restart apteka LOCAL\_ADDRESS=192.168.2.45
		24. **Устанавливаем и пересобираем:**
		25. dokku git:from-image *apteka* docker.pkzdrav.ru/zdraw/apteka:1.262.0-1.16.18\_\_bf2
		26. dokku ps:rebuild *reception*
2. **Импортировать NSI справочники в БД**
	* 1. **Выполнить импорт таблиц:**
		2. *DB\_URL=postgres://zdraw:zdraw@localhost:5432/zdraw*
		3. *psql $DB\_URL -f ./frmo\_202406180959.sql*
		4. *psql $DB\_URL -f ./apteka\_drug\_202406180959.sql*
		5. *psql $DB\_URL -f ./lpu\_202406180959.sql*
		6. *psql $DB\_URL -f ./test\_analysis\_202406180959.sql*
		7. *psql $DB\_URL -f ./apteka\_drug\_trade\_name\_202406180959.sql*
		8. *psql $DB\_URL -f ./frmo\_subdivision\_202406180959.sql*
		9. *psql $DB\_URL -f ./f003\_202406180959.sql*
		10. *psql $DB\_URL -f ./mkb10\_202406180959.sql*
		11. *psql $DB\_URL -f ./nmu\_202406180959.sql*
		12. *psql $DB\_URL -f ./apteka\_drug\_maker\_202406180959.sql*
		13. *psql $DB\_URL -f ./сharacter\_toxicology\_screening\_methods\_202406180959.sql*
		14. *psql $DB\_URL -f ./atx\_classification\_202406180959.sql*
		15. *psql $DB\_URL -f ./apteka\_drug\_international\_name\_202406180959.sql*
		16. *psql $DB\_URL -f ./frmr\_fed\_position\_202406180959.sql*
		17. *psql $DB\_URL -f ./apteka\_drug\_form\_202406180959.sql*
		18. *psql $DB\_URL -f ./anatomic\_localization\_202406180959.sql*
		19. *psql $DB\_URL -f ./apteka\_drug\_package\_202406180959.sql*
		20. *psql $DB\_URL -f ./pharm\_therapeutic\_group\_202406180959.sql*
		21. *psql $DB\_URL -f ./smo\_202406180959.sql*
		22. *psql $DB\_URL -f ./units\_202406180959.sql*
		23. *psql $DB\_URL -f ./positions\_202406180959.sql*
		24. *psql $DB\_URL -f ./hst1080\_202406180959.sql*
		25. *psql $DB\_URL -f ./field\_202406180958.sql*
		26. *psql $DB\_URL -f ./types\_med\_doc\_202406180959.sql*
		27. *psql $DB\_URL -f ./frmr\_post\_202406180959.sql*
		28. *psql $DB\_URL -f ./hst1070\_202406180959.sql*
		29. *psql $DB\_URL -f ./device\_type\_202406180959.sql*
		30. *psql $DB\_URL -f ./insurance\_lpu\_202406180959.sql*
		31. *psql $DB\_URL -f ./lab\_biomatters\_202406180959.sql*
		32. *psql $DB\_URL -f ./citizenship\_202406180959.sql*
		33. *psql $DB\_URL -f ./apteka\_provider\_202406180959.sql*
		34. *psql $DB\_URL -f ./refbook\_202406180958.sql*
		35. *psql $DB\_URL -f ./benefits\_category\_202406180959.sql*
		36. *psql $DB\_URL -f ./apteka\_drug\_concentration\_unit\_202406180959.sql*
		37. *psql $DB\_URL -f ./\_version\_\_202406180958.sql*
		38. *psql $DB\_URL -f ./specialities\_202406180959.sql*
		39. *psql $DB\_URL -f ./subjects\_rf\_202406180959.sql*
		40. *psql $DB\_URL -f ./nationality\_202406180959.sql*
		41. *psql $DB\_URL -f ./classifier\_doc\_proving\_identity\_202406180959.sql*
		42. *psql $DB\_URL -f ./field\_set\_202406180958.sql*
		43. *psql $DB\_URL -f ./hiv\_analysis\_test\_202406180959.sql*
		44. *psql $DB\_URL -f ./data\_table\_202406180958.sql*
		45. *psql $DB\_URL -f ./p\_doc\_types\_202406180959.sql*
		46. *psql $DB\_URL -f ./apteka\_drug\_mass\_unit\_202406180959.sql*
		47. *psql $DB\_URL -f ./apteka\_drug\_group\_202406180959.sql*
		48. *psql $DB\_URL -f ./keyboard\_codes\_202406180959.sql*
		49. *psql $DB\_URL -f ./frmr\_speciality\_202406180959.sql*
		50. *psql $DB\_URL -f ./apteka\_drug\_size\_unit\_202406180959.sql*
		51. *psql $DB\_URL -f ./apteka\_drug\_group\_apteka\_warehouse\_type\_202406180959.sql*
		52. *psql $DB\_URL -f ./apteka\_drug\_action\_unit\_202406180959.sql*
