Галактика EAM на платформе Sezal

Инструкция по установке приложения

Корпорация Галактика

2024

Программные и аппаратные требования

Сервер базы данных

Программные требования:

- Postgres Pro Standard v16, Postgres Pro Enterprise v16, PostgreSQL v16.

Аппаратные требования:

Требования	**Минимальные**	**Рекомендуемые**
Количество ядер	4	Определяется индивидуально для конкретного проекта
Оперативная память (ГБ)	16	Определяется индивидуально для конкретного проекта
Количество дискового пространства (ГБ), не учитывая ОС	100	Определяется индивидуально для конкретного проекта
Производительность СХД в IOPS	Требований нет	15000+
Уровень RAID	Требований нет	RAID 10
Производительность сетевых адаптеров на сервер (100 Мбит / 1 Гбит)	Требований нет	>=1 Гбит/с

Сервер приложений

Программные требования:

- Операционная система Astra Linux Common Edition 2.12 / Astra Linux Special Edition 1.6 / Alt Linux Workstation 10.2, Ubuntu 18.04, Docker, Docker compose.
- dotnet 8.0.

Аппаратные требования:

Требования	**Минимальные**	**Рекомендуемые**
Количество ядер	4	Определяется индивидуально для конкретного проекта

Требования	Минимальные	Рекомендуемые
Оперативная память (ГБ)	16	Определяется индивидуально для конкретного проекта
Количество дискового пространства (ГБ), не учитывая ОС	100	Определяется индивидуально для конкретного проекта
Производительность СХД в IOPS	Требований нет	15000+
Уровень RAID	Требований нет	RAID 10
Производительность сетевых адаптеров на сервер (100 Мбит / 1 Гбит)	Требований нет	>=1 Гбит/с

Клиенты

Программные требования:

- Операционная система Astra Linux Common Edition 2.12 / Astra Linux Special Edition 1.6 / Ubuntu 18.04, Alt Linux Workstation 10.2 и выше.
- Яндекс браузер версии 24 и выше.

Аппаратные требования:

Требования	Минимальные	Рекомендуемые
Количество ядер	1	4
Оперативная память (ГБ)	4	8
Количество дискового пространства (ГБ), не учитывая ОС	1	1
Производительность СХД в IOPS	Требований нет	Требований нет
Уровень RAID	Требований нет	Требований нет
Производительность сетевых адаптеров на сервер (100 Мбит / 1 Гбит)	Требований нет	>= 1 Гбит/с

Требования к наличию ПО в ОС Linux

В ОС Linux должно быть установлено:

- Docker;
- Инструментальное средство Docker Compose.

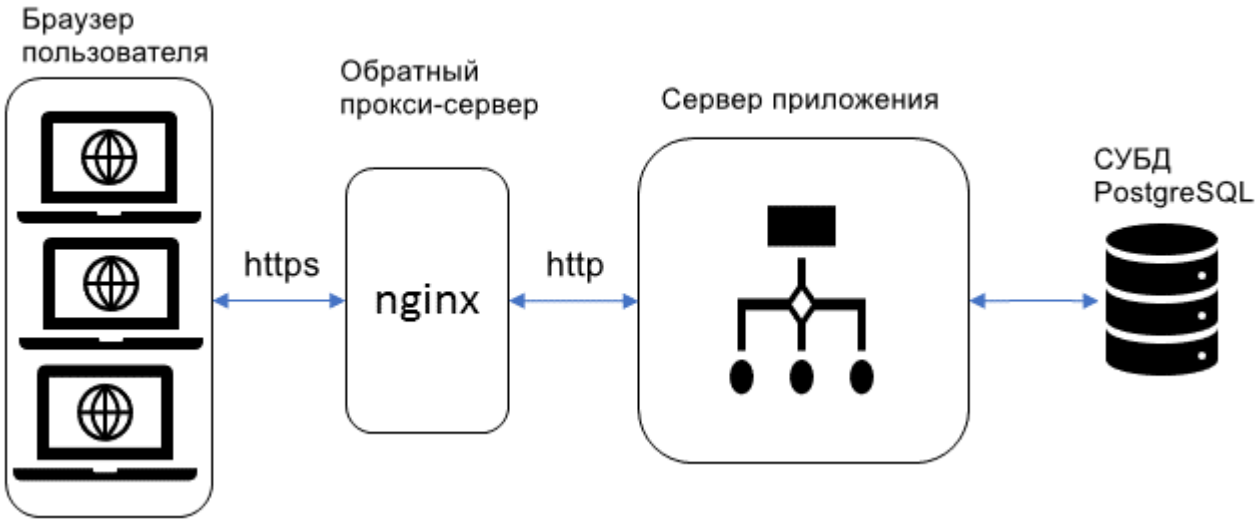
> **ПРИМЕЧАНИЕ**
>
> При установке Docker, Docker Compose следует руководствоваться справочными материалами конкрентной ОС.

