

ОГБУ «Центр энергосбережения Белгородской области»

«Система управления энергетическими ресурсами
Белгородской области»

Руководство администратора

г. Белгород, 2026 г.

СОДЕРЖАНИЕ

1. Общие сведения о программном продукте	3
1.1. О системе	3
1.2. Уровень подготовки администратора	3
2. Назначение и условия применения	5
2.1. Назначение системы	5
2.2. Программные и аппаратные требования к системе	5
3. Установка и настройка серверного программного обеспечения	7
3.1. Установка контейнеризатора приложений docker и программного обеспечения docker-compose	7
3.1.1. Установка Docker	7
3.1.2. Установка Docker-compose	8
3.2. Копирование исходного кода веб-приложения	8
3.3. Установка переменных окружения	8
3.4. Запуск и остановка docker-контейнеров	9
3.5. Обслуживание базы данных	10

1. Общие сведения о программном продукте

1.1. О системе

Данное руководство предназначено для специалистов, осуществляющих установку и администрирование серверного программного обеспечения «Системы управления энергетическими ресурсами Белгородской области» (далее – система).

В соответствии с пунктом 1.2 Положения о Системе управления энергетическими ресурсами Белгородской области, утвержденного распоряжением правительства Белгородской области от 27 сентября 2017 № 452-рп «О внедрении Системы управления энергетическими ресурсами Белгородской области», СУЭР является информационной системой Белгородской области, предназначенной для формирования и использования информации об объемах потребления, производства, распределения топливно-энергетических ресурсов и воды, о характеристиках объектов энергосбережения (зданий, строений, сооружений, помещений), об узлах и приборах учета потребляемых топливно-энергетических ресурсов и воды, о мероприятиях по энергосбережению и повышению энергетической эффективности и результатах их реализации, о заключенных договорах электро-, тепло-, водо- и газоснабжения, о потреблении топливно-энергетических ресурсов и воды, об использовании средств бюджета Белгородской области при реализации мероприятий в сфере энергосбережения и повышения энергетической эффективности.

Для разворачивания системы на серверном аппаратном обеспечении необходимо выполнить следующие действия

- настроить ssh-доступ
- настроить рабочий каталог с исходными кодами системы;
- установить docker;
- установить утилиту docker-compose.

На поставляемом с системой диске находится дистрибутив программного обеспечения в виде серверных скриптов и базы данных.

1.2. Уровень подготовки администратора

Для установки и обслуживания серверного программного обеспечения администратор должен уметь:

- обеспечивать функционирование компьютерных систем и компьютерных сетей;
- выполнять организацию и контроль доступа сотрудников к локальной сети и сети Интернет;

- выполнять администрирование и обновление web-приложения и его компонентов;
- выполнять регистрацию, назначение паролей и создание учетных записей пользователей;
- выполнять интеграцию программного обеспечения на файл-серверах, серверах баз данных и на рабочих местах;
- выполнять установку антивирусных программ и контроль их своевременного обновления;
- обеспечивать безопасность компьютерной системы и защиту доступа к системным файлам;
- выполнять выделение места на диске для сохранности данных и резервное копирование информации;
- уметь работать с системами управления базами данных (PostgreSQL);
- выполнять обучение пользователей работе с компьютерными программами;
- выполнять обновление работоспособности компьютерных систем после сбоев;
- проводить мониторинг системы, выполнять просмотр и анализ логов;
- осуществлять ведение технической документации.

2. Назначение и условия применения

2.1. Назначение системы

СУЭР является информационной системой Белгородской области, предназначенной для формирования и использования информации об объемах потребления, производства, распределения топливно-энергетических ресурсов и воды, о характеристиках объектов энергосбережения (зданий, строений, сооружений, помещений), об узлах и приборах учета потребляемых топливно-энергетических ресурсов и воды, о мероприятиях по энергосбережению и повышению энергетической эффективности и результатах их реализации, о заключенных договорах электро-, тепло-, водо- и газоснабжения, о потреблении топливно-энергетических ресурсов и воды, об использовании средств бюджета Белгородской области при реализации мероприятий в сфере энергосбережения и повышения энергетической эффективности.

Система обеспечивает автоматизацию выполнения требований нормативных правовых актов в сфере энергосбережения и повышения энергоэффективности, в части касающейся государственных и муниципальных учреждений, в целях обеспечения выполнения требований к снижению государственными и муниципальными учреждениями в сопоставимых условиях суммарного объема потребляемых ими ресурсов в соответствии с положениями ПП РФ № 1289.

Полномочия Пользователей определены в разделе 2 Положения о Системе управления энергетическими ресурсами Белгородской области, утвержденного распоряжением Правительства Белгородской области № 452-рп.

2.2. Программные и аппаратные требования к системе

Для корректной работы системы следует использовать браузеры, обеспечивающие полную совместимость со стандартами: HTML 4.01 и выше, CSS 3 и JavaScript.

В таблицах 1 и 2 приведены основные требования к программному и аппаратному обеспечению.

