T2Plus Управление эффективностью активов PRO

Инструкция по установке приложения

Т2СОФТ

2024

Программные и аппаратные требования

Сервер базы данных

Программные требования:

- Postgres Pro Standard v15, Postgres Pro Enterprise v15, PostgreSQL v15.

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 6.0.4.

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.
- Браузер Coogle Chrome версии 63 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

Развертывание приложения T2Plus APM в linux с помощью Docker

Предварительные требования

В ОС Linux должен быть установлен Docker и инструментальное средство Docker Compose. Должен быть настроен доступ к интернет.

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

Инсталляционный пакет

Включает в себя:

- docker-образ приложения;
- архив конфигурационных файлов *apm-config-files.zip*.
- efbundle для миграции базы данных при обновлении приложения.

Схема работы

Приложение T2Plus APM работает с СУБД PostgreSQL и имеет встроенный веб-сервер, который работает по протоколу http.
Ниже будет рассмотрен пример docker-compose.yml файла, с помощью которого запускается PostgreSQL и приложение T2Plus APM.

Для работы приложения по протоколу https требуется настройка nginx. Установка *nginx* в данном руководстве не рассматривается.

Распаковка конфигурационных файлов

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

Должна получится следующая структура:

```
apm
└─docker-compose.yml
└─postgres.conf
└─apm-config
```

```
   ├─nlog.config
   ├─appsettings.Production.json
   └─clientsettings.Production.json
   └─Model.xafml
```

Пример `docker-compose.yml` приведен ниже:

```yaml
version: '3.8'

services:
  apm-postgres:
    image: postgres:15.3
    ports:
      - "5432:5432"
    restart: always
    container_name: apm-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'

  apm-app:
    restart: always
    image: t2plus.apmpro:14.0.1.6
    ports:
      - "80:80"
    container_name: apm-app
    command: -u
    volumes:
      - "./apm-config/nlog.config:/app/nlog.config:ro"
      - "./apm-config/appsettings.Production.json:/app/appsettings.Production.json:ro"
      - "./apm-config/clientsettings.Production.json:/app/clientsettings.Production.json:ro"
      - "./apm-config/Model.xml:/app/Model.xml:ro"
      - "app-logs:/var/log/apm-app"
    environment:
      - ASPNETCORE_ENVIRONMENT=Production
      - ASPNETCORE_URLS=http://*:80
      - Sezal_Environment=Production
      - ASPNETCORE_FORWARDEDHEADERS_ENABLED=true
    security_opt:
```

```
      - seccomp=unconfined
    depends_on:
      - apm-postgres

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

Добавление в локальный репозиторий контейнера с приложением

В консоли перейти в папку *apm* и загрузить образ T2Plus APM в локальный репозиторий командой:

> docker load -i T2Plus.ApmPro-14.0.1.14.tar.gz

Указание hostname в конфигурационном файле

В конфигурационном файле приложения `clientsettings.Production.json` указать имя hostname сервера/машины, где будет запускаться контейнер с T2Plus APM.

