

**Программно-аппаратный комплекс табло переменной
информации "Идема"**

Руководство администратора по развертыванию

Версия 1.0.0

На 25 листах

Содержание

1 Информация о системе	3
1.1 Наименование и версия системы	3
1.2 Изготовитель	3
2 Общие сведения	4
2.1 Назначение системы	4
2.2 Требования к пользователю системы	4
2.3 Системные требования	4
3 Установка серверной части на Ubuntu 16.04 – 22.04	6
3.1 Установка PHP v7.4	6
3.2 Установка Composer v2.4.4	6
3.3 Установка Nginx v1.18.0	8
3.4 Установка Node.js v16.15.1	11
3.5 Установка PostgreSQL v14.5	12
3.6 Установка программно-аппаратного комплекса табло переменной информации "Идема"	14
4 Установка серверной части через Docker compose	18
4.1 Установка Docker и docker compose	18
4.2 Установка программно-аппаратного комплекса табло переменной информации "Идема"	18
5 Установка модуля локальной части системы ИДЕМА - ИДЕМА "Локальный узел"	22
6 Установка модуля локальной части системы ИДЕМА - ИДЕМА "Display Client"	24

1 Информация о системе

1.1 Наименование и версия системы

Программно-аппаратный комплекс табло переменной информации "Идема", версия 1.0.0.

1.2 Изготовитель

Полное наименование: Общество с ограниченной ответственностью «ИДЕМА»

Краткое наименование: ООО «ИДЕМА»

ИНН: 1300002184

КПП: 130001001

Юридический адрес: 430030, Республика Мордовия, г. Саранск, ул. Васенко д. 38Б, офис 6

www: <https://идема.рф/о-компании/>

1.3 Контакты службы разработки и технической поддержки

Телефон: +79297476364

E-mail: idemarm@yandex.ru

2 Общие сведения

2.1 Назначение системы

Программно-аппаратный комплекс табло переменной информации "Идема" предназначен для настройки сценариев работы, их комбинирования в рамках группы оборудования, обеспечивающее отображения текстовой информации на табло в зависимости от сценария работы.

2.2 Требования к пользователю системы

В системе предусмотрена ролевая модель:

- Супер администратор;
- Администратор;
- Оператор.

Требования к квалификации пользователя с ролью Администратор:

- Навыки администрирования и сопровождения ОС Linux;
- Изучение инструкций пользователя и администратора.

Требования к квалификации пользователя с ролью Оператор:

- Уверенное владение ПК на уровне пользователя;
- Изучение инструкции пользователя по работе с Системой.

2.3 Системные требования

Характеристики сервера для установки серверной версии:

- ОС: Ubuntu 20.04 LTS (ENG);
- Конфигурация: 1x2.2ГГц, 1Гб RAM, 1IP;
- Диск: 20Гб SSD RAID (Операционная система).

Характеристики рабочего места для установки локальной версии:

- Процессор: 2 ядра;
- Оперативная память: 1 Гб;
- Hdd: 2 Гб.

Требования к информационной и программной совместимости:

Таблица 1 - Требование к взаимодействию локальной и серверной версии

Отправитель	Интерфейс отправителя	Сообщение	Получатель
Серверная версия	RestAPI	Передача/получение конфигурации оборудования	Локальная версия
Серверная версия	RestAPI	Передача/получение конфигурации сценариев	Локальная версия

3 Установка серверной части на Ubuntu 16.04 – 22.04

3.1 Установка PHP v7.4

Необходимо обновить пакеты и добавить ppa репозиторий от Ondrej. Выполняются следующие команды:

1. «sudo apt update && sudo apt upgrade»;
2. «sudo apt install software-properties-common»;
3. «sudo add-apt-repository ppa:ondrej/php -y»;

После выполнения команд нужно установить php7.4 при помощи менеджера пакетов apt.

Необходимо выполнить следующую команду:

- «sudo apt-get install php7.4-cli php7.4-common php7.4-curl php7.4-fpm php7.4-gd php7.4-json php7.4-mbstring php7.4-opcache php7.4-pgsql php7.4-soap php7.4-xml php7.4-zip»;

После выполнения команды PHP7.4 установлен, но нужно проверить его текущую версию по умолчанию (см. рис. 3.1)

Необходимо выполнить следующую команду:

- «php -v»;



```
andrey@b450-x: ~
andrey@b450-x:~$ php -v
PHP 7.4.33 (cli) (built: Nov 8 2022 11:33:53) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.33, Copyright (c), by Zend Technologies
andrey@b450-x:~$
```

Рис. 3.1 – Проверка версии php7.4

3.2 Установка Composer v2.4.4

Для установки composer необходимо добавить его в список репозиторий ubuntu.