Состав инсталляционного пакета

Включает в себя:

- docker-образ приложения;
- архив конфигурационных файлов eam-config-files.zip.
- инструменты для обновления версии базы данных (обновлении версии приложения):
  - приложение **efbundle** для ОС Linux;
  - скрипт **EAMDbUpdateScript.sql** для ОС Linux, Windows.

Схема работы



Приложение работает с СУБД PostgreSQL и имеет встроенный веб-сервер, который работает по протоколу http.

В руководстве приведена настройка конфигурационных файлов приложения и приведен пример docker-compose.yml файла:

- для работы по протоколу http (СУБД PostgreSQL и приложение запускаются с помощью docker);
- для работы по протоколу https (СУБД PostgreSQL и приложение запускаются с помощью docker, NGINX развернут отдельно);
- для работы по протоколу https (PostgreSQL развернут отдельно) (приложение запускается с помощью docker, NGINX и PostgreSQL развернуты отдельно).

> **ПРИМЕЧАНИЕ**
>
> Установка **NGINX, PostgreSQL** в данном руководстве не рассматривается.

Распаковка конфигурационных файлов

Распаковать архив с конфигурационными файлами **eam-config-files.zip** в папку **eam**.
Например, */home/user/eam*.

Должна получится следующая структура:

```
eam
└─docker-compose.yml
└─postgres.conf
└─eam-config
  ├─nlog.config
  ├─appsettings.Production.json
  └─clientsettings.Production.json
  └─Model.xafml
```

Добавление docker-образа приложения в локальный репозиторий

В консоли перейти в папку, где находится образ приложения, и загрузить образ в локальный репозиторий командой:

```
docker load -i Galaktika.Eam-11.1.0-alpha.52.tar.gz
```

Настройка приложения для работы по http

Описание настроек приведено в следующих конфигурационных файлах:
appsettings.Production.json
clientsettings.Production.json
docker-compose.yml

Правка clientsettings.Production.json

В конфигурационном файле приложения `clientsettings.Production.json` требуется:

Указать имя hostname сервера/машины, где будет запускаться контейнер с приложением и имя приложения (**eam**).

```
"host": "http://demo.galaktika.ru/eam",
```

> **ПРИМЕЧАНИЕ**
>
> Hostname можно узнать, если ввести в консоли команду hostname. Например, строка в конфигурационном файле примет вид: "host": "http://hostname/eam"

Имя приложения **eam**, должно быть одинаковым также в конфигурационном файле appsettings.Production.json.

Также можно изменить оформление приложения по умолчанию.

```
"skin": "Compact"
```

Для настройки доменной авторизации требуется добавить REALM и адрес запуска приложения.

```
"oidc-client": {
        "authority": "https://auth.galaktika.local/realms/galaktika",
        "client_id": "xum_development",
        "response_type": "id_token token",
        "scope": "openid profile xum",
        "popup_redirect_uri": "http://demo.galaktika.ru/eam/logon",
        "silent_redirect_uri": "http://demo.galaktika.ru/eam/silentcallback",
        "redirect_uri": "http://demo.galaktika.ru/eam/logon",
        "post_logout_redirect_uri": "http://demo.galaktika.ru/eam",
        "accessTokenExpiringNotificationTime": 60,
        "automaticSilentRenew": true,
        "loadUserInfo": true,
        "logLevel": "None"
        }
```

Включение доменной авторизации описано в данной статье.

