



ПК «СИСТЕМА АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССОВ ЭНЕРГИТИЧЕСКИХ
КОМПАНИЙ»

Инструкция по развертыванию

Барнаул
2023

Содержание

Начало работы	3
Подготовка сервера:.....	3
Развертывание	4
Устанавливаем Portainer:	4
Настраиваем Portainer:	4
Создаем stack для приложения:	5
Установка PostgreSQL:	5
Установка pgAdmin:.....	6
Поднимаем nginx:.....	7
Установка Seq:	8
Установка jaeger:.....	8
Поднимаем сервис авторизации:.....	9
Поднимем клиент авторизации:	9
Поднимаем медиасервис:	10
Поднимаем сервис подразделений:.....	10
Поднимаем сервис СОТРУДНИКОВ-ИСПОЛНИТЕЛЕЙ:.....	11
Поднимаем сервис СХЕМЫ СЕТИ:.....	11
Поднимаем сервис linkeddatastorage:	12
Поднимаем сервис managementservice:.....	12
Поднимаем сервис ТОЧЕК УЧЕТА И СЧЕТЧИКОВ абонентов:	12
Поднимаем сервис exportdocxservice:.....	13
Поднимаем сервис exceleport:	13
Поднимаем сервис Тех. нарушения:.....	14
Поднимаем клиента Тех. нарушения:.....	14
Настраиваем DNS:	15

Начало работы

Важно. Образ виртуальной машины в формате Hyper-V. Необходима версия hyper-v сервера не ниже 2019.

При развертывании ТН на Ubuntu необходимо выполнить каждый шаг. Если разворачивается образ виртуальной машины, достаточно добавить DNS записи (последняя страница). В целях безопасности все пароли от регистров были удалены с образа. Не забудьте добавить их если разворачиваете ТН “с нуля”.

Данные для авторизации:

<http://sabpek-blazor.sabpek.com/>

- Логин - admin
- Пароль - admin

portainer:

- Логин - admin
- Пароль –задаётся после развёртывания docker-образа по инструкции ниже

[http://auth-sabpek_blazor.sabpek.com/:](http://auth-sabpek_blazor.sabpek.com/)

- Логин - admin
- Пароль - admin

pgadmin:

- Пароль - baguvix

postgres:

- user: postgres
- password: baguvix

Подготовка сервера:

1. Подключаемся по ssh к виртуальной машине с Ubuntu
2. Получаем root-права командой: **sudo -s** (далее вводим пароль и логин)
3. Обновляем информацию по пакетам: **apt upgrade**
4. Обновляем пакеты: **apt upgrade**
5. Устанавливаем Docker командой **apt install docker.io**

Развертывание

Устанавливаем Portainer:

1. Создаем хранилище данных для Portainer: **docker volume create portainer_data**
2. Устанавливаем и запускаем контейнер Portainer: **docker run -d -p 9000:9000 --name=portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce**. Эта команда настраивает автоматический запуск Portainer после перезагрузки, а также постоянное хранилище, чтобы Ваши настройки не потерялись при удалении и повторном разворачивании контейнера.
3. Убеждаемся, что portainer запущен: **docker ps**

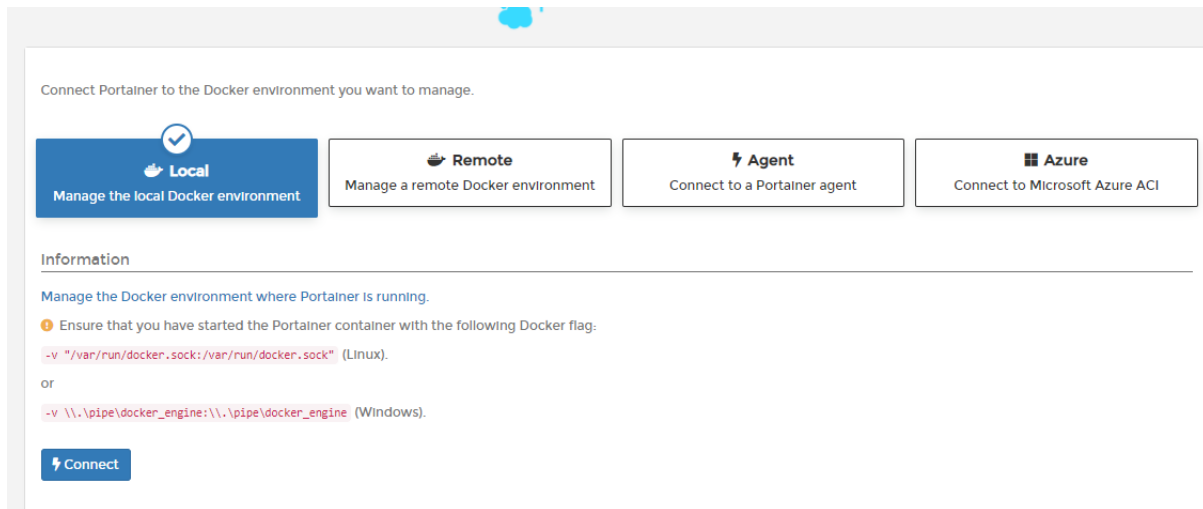
Настраиваем Portainer:

1. Получить доступ к программе Вы можете через веб-интерфейс на порту 9000. Откройте его в браузере. Введите имя пользователя и пароль, под



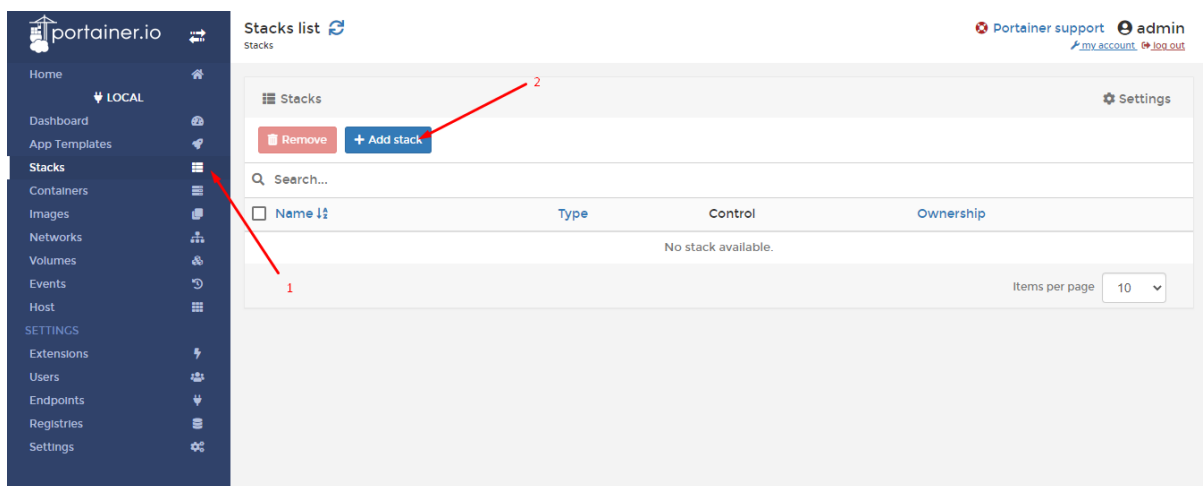
которым вы будете входить в систему:

2. Выберем метод подключения к Docker. Для начала можно подключиться к локальному сервису Docker. Для этого выберите Local;



- Используя любой ftp manager закидываем папку с конфигами на сервер (далее будем копировать их в нужные директории).

Создаем stack для приложения:



- Переходим на панель Stacks в Portainer;
- Нажимаем на Add stack;
- Вводим название stack и пишем версию в compose файл;
- Спускаемся вниз страницы и нажимаем 'Deploy the stack'.

Установка PostgreSQL:

- Добавляем в compose файл (блок services) конфигурацию для контейнера (необходимо подставить значения логина и пароля):

```
#POSTGRESQL
postgres:
  hostname: postgres
  container_name: postgres
```

```
image: postgres
environment:
  POSTGRES_USER: ${POSTGRES_USER:-postgres}
  POSTGRES_PASSWORD: ${POSTGRES_PASSWORD:-baguvix}
  PGDATA: /data/postgres
volumes:
  - postgres:/data/postgres
ports:
  - "5432:5432"
  - "6556:5432"
restart: always
```

2. Обновляем блок volumes:

```
volumes:
  postgres:
```

3. Нажимаем ‘Update the stack’ в portainer;
4. Далее нам понадобится 2 базы для сервисов (медиа сервис и сервис авторизации). Остальные сервисы используют EF core с подходом code first, поэтому базы для них создавать не нужно. Для создания оставшихся баз используются контейнеры.

- **sudo docker run -it --