		53. *psql $DB\_URL -f ./trigger\_point\_202406180959.sql*
		54. *psql $DB\_URL -f ./apteka\_write\_off\_motive\_202406180959.sql*
		55. *psql $DB\_URL -f ./contingent\_202406180959.sql*
		56. *psql $DB\_URL -f ./employee\_role\_202406180959.sql*
		57. *psql $DB\_URL -f ./mdlp\_withdrawal\_type\_202406180959.sql*
		58. *psql $DB\_URL -f ./mdlp\_schema\_202406180959.sql*
		59. *psql $DB\_URL -f ./toxicology\_screening\_methods\_202406180959.sql*
		60. *psql $DB\_URL -f ./hiv\_analysis\_202406180959.sql*
		61. *psql $DB\_URL -f ./payment\_sources\_202406180959.sql*
		62. *psql $DB\_URL -f ./apteka\_drug\_volume\_unit\_202406180959.sql*
		63. *psql $DB\_URL -f ./social\_statuses\_202406180959.sql*
		64. *psql $DB\_URL -f ./\_source\_\_202406180958.sql*
		65. *psql $DB\_URL -f ./house\_call\_result\_202406180959.sql*
		66. *psql $DB\_URL -f ./house\_call\_symptom\_202406180959.sql*
		67. *psql $DB\_URL -f ./payment\_type\_202406180959.sql*
		68. *psql $DB\_URL -f ./education\_202406180959.sql*
		69. *psql $DB\_URL -f ./slot\_type\_202406180959.sql*
		70. *psql $DB\_URL -f ./apteka\_warehouse\_type\_202406180959.sql*
		71. *psql $DB\_URL -f ./district\_type\_202406180959.sql*
		72. *psql $DB\_URL -f ./mdlp\_upload\_status\_202406180959.sql*
		73. *psql $DB\_URL -f ./marital\_status\_202406180959.sql*
		74. *psql $DB\_URL -f ./mdlp\_contract\_type\_202406180959.sql*
		75. *psql $DB\_URL -f ./referral\_statuses\_202406180959.sql*
		76. *psql $DB\_URL -f ./mdlp\_source\_type\_202406180959.sql*
		77. *psql $DB\_URL -f ./file\_formats\_202406180959.sql*
		78. *psql $DB\_URL -f ./book\_status\_202406180959.sql*
		79. *psql $DB\_URL -f ./booking\_source\_202406180959.sql*
		80. *psql $DB\_URL -f ./policy\_kind\_202406180959.sql*
		81. *psql $DB\_URL -f ./age\_category\_202406180959.sql*
		82. *psql $DB\_URL -f ./mdlp\_workflow\_202406180959.sql*
		83. *psql $DB\_URL -f ./slot\_status\_202406180959.sql*
		84. *psql $DB\_URL -f ./snils\_missing\_reason\_202406180959.sql*
		85. *psql $DB\_URL -f ./house\_call\_type\_202406180959.sql*
		86. *psql $DB\_URL -f ./attachment\_type\_202406180959.sql*
		87. *psql $DB\_URL -f ./apteka\_document\_numbering\_parameter\_202406180959.sql*
		88. *psql $DB\_URL -f ./policy\_type\_202406180959.sql*
		89. *psql $DB\_URL -f ./test\_statuses\_202406180959.sql*
		90. *psql $DB\_URL -f ./resource\_type\_202406180959.sql*
		91. *psql $DB\_URL -f ./waiting\_list\_entry\_status\_202406180959.sql*
		92. *psql $DB\_URL -f ./sex\_202406180959.sql*
		93. *psql $DB\_URL -f ./house\_call\_status\_202406180959.sql*
		94. *psql $DB\_URL -f ./barcode\_lpu\_code\_202406180959.sql*
		95. *psql $DB\_URL -f ./sign\_types\_202406180959.sql*
		96. *psql $DB\_URL -f ./file\_types\_202406180959.sql*
		97. *psql $DB\_URL -f ./waiting\_list\_cancel\_reason\_202406180959.sql*

# **Запуск программы на рабочем месте**

Для запуска программы необходимо открыть браузер, ввести в адресную строку адрес программы и нажать enter. В открывшемся окне ввести логин и пароль. Далее следовать соответствующим инструкциям в руководствах пользователя.

# **Клиент системные требования**

* Браузер
	+ **Google Chrome** версии **105.0.5195.54** от 31 авг 2022 и выше
	+ **Yandex.Browser**версии **22.7.5** от 31 авг 2022 и выше
* Доступ к принтеру, МФУ (при необходимости печатать с рабочего места)
* 4 ГБ Оперативной памяти и выше