Пример полного содержания файла приложения `clientsettings.Production.json`:

```
{
    "application": {
        "theme": {
                "name": "Webix",
                "skin": "Compact"
            },
        "apps": [
            {
                "host": "http://demo.galaktika.ru/eam",
                "localStorage": true
            }
        ]
    },
    "runtime": {
        "debugMode": false,
        "oidc-client": {
                "authority": "https://auth.galaktika.local/realms/galaktika",
                "client_id": "xum_development",
                "response_type": "id_token token",
                "scope": "openid profile xum",
                "popup_redirect_uri": "http://demo.galaktika.ru/eam/logon",
                "silent_redirect_uri": "http://demo.galaktika.ru/eam/silentcallback",
                "redirect_uri": "http://demo.galaktika.ru/eam/logon",
                "post_logout_redirect_uri": "http://demo.galaktika.ru/eam",
                "accessTokenExpiringNotificationTime": 60,
                "automaticSilentRenew": true,
                "loadUserInfo": true,
                "logLevel": "None"
                }
    }
}
```

Правка appsettings.Production.json

В конфигурационном файле `appsettings.Production.json` указывается имя приложения (**eam**), которое должно совпадать с именем в конфигурационном файле clientsettings.Production.json:

```
"PathBase": "/eam"
```

Указывается сервер БД, данные аутентификации на сервере БД, имя БД:

```
"Default": "Server=ru01-vm06;User ID=admineam;Password=pass;Pooling=false; Persist Security Info=true; DataBase=eam_DB"
```

Для включения доменной авторизации с помощью KeyCloak требуется установить значение *Oidc* (стандартная аутентификация - значение *Server*).
Также необходимо указазать адрес **Realm**.

```
"Authentication": {
  "UserNameClaim": "preferred_username",
  "Type": "Oidc",
  "OpenIdAuthentication": {
    "Authority": "https://auth.galaktika.local/realms/galaktika",
    "HTTPMode": false,
    "ValidateIssuer": true,
    "ValidIssuer": "https://auth.galaktika.local/realms/galaktika",
```

Ниже приведено полное содержимое конфигурационного файла, вышеописанные параметры выделены заливкой.

```
{
  "Web": {
    "compressionLevel": "Optimal"
  },
  "Host": {
    "PathBase": "/eam"
  },
  "Application": {
    "DebugMode": false
  },
  "Logging": {
    "LogLevel": {
      "Default": "Error"
    }
```

```
  },
  "Authentication": {
    "UserNameClaim": "preferred_username",
    "Type": "Oidc",
    "OpenIdAuthentication": {
      "Authority": "https://auth.galaktika.local/realms/galaktika",
      "HTTPMode": false,
      "ValidateIssuer": true,
      "ValidIssuer": "https://auth.galaktika.local/realms/galaktika",
      "ValidateAudience": false,
      "ValidateLifetime": true
    }
  },
      "ConnectionStrings": {
      "Default": "Server=ru01-vm06;User ID=admineam;Password=pass;Pooling=false; Persist
Security Info=true; DataBase=eam_DB"
    },
  "Sezal": {
    "EnableDiagnosticActions": false,
    "Extensions": {
      "AppHost": {
        "Modules": "Galaktika.Eam.Views.Xum;Galaktika.Eam.Core.DataViewer.Views"
      },
      "Reports": {
        "Xum": {
          "ExportedDocumentsDirectory": {
            "AbsolutePath": "var/tmp"
            //"RelativePath": "ExportedDocuments"
          }
        }
      }
    },
    "DaperPooling": {
      "Enable": true,
      "PoolSize": 8,
      "MaxConnections": 96
    },
    "Eam": {
      "EnableModelCache": false
    }
  }
}
```

Правка docker-compose.yml

Содержимое `docker-compose.yml` приведено ниже:

```
version: '3.8'

services:
  eam-postgres:
    image: postgres:15.3
    ports:
      - "5432:5432"
    restart: always
    container_name: eam-postgres
    environment:
      TZ: 'Europe/Moscow'
      PGTZ: 'Europe/Moscow'
      POSTGRES_PASSWORD: postgres
      POSTGRES_USER: postgres
    volumes:
      - postgres:/var/lib/postgresql/data
      - ./postgres.conf:/etc/postgresql/postgresql.conf
    command: -c 'config_file=/etc/postgresql/postgresql.conf'

  eam-app:
    restart: always
    image: galaktika.eam:11.1.0-alpha.52
    ports:
      - "80:80"
    container_name: eam-app
    command: -u
    volumes:
      - "/etc/ssl/certs:/etc/ssl/certs:ro" #для доменной авторизации KeyCloak
      - "./eam-config/nlog.config:/app/nlog.config:ro"
      - "./eam-config/appsettings.Production.json:/app/appsettings.Production.json:ro"
      - "./eam-config/clientsettings.Production.json:/app/clientsettings.Production.json:ro"
      - "./eam-config/Model.xml:/app/Model.xml:ro"
      - "app-logs:/var/log/eam-app"
    environment:
      - ASPNETCORE_ENVIRONMENT=Production
      - ASPNETCORE_URLS=http://*:80
      - Sezal_Environment=Production
      - ASPNETCORE_FORWARDEDHEADERS_ENABLED=true
      - TZ=Europe/Moscow
    security_opt:
      - seccomp=unconfined
```

```
    depends_on:
      - eam-postgres

volumes:
  postgres:
  app-logs:
    external: false
```

Настройка приложения для работы по https

В данной статье будет описана разница в конфигурационных настройках по сравнению с настройками работы по http протоколу:

- Настройка виртуального хоста nginx для приложения;
- Правка clientsettings.Production.json;
- Правка docker-compose.yml.

Все остальные действия - идентичны:

- правка appsettings.Production.json;
- обновление версии базы данных;
- запуск docker-контейнера;
- запуск приложения;
- подключение справки.

Настройка виртуального хоста nginx для приложения

> **ПРИМЕЧАНИЕ**
>
> Для установки **NGINX** необходимо руководствоваться следующими материалами.

Настройка виртуального хоста nginx для приложения приведена ниже:

```
map $http_connection $connection_upgrade {
  "~*Upgrade" $http_connection;
  default keep-alive;
}

server {
    listen 443 ssl;
    server_name demo.galaktika.ru;

    ssl on;
    ssl_certificate      /etc/nginx/certs/galaktika.pem;
    ssl_certificate_key  /etc/nginx/certs/galaktika.key;

    location /eam {
        proxy_pass_header Authorization;
        proxy_pass http://localhost:10020;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        proxy_buffering off;
        client_max_body_size 0;
        proxy_read_timeout 36000s;
        proxy_redirect off;
        proxy_ssl_session_reuse off;
    }

    location /eam/notificationHub {
        proxy_pass http://localhost:10020;

        # Configuration for WebSockets
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
```

```
        proxy_cache off;

        # WebSockets were implemented after http/1.0
        proxy_http_version 1.1;

        # Configuration for ServerSentEvents
        proxy_buffering off;

        # Configuration for LongPolling or if your KeepAliveInterval is longer than
60 seconds
        proxy_read_timeout 600s;

        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_ssl_session_reuse off;
    }
}
```

Выделены следующие параметры:

- имя веб-сервера nginx - demo.galaktika.ru. Обращение к приложению будет просходить по адресу https://demo.galaktika.ru/eam;
- при обращении по адресу приложения https://demo.galaktika.ru/eam, nginx перенаправит запрос по порту 10020 на внутренний порт веб-сервера приложения Галактика EAM на платформе Sezal.

> **ПРИМЕЧАНИЕ**
>
> Имя приложения **eam** должно совпадать:
>
> - в файле настройки виртуального хоста nginx для Галактика EAM на платформе Sezal;
> - в clientsettings.Production.json;
> - в appsettings.Production.json.

> **ПРИМЕЧАНИЕ**
>
> Номер порта (*10020*) должен совпадать с номером порта, указанном в файле docker-compose.yml.

Правка clientsettings.Production.json

В конфигурационном файле приложения `clientsettings.Production.json`, указать имя hostname сервера/машины (протокол https), где будет запускаться контейнер с приложением.

```
{
    "application": {
        "theme": {
                "name": "Webix",
                "skin": "Compact"
            },
        "apps": [
            {
                "host": "https://demo.galaktika.ru/eam",
                "localStorage": true
            }
        ]
    },
    "runtime": {
        "debugMode": false,
        "oidc-client": {
                "authority": "https://auth.galaktika.local/realms/galaktika",
                "client_id": "xum_development",
                "response_type": "id_token token",
                "scope": "openid profile xum",
                "popup_redirect_uri": "https://demo.galaktika.ru/eamtest/logon",
                "silent_redirect_uri": "https://demo.galaktika.ru/eamtest/silentcallback",
                "redirect_uri": "https://demo.galaktika.ru/eamtest/logon",
                "post_logout_redirect_uri": "https://demo.galaktika.ru/eamtest",
                "accessTokenExpiringNotificationTime": 60,
                "automaticSilentRenew": true,
                "loadUserInfo": true,
                "logLevel": "None"
                }
    }
}
```

Остальные параметры описаны в данной статье.

Корректировка файла docker-compose.yml

В файле `docker-compose.yml` требуется указать порт (*10020*):

```
ports:
  - "10020:80"
```

> **ПРИМЕЧАНИЕ**
>
> Номер порта (*10020*) должен совпадать с номером порта, указанном в файле настройки виртуального хоста nginx для Галактика EAM на платформе Sezal.

Ниже приведено полное содержимое файла, вышеописанные параметры выделены заливкой.

```
version: '3.8'

services:
  eam-postgres:
    image: postgres:15.3
    ports:
      - "5432:5432"
    restart: always
    container_name: eam-postgres
    environment:
      TZ: 'Europe/Moscow'
      PGTZ: 'Europe/Moscow'
      POSTGRES_PASSWORD: postgres
      POSTGRES_USER: postgres
    volumes:
      - postgres:/var/lib/postgresql/data
      - ./postgres.conf:/etc/postgresql/postgresql.conf
    command: -c 'config_file=/etc/postgresql/postgresql.conf'

  eam-app:
    restart: always
    image: galaktika.eam:11.1.0-alpha.52
    ports:
      - "10020:80"
    container_name: eam-app
    command: -u
    volumes:
      - "/etc/ssl/certs:/etc/ssl/certs:ro" #для доменной авторизации KeyCloak
```

```
      - "./eam-config/nlog.config:/app/nlog.config:ro"
      - "./eam-config/appsettings.Production.json:/app/appsettings.Production.json:ro"
      - "./eam-config/clientsettings.Production.json:/app/clientsettings.Production.json:ro"
      - "./eam-config/Model.xml:/app/Model.xml:ro"
      - "app-logs:/var/log/eam-app"
    environment:
      - ASPNETCORE_ENVIRONMENT=Production
      - ASPNETCORE_URLS=http://*:80
      - Sezal_Environment=Production
      - ASPNETCORE_FORWARDEDHEADERS_ENABLED=true
      - TZ=Europe/Moscow
    security_opt:
      - seccomp=unconfined
    depends_on:
      - eam-postgres

volumes:
  postgres:
  app-logs:
    external: false
```

Пример docker-compose.yml (PostgreSQL запускается отдельно)

По сравнению с настройками приведенными в данной статье, убраны строки запуска контейнера с PostgreSQL.

```
version: '3.8'

services:
  eam-app:
    restart: always
    image: galaktika.eam:11.1.0-alpha.52
    ports:
      - "10020:80"
    container_name: eam-app
    environment:
      - ASPNETCORE_ENVIRONMENT=Production
      - ASPNETCORE_URLS=http://*:80
      - Sezal_Environment=Production
      - ASPNETCORE_FORWARDEDHEADERS_ENABLED=true
      - TZ=Europe/Moscow

    command: -u
    volumes:
      - "/etc/ssl/certs:/etc/ssl/certs:ro" #для доменной авторизации KeyCloak
      - "./eam-config/nlog.config:/app/nlog.config:ro"
      - "./eam-config/appsettings.Production.json:/app/appsettings.Production.json:ro"
      - "./eam-config/clientsettings.Production.json:/app/clientsettings.Production.json:ro"
      - "./eam-config/Model.xml:/app/Model.xml:ro"
      - "app-logs:/var/log/eam-app"

    security_opt:
      - seccomp=unconfined

volumes:
  app-logs:
    external: false
```

Обновление версии базы данных

ПРИМЕЧАНИЕ

Выполнение обновления версии базы данных производится любым удобным способом:

- или с помощью утилиты **efbundle**;
- или с помощью выполнения скрипта **EAMDbUpdateScript.sql**.

ВАЖНО

Перед первым запуском системы требуется создать пустую БД и затем использовать средства обновления версии базы данных.
Выполнение обновления версии базы данных требуется:

- при первоначальном запуске Системы (будет сгенерирована структура БД);
- при переходе на новую версию Системы.

Обновление версии базы данных с помощью утилиты efbundle

Для выполнения обновления версии базы данных требуется при работающем сервере СУБД PostgreSQL запустить утилиту **efbundle** со строкой подключения к БД.

> efbundle --connection "Server=postgres;User ID=postgres;Password=postgres;Pooling=false; Persist Security Info=true; DataBase=eam_db"

где:

- **Server** - имя сервера СУБД;
- **User ID**, **Password** - имя пользователя, пароль при подключении к СУБД;
- **DataBase** - имя БД.

После успешного выполнения обновления версии базы данных, утилита вернет ответ *done*.

Обновление версии базы данных с помощью скрипта EamDbUpdateScript.sql

Для выполнения обновления версии базы данных требуется в PgAdmin4 или в терминале PostgreSQL выполнить скрипт **EamDbUpdateScript.sql**.

Подключение справки

Справка к приложению поставляется в формате html.

Для изменения адреса портала справки следует использовать конфигурационный файл `Model.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<Application PreferredLanguage="ru">
  <Options>
    <SezalExtensions>
        <Help HelpSite="https://demo.galaktika.ru/pikts/eam-docs/map-view-help-pages"/>
    </SezalExtensions>
  </Options>
</Application>
```

Запуск docker-контейнера

В консоли перейти в папку eam и запустить контейнер командой:

> docker-compose up

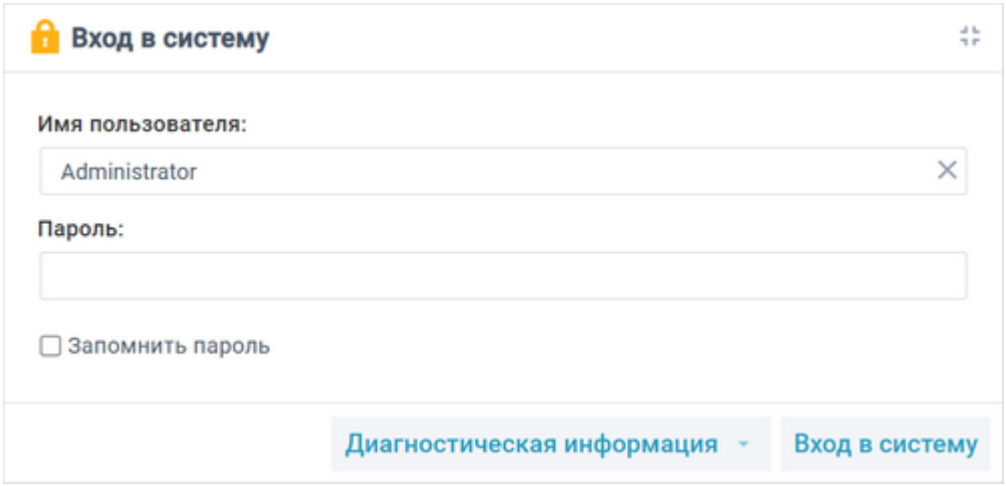
Для запуска контейнера в фоне следует использовать команду:

> docker-compose up -d

Запуск приложения

Открыть браузер и перейти по адресу http(s)://hostname/eam

В окне аутентификации ввести имя *Administrator* и нажать кнопку **Вход в систему**.



Обновление приложения

Обновление приложения происходит в следующем порядке:

- перейти в папку eam и остановить контейнер с приложением командой `docker-compose down`;
- добавить новую версию образа приложения в локальное хранилище;
- сделать бэкап базы данных любым доступным ПО (PgAdmin4 либо консольными утилитами);
- выполнить обновление версии базы данных;
- в папке eam, в файле `docker-compose.yml` изменить номер версии приложения до актуальной;

```
eam-app:
  restart: always
  image: galaktika.eam:11.1.0-alpha.52
```

- перейти в папку eam и запустить контейнер с приложение командой `docker-compose up -d`;
- залогиниться в приложении и проверить версию.