```
{
    "application": {
    "apps": [
        {
        "host": "http://t2soft/apm",
        "localStorage": true
        }
    ]
    },
    "runtime": {
        "debugMode": false
    }
}
```

ⓘ ПРИМЕЧАНИЕ

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

Миграция базы данных

Выполнение миграции требуется при первоначальном запуске Системы (будет сгенерирована структура БД), а также при переходе на новую версию Системы.
Для выполнения миграции базы данных требуется при работающем сервере СУБД PostgreSQL запустить утилиту **efbundle** со строкой подключения к БД.

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

где:

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

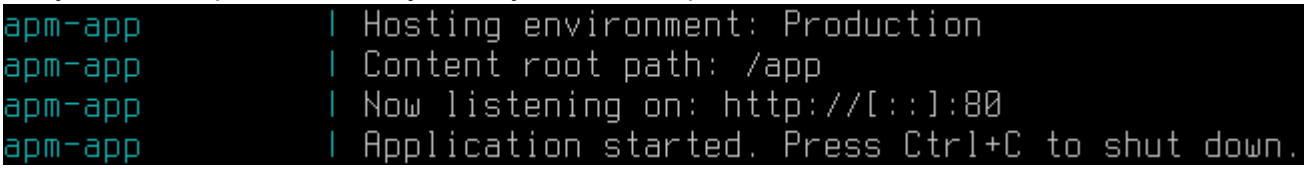
После успешного выполнения миграции, утилита вернет ответ *done*.

Запуск docker-контейнера с приложением

В консоли перейти в папку *apm* и запустить контейнер командой:

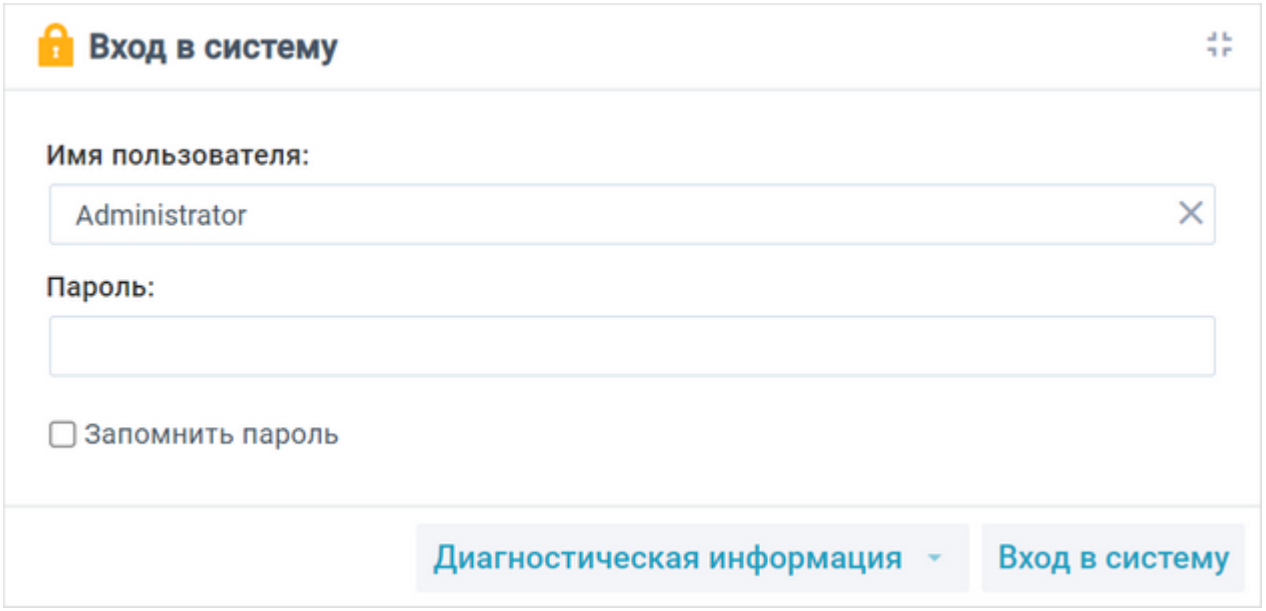
> docker-compose up

Запущенное приложение будет слушать 80 порт.



Запуск приложения APM

Открыть браузер и перейти по адресу http://hostname/apm
В окне аутентификации ввести имя *Administator* и нажать кнопку **Вход в систему**.



Настройка приложения для работы по протоколу https с помощью nginx

Ниже будет описана разница в конфигурационных настройках по сравнению с настройками работы по http протоколу. Все остальные действия, такие как миграция базы данных, запуск docker-контейнера , запуск приложения - идентичны.

Настройка хоста

Настройка хоста `apm` для nginx приведена ниже:

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

server {
    listen 443 ssl;
    server_name demo;

    ssl on;
    ssl_certificate      /etc/nginx/certs/t2soft.pem;
    ssl_certificate_key  /etc/nginx/certs/t2soft.key;

    location /apm {
        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 /apm/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. Обращение к приложению будет просходить по адресу https://demo/apm;
- при обращении по адресу приложения https://demo/apm, nginx перенаправит запросы на внутренний веб-сервер приложения.

Указание hostname в конфигурационном файле

В конфигурационном файле приложения `clientsettings.Production.json`,приведенном выше, указать имя hostname сервера/машины, где будет запускаться контейнер с APM.

```
{
    "application": {
    "apps": [
        {
        "host": "https://demo/apm",
        "localStorage": true
        }
    ]
    },
    "runtime": {
        "debugMode": false
    }
}
```

Корректировка файла docker-compose.yml

В файле `docker-compose.yml`, приведенном выше требуется изменить порт:

```
apm-app:
  restart: always
  image: t2plus.apmpro:14.0.1.6
  ports:
    - "10020:80"
  container_name: apm-app
  command: -u
  volumes:
```