Необходимо выполнить следующие команды:

1. «sudo apt install php-cli unzip»;
2. «cd ~»;
3. «curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php»;
4. «HASH=`curl -sS [https://composer.github.io/installer.sig`»;](https://composer.github.io/installer.sig)
5. «echo \$HASH»;
6. «php -r "if (hash_file('SHA384', '/tmp/composer-setup.php') === '\$HASH') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"»;

После выполнения списка команд, мы должны увидеть сообщение в консоли «Installer verified» (см. рис. 3.2).



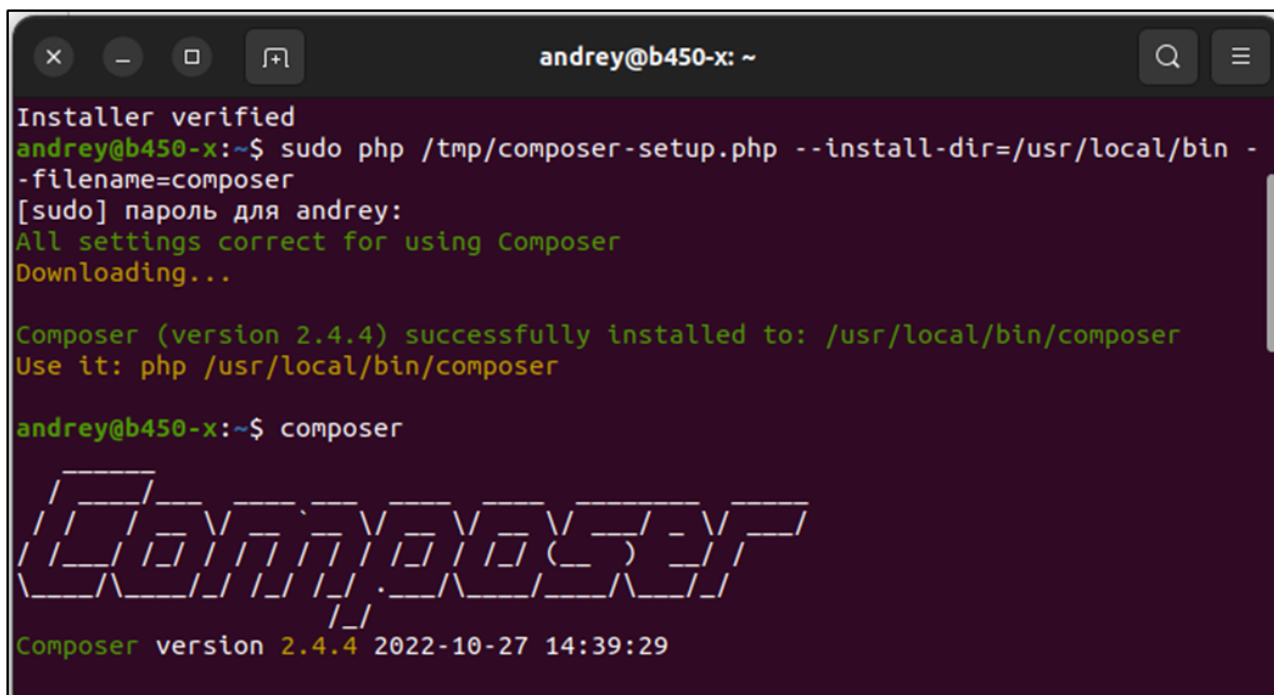
```
andrey@b450-x: ~  
andrey@b450-x:~$ curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php  
andrey@b450-x:~$ HASH=`curl -sS https://composer.github.io/installer.sig`  
andrey@b450-x:~$ echo $HASH  
55ce33d7678c5a611085589f1f3ddf8b3c52d662cd01d4ba75c0ee0459970c2200a51f492d557530  
c71c15d8dba01eae  
andrey@b450-x:~$ php -r "if (hash_file('SHA384', '/tmp/composer-setup.php') ===  
'$HASH') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink(  
'composer-setup.php'); } echo PHP_EOL;"  
Installer verified  
andrey@b450-x:~$
```

Рис 3.2 – Выполнение списка команд для установки composer

Composer необходимо сделать так, чтобы он был доступен в любой директории. Его нужно сделать «глобальным». Затем проверим версию установленного пакета (см. рис. 3.3).