Таблица 1 – Рекомендуемое программное обеспечение

Вид ПО	Программный продукт	Версия
Операционная система	Microsoft Windows	10 и выше
	Linux	Debian 12 / Ubuntu 22.04 и выше / Astra 1.7 SE
	MacOS	10.6 и выше

Вид ПО	Программный продукт	Версия
Web-браузер	Microsoft Edge	40. и выше
	Google Chrome	70. и выше
	Mozilla Firefox	63. и выше
	Opera	54. и выше

Таблица 2 – Минимальные рекомендуемые требования к аппаратному обеспечению

Компонент	Характеристики
Аппаратные ресурсы сервера	
Процессор	Архитектура 64-битный ПК (amd64), не менее 8 ядер
Оперативная память (ОЗУ)	16 Гбайт
ПЗУ	500 Гбайт
Сетевое обеспечение	Сетевое подключение с выходом в Интернет
Аппаратные ресурсы клиента	
Процессор	Intel Core i3 / AMD Ryzen 3 2.2 ГГц и выше
Оперативная память (ОЗУ)	4 Гбайт
ПЗУ	Не менее 80 Гбайт
Сетевое обеспечение	Сетевое подключение с выходом в Интернет

3. Установка и настройка серверного программного обеспечения

3.1. Установка контейнеризатора приложений docker и программного обеспечения docker-compose.

3.1.1. Установка Docker

Обновить существующий список пакетов:

```
sudo apt update
```

Установка необходимых пакетов, которые позволяют apt использовать пакеты через HTTPS:

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Добавить ключ GPG для официального репозитория Docker в систему:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Добавить репозиторий Docker в источники APT:

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
```

Обновление базы данных пакетов и добавление в нее пакетов Docker из добавленного репозитория:

```
sudo apt update
```

Установка Docker:

```
sudo apt install docker-ce
```

Проверить, что docker запущен:

```
sudo systemctl status docker
```

Вывод должен быть примерно следующий:

```
docker.service - Docker Application Container Engine
```

```
Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
```

```
Active: active (running) since Пн 2024-05-27 15:50:48 MSK; 6 days ago
```

```
Docs: https://docs.docker.com
```

Main PID: 1001 (dockerd)

Tasks: 54

Memory: 2.4G

CPU: 13min 49.975s

3.1.2. Установка Docker-compose

Загрузить стабильную версию:

```
sudo curl -L  
"https://github.com/docker/compose/releases/download/1.29.2/docker-compose-  
$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Задать права для исполняемого файла:

```
sudo chmod +x /usr/local/bin/docker-compose
```

Создать символическую ссылку для использования из под текущего пользователя:

```
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Проверить результат установки, командой:

```
docker-compose --version
```

3.2. Копирование исходного кода веб-приложения

Стандартный каталог для размещения веб-приложений находится в `/var/www/`. Создайте подкаталог по имени `suer` в этом каталоге и скопируйте в него исходный код проекта.

3.3. Установка переменных окружения

Переменные окружения используются для настройки различных системных параметров в docker контейнерах и должны быть установлены перед сборкой контейнеров. Переменные окружения, необходимые для корректной работы системы указаны в файлах `/var/www/suer/envs`

`common.env` – содержит настройки Django-проекта, а также настройки электронной почты и настройки для подключения к базе данных.

```
ALLOWED_HOSTS=*
```

```
DB_NAME=<db_name>
```

```
DB_CACHE_NAME=<db_name_cache>
```

```
DB_USER=<user>
```

```
DB_PASSWORD=<pass>
DB_HOST=<hostname>
DB_PORT=<port>
EMAIL_HOST=<host>
EMAIL_PORT=<port>
EMAIL_HOST_USER=<email>
EMAIL_HOST_PASSWORD=<pass>
EMAIL_TECH_SUPPORT=<email>
SUER_ACCESS_TOKEN=<token>
SECRET_KEY=not-for-production-key
```

celery.env – содержит настройки брокера сообщений для celery.
CELERY_BROKER=amqp://<login>:<password>@rabbitmq:5672/

db.env – содержит настройки для контейнера с СУБД.
POSTGRES_DATABASE=suer
POSTGRES_USER=postgres
POSTGRES_PASSWORD=postgres

backup.env – содержит настройки для выполнения резервного копирования базы данных.

```
POSTGRES_HOST=<host>
POSTGRES_DB=<db_name>
POSTGRES_USER=<db_user>
POSTGRES_PASSWORD=<db_pass>
POSTGRES_EXTRA_OPTS=-Z6 --schema=public --blobs
SCHEDULE=@daily
BACKUP_KEEP_DAYS=7
BACKUP_KEEP_WEEKS=4
BACKUP_KEEP_MONTHS=6
HEALTHCHECK_PORT=8080
```

3.4. Запуск и остановка docker-контейнеров

Описание конфигурации для запуска докер-контейнеров находится в файле `/var/www/suer/docker-compose.yml`. Для запуска системы, разворачиваемой в докер-контейнерах необходимо выполнить команду:

```
sudo docker-compose -f docker-compose.yml up -d --build
```

Для остановки контейнеров:

```
sudo docker-compose -f docker-compose.yml stop
```

Для остановки и удаления контейнеров, в случае необходимости их пересборки:

```
sudo docker-compose -f docker-compose.yml down
```

3.5. Обслуживание базы данных

Резервное копирование базы данных выполняется по расписанию, которое устанавливается в переменных окружения системы (см. п. 3.3). Для выполнения принудительного резервного копирования, необходимо войти в терминал докер-контейнера postgres-backup:

```
sudo docker exec -it suer_backup bash
```

и выполнить команду: *. /backup*

Для восстановления из резервной копии:

```
docker exec -it postgres psql -U postgres -d postgres -c "DROP DATABASE  
<db_name>; CREATE DATABASE <db_name>»
```

```
gunzip < backup.sql.gz | docker exec -i postgres psql -U <user> -d <db_name>
```