Для этого нужно выполнить следующие команды:

1. «sudo php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer»;
2. «composer»;



```
andrey@b450-x: ~
Installer verified
andrey@b450-x:~$ sudo php /tmp/composer-setup.php --install-dir=/usr/local/bin -
filename=composer
[sudo] пароль для andrey:
All settings correct for using Composer
Downloading...

Composer (version 2.4.4) successfully installed to: /usr/local/bin/composer
Use it: php /usr/local/bin/composer

andrey@b450-x:~$ composer

COMPOSER

Composer version 2.4.4 2022-10-27 14:39:29
```

Рис. 3.3 – Установка Composer

3.3 Установка Nginx v1.18.0

Для установки Nginx необходимо выполнить следующую команду:

- «sudo apt-get install nginx»;

После установки nginx необходимо создать виртуальный хост со следующим содержимым (необходимо полностью выделить и скопировать содержимое):

```
server {
    listen 80;
    server_name aggregator.local;
    root /var/www/aggregator/public;
    index index.php;

    client_max_body_size 256m;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
```

```

}

location ~* ^.+.(js|css|png|jpg|jpeg|gif|ico)$ {
    access_log      off;
    expires         max;
}

location ~ /\.php$ {
    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    fastcgi_index  index.php;

    include fastcgi_params;

    fastcgi_param          SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    try_files $uri =404;
    fastcgi_read_timeout 600;
}
location = /favicon.ico {
    log_not_found off;
    access_log off;
}

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}

## Disable viewing .htaccess & .htpassword
location ~ /\.ht {
    deny all;
}
}

```

Далее открываем утилиту nano, вставляем скопированное содержимое и жажимаем сочетание клавиш «ctrl + O» для сохранения изменений. После этого необходимо выйти из утилиты nano «ctrl + X».

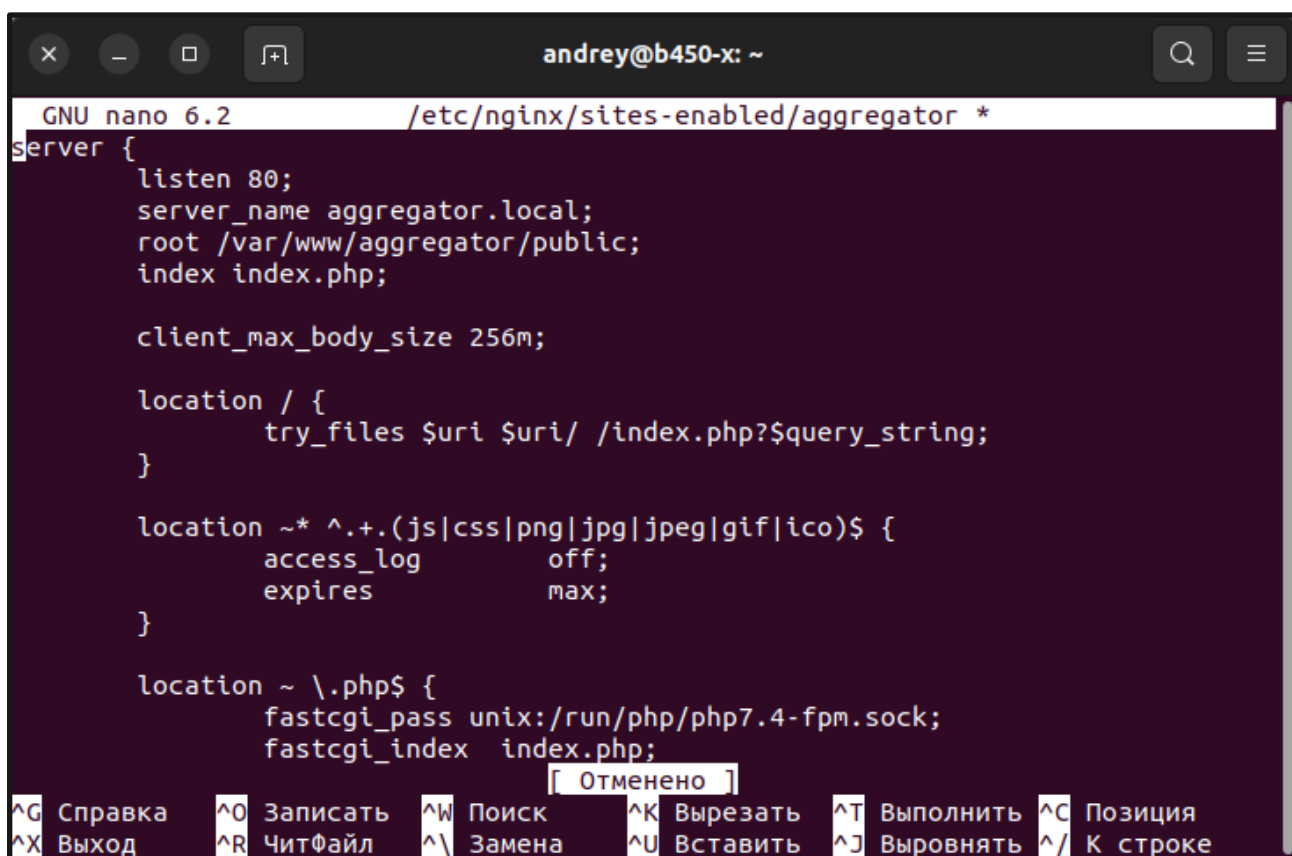
Выполняем следующие команды:

1. «sudo nano /etc/nginx/sites-available/aggregator» (после выполнения этой команды откроется утилита nano, куда нужно вставить текст);
2. «sudo ln -s /etc/nginx/sites-available/skuts /etc/nginx/sites-enabled/aggregator»

Проверяем, что виртуальный хост доступен в папке «sites-enabled» (см. рис. 3.4).

Выполняем следующую команду:

- «sudo nano /etc/nginx/sites-enabled/aggregator»;



```
GNU nano 6.2 /etc/nginx/sites-enabled/aggregator *
server {
    listen 80;
    server_name aggregator.local;
    root /var/www/aggregator/public;
    index index.php;

    client_max_body_size 256m;

    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }

    location ~* ^.+.(js|css|png|jpg|jpeg|gif|ico)$ {
        access_log off;
        expires max;
    }

    location ~ \.php$ {
        fastcgi_pass unix:/run/php/php7.4-fpm.sock;
        fastcgi_index index.php;
    }
}
[ Отменено ]
^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/_ К строке
```

Рис. 3.4 – Создание виртуального хоста в папке «sites-enabled»

Чтобы изменения применились, необходимо перезапустить веб-сервер nginx следующей командой:

- «sudo service nginx restart»

Добавляем 1 строчку в файл hosts при помощи nano для того, чтобы проект был доступен в браузере по адресу <http://aggregator.local>. Для этого выполняем следующую команду:

- «sudo nano /etc/hosts» (добавляем в конец файла строку «127.0.0.1 skuts.local». Сохраняем «Ctrl + O» и выходим «Ctrl + X»).

3.4 Установка Node js v16.15.1

Установку node js будем производить через пакетный менеджер nvm (см. рис. 3.5).

Необходимо выполнить следующие команды:

1. «cd ~»
2. «curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash»;
3. «source ~/.bashrc»;
4. «nvm install v16.15.1»;
5. «node -v»

```
andrey@b450-x: ~
andrey@b450-x:~$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 14926  100 14926    0     0  56207      0  --:--:-- --:--:-- --:--:-- 56324
=> nvm is already installed in /home/andrey/.nvm, trying to update using git
=> => Compressing and cleaning up git repository

=> nvm source string already in /home/andrey/.bashrc
=> bash_completion source string already in /home/andrey/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use
it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads
nvm bash_completion
andrey@b450-x:~$ source ~/.bashrc
andrey@b450-x:~$ nvm install v16.15.1
v16.15.1 is already installed.
Now using node v16.15.1 (npm v8.11.0)
andrey@b450-x:~$ node -v
v16.15.1
```

Рис. 3.5 – Установка node js v16.15.1

3.5 Установка PostgreSQL v14.5

Для установки БД PostgreSQL необходимо выполнить следующую команду:

- «sudo apt-get install postgresql»;

Для настройки аутентификации **ident** необходимо открыть конфигурационный файл и раскомментировать строку (то есть убрать символ «#») при помощи утилиты nano.

Выполняем следующую команду:

- «sudo nano /etc/postgresql/14/main/postgresql.conf»; (после запуска команды, нажимаем Ctrl + W и вводим в поиске «#listen_localhost» и в найденной строке убираем символ «#». Затем сохраняем изменения Ctrl + O и выходим Ctrl+ X);

После включения аутентификации, подключаемся к БД и выполняем следующие команды:

1. «sudo -u postgres psql template1»;
2. «ALTER USER postgres with encrypted password 'your_password';» (где your_password – придумываем новый пароль к пользователю «postgres»). Можно воспользоваться генератором паролей <http://www.onlinepasswordgenerator.ru>);
3. «CREATE DATABASE aggregator;»;
4. «\q» - вводим эту команду и нажимаем Enter, чтобы выйти из консоли PostgreSQL;

После создания БД и настройки пароля для пользователя postgres, необходимо включить использование MD5 аутентификации (см. рис. 3.6). Необходимо выполнить несколько команд:

«sudo nano /etc/postgresql/14/main/pg_hba.conf» (после запуска команды, нажимаем Ctrl + W и вводим в поиске (пробелы не удаляем) «local all postgres peer» и в найденной строке в конце заменяем слово с «peer» на «md5». Затем сохраняем изменения Ctrl + O и выходим Ctrl+ X);

```

GNU nano 6.2 /etc/postgresql/14/main/pg_hba.conf *
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres md5
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# 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

```

Legend for nano editor:

- ^G Справка
- ^X Выход
- ^O Записать
- ^R ЧитФайл
- ^W Поиск
- ^_ Замена
- ^K Вырезать
- ^U Вставить
- ^T Выполнить
- ^J Выровнять
- ^C Позиция
- ^/ К строке

Рис. 3.6 – Использование MD5 аутентификации

Для того, чтобы изменения применились, необходимо перезагрузить PostgreSQL следующей командой:

- «sudo service postgresql restart»;

3.6 Установка программно-аппаратного комплекса табло переменной информации "Идема"

Скачиваем установочный пакет.

Распаковываем проект (в zip архиве) при помощи следующих команд:

1. «sudo apt-get install unzip»;
2. «sudo mkdir /var/www/aggregator»;
3. «sudo chown \$USER:\$USER /var/www/aggregator»;
4. «unzip ~/Downloads/aggregator.zip -d /var/www/aggregator» (где ~/Downloads/aggregator.zip – это путь к архиву);
5. «export ARCHIVE_NAME=название_архива_без_расширения»;
6. «mv /var/www/aggregator/\$ARCHIVE_NAME/.[!.]* /var/www/aggregator && mv /var/www/aggregator/\$ARCHIVE_NAME/* /var/www/aggregator && cd ~»;
7. «rm -rf /var/www/aggregator/\$ARCHIVE_NAME/»;
8. «cd /var/www/aggregator && cp .env.example .env»;

После распаковки архива и частичной настройки проекта, необходимо отредактировать .env файл, ориентируясь на рис. 3.7-3.9:

- «nano .env» (после редактирования сохраняем «Ctrl + O», закрываем «Ctrl + X»);

```
GNU nano 6.2 .env *
APP_NAME=Aggregator
APP_ENV=production
APP_KEY=base64:RJdCmr+5UVZdDhLu7F7Qq0sr80neoYzBKBEc+9AbyLI=
APP_DEBUG=false
APP_URL=http://aggregator.local

LOG_CHANNEL=stack
LOG_LEVEL=debug

DB_CONNECTION=pgsql
DB_HOST=localhost
DB_PORT=5432
DB_DATABASE=aggregator
DB_USERNAME=postgres
DB_PASSWORD=postgres

BROADCAST_DRIVER=log
CACHE_DRIVER=file

^G Справка   ^O Записать  ^W Поиск     ^K Вырезать  ^T Выполнить  ^C Позиция
^X Выход     ^R ЧитФайл  ^\ Замена    ^U Вставить  ^J Выровнять  ^/ К строке
```

Рис. 3.7 – Настройка .env файла (часть 1)

```
BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
SESSION_DOMAIN=

MEMCACHED_HOST=127.0.0.1
```

Рис. 3.8 – Настройка .env файла (часть 2)

```
GNU nano 6.2 .env *
AWS_BUCKET=

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"

SANCTUM_STATEFUL_DOMAINS=aggregator.local
GATEWAY_TOKEN=secret
GATEWAY_TIMEOUT=5
```

← Должно быть точно также

← Запросить у клиентской части проекта TOKEN.

^G Справка ^O Записать ^W Поиск ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить ^J Выровнять ^/_ К строке

Рис. 3.9 – Настройка .env файла (часть 3)

Устанавливаем проект и настраиваем права доступа к папкам следующими командами:

1. «composer i && npm i && npm run prod && php artisan refresh»;
2. «sudo usermod -a -G www-data \$USER»
3. «sudo chown -R www-data:www-data storage»
4. «sudo chown -R www-data:www-data bootstrap/cache»

После установки проекта, перезагружаем сервер или компьютер при помощи команды:

- «sudo reboot»

Далее открываем проект в браузере <http://idema.local>

Если проект устанавливается на сервер, то необходимо во всех упоминаниях домена aggregator.local изменить на Ваш домен или ip-адрес. Нужно изменить следующие файлы:

- /etc/nginx/sites-enabled/aggregator;

- `/var/www/aggregator/.env`;

- `/etc/hosts` – (если на локальной машине. Если на сервере, то не нужно вписывать).

4 Установка серверной части через Docker compose

4.1 Установка Docker и docker compose

Для установки docker необходимо выполнить несколько команд:

1. «sudo apt-get remove docker docker-engine docker.io containerd runc»;
2. «sudo apt-get update»
3. «sudo apt-get install ca-certificates curl gnupg lsb-release»;
4. «sudo apt-get install docker.io»

Для установки docker compose необходимо выполнить несколько команд:

- «mkdir -p ~/.docker/cli-plugins/»;
- «curl -SL https://github.com/docker/compose/releases/download/v2.3.3/docker-compose-linux-x86_64 -o ~/.docker/cli-plugins/docker-compose»;
- «chmod +x ~/.docker/cli-plugins/docker-compose»;
- «docker compose version»

Результат установки docker и docker compose представлены на рис. 3.10:

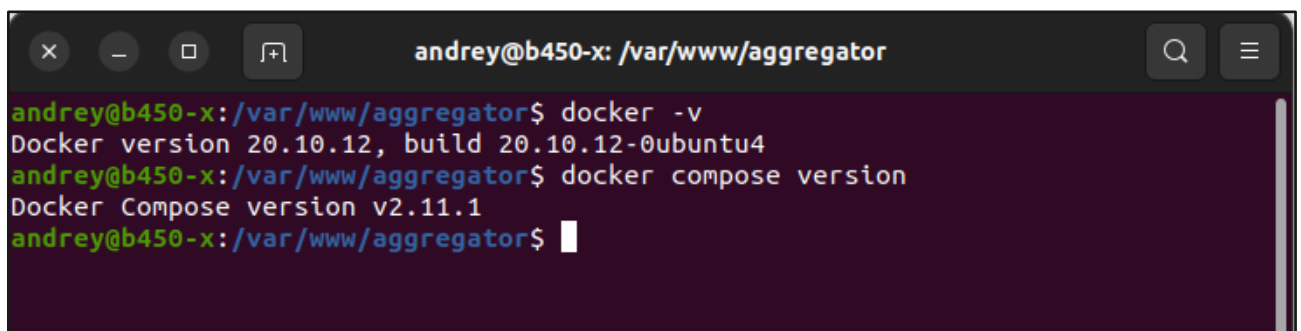
A screenshot of a terminal window with a dark background. The window title is "andrey@b450-x: /var/www/aggregator". The terminal shows three commands and their outputs: 1. "andrey@b450-x:/var/www/aggregator\$ docker -v" followed by "Docker version 20.10.12, build 20.10.12-0ubuntu4". 2. "andrey@b450-x:/var/www/aggregator\$ docker compose version" followed by "Docker Compose version v2.11.1". 3. "andrey@b450-x:/var/www/aggregator\$" followed by a cursor. The window has standard Linux window controls (close, maximize, search, menu) in the top right corner.

Рис. 3.10 – Результат установки docker и docker compose

4.2 Установка программно-аппаратного комплекса табло переменной информации "Идема"

Скачиваем установочный пакет.

Распаковываем проект (в zip архиве) при помощи следующих команд:

1. «sudo apt-get install unzip»;
2. «sudo mkdir /var/www/aggregator»;
3. «sudo chown \$USER:\$USER /var/www/aggregator»;
4. «unzip ~/Downloads/aggregator.zip -d /var/www/aggregator» (где ~/Downloads/aggregator.zip – это путь к архиву);
5. «export ARCHIVE_NAME=название_архива_без_расширения»;
6. «mv /var/www/aggregator/\$ARCHIVE_NAME/.[!]* /var/www/aggregator && mv /var/www/aggregator/\$ARCHIVE_NAME/* /var/www/aggregator && cd ~»;
7. «rm -rf /var/www/aggregator/\$ARCHIVE_NAME/»;
8. «cd /var/www/aggregator»;

– Устанавливаем проект одной командой через docker compose, представленный на рис. 3.11:

- «cp .env.docker .env && docker compose up -d --build && docker exec -it aggregator_app /bin/bash --login -c "chown -R www-data:www-data storage bootstrap/cache && composer i && php artisan refresh && npm i && npm run production"»



```
assets/js/657.60205bc548d3b272.chunk.js 39.1 KiB
assets/js/657.60205bc548d3b272.chunk.js.map 126 KiB
assets/js/787.cb4a428bae6bdda4.chunk.js 10.2 KiB
assets/js/787.cb4a428bae6bdda4.chunk.js.map 30.2 KiB
assets/js/944.8f269543a47863b1.chunk.js 6.54 KiB
assets/js/944.8f269543a47863b1.chunk.js.map 22.8 KiB
assets/js/99.739b8a5d01668c3a.chunk.js 10.7 KiB
assets/js/99.739b8a5d01668c3a.chunk.js.map 36.4 KiB
fonts/vendor/element-ui/lib/theme-chalk/element-icons.ttf?4520188144a17fb24a6af28a70dae0ce 54.6 KiB
fonts/vendor/element-ui/lib/theme-chalk/element-icons.woff?313f7dacf2076822059d2dca26dedfc6 27.5 KiB

webpack compiled successfully
andrey@b450-x: /var/www/aggregator$
```

Рис. 3.11 – Успешная сборка проекта

Необходимо проверить .env файл после установки, представленный на рис.

3.12. Открываем через nano:

- «nano /var/www/aggregator/.env»

```
GATEWAY_TOKEN=secret
GATEWAY_TIMEOUT=5
```

← Необходимо запросить TOKEN у клиентской части.

Рис. 3.12 – Редактирование .env файла

Далее открываем проект в браузере (рис. 3.13) <http://localhost:9097>

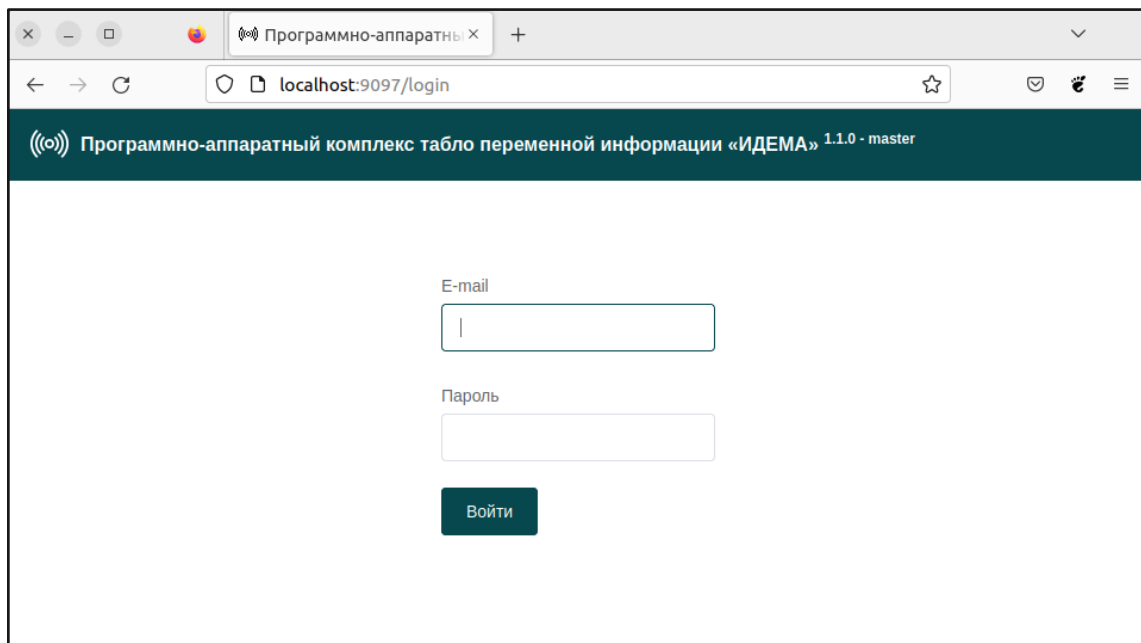


Рис. 3.13 – Просмотр проекта в браузере

5 Установка модуля локальной части системы ИДЕМА - ИДЕМА "Локальный узел"

Шаги по установке:

1. Подготовка окружения:

Настройте соединение с сетью Интернет на оборудовании, на котором планируется установка. Это необходимо для автоматической установки всех зависимостей.

Выполните любым удобным способом передачу установочного пакета на оборудование.

Для упрощения передачи и обновления deb-пакетов создан apt-репозиторий. Для его добавления воспользуйтесь командой:

- `wget -q -O - http://gateway:licIO@185.39.119.244:8505/repository/raw-file-storage/apt/public.gpg.key | sudo apt-key add -`

```
echo "deb [arch=all] http://gateway:licIO@185.39.119.244:8505/repository/idema-local-release/ release main" | sudo tee /etc/apt/sources.list.d/idema.list
```

```
sudo apt-get update
```

2. Установка/обновление модуля:

Установка или обновление модуля выполняется следующей командой:

- `sudo apt install ./idema-local-gateway-x.y.z.deb`

Если выполнено подключение к apt-репозиторию:

- `sudo apt install idema-local-gateway`

При необходимости соглашайтесь на установку требуемых зависимостей.

Если установка завершилось успешно, то будет создана рабочая директория **`/opt/idema-local-gateway/`**.

3. Регистрация в качестве сервиса:

Для автоматического запуска модуля при старте системы, его необходимо зарегистрировать в качестве сервиса. Для этого необходимо выполнить скрипт:

- `sudo bash /opt/idema-local-gateway/systemd/installDaemon.sh`

4. Запуск системы:

Если система установлена в качестве сервиса, то выполните:

- `sudo bash /opt/idema-local-gateway/systemd/startDaemon.sh`

или

- `sudo systemctl start idema-local-gateway`

5. Проверка состояния

Для проверки состояния используется следующая команда:

- `systemctl status idema-local-gateway`

6. Завершение работы:

Для завершения работы выполните команду:

- `sudo systemctl stop idema-local-gateway`

Удаление:

Удаление выполняется командой:

- `sudo apt remove idema-local-gateway`

6 Установка модуля локальной части системы ИДЕМА - ИДЕМА "Display Client"

Шаги по установке:

1. Подготовка окружения:

Настройте соединение с сетью Интернет на оборудовании, на котором планируется установка. Это необходимо для автоматической установки всех зависимостей.

Выполните любым удобным способом передачу установочного пакета на оборудование.

2. Установка/обновление модуля:

Установка или обновление модуля выполняется следующей командой:

- `sudo apt install ./idema-local-displayclient-x.y.z.deb`

При необходимости соглашайтесь на установку требуемых зависимостей.

Если установка завершилась успешно, то будет создана рабочая директория **`/opt/idema-local-displayclient/`**.

3. Регистрация в качестве сервиса:

Для автоматического запуска модуля при старте системы, его необходимо зарегистрировать в качестве сервиса. Для этого необходимо выполнить скрипт:

- `bash /opt/idema-local-displayclient/systemd/installDaemon.sh`

Примечание: запуск команды осуществляется без прав `sudo`.

4. Запуск системы:

Если система установлена в качестве сервиса, то выполните:

- `bash /opt/idema-local-displayclient/systemd/startDaemon.sh`

или

- `systemctl --user start idema-local-displayclient`

5. Проверка состояния

Для проверки состояния используется следующая команда:

- `systemctl --user status idema-local-displayclient`

6. Завершение работы:

Для завершения работы выполните команду:

- `systemctl --user stop idema-local-displayclient`

Удаление:

Удаление выполняется командой:

- `sudo apt remove idema-local-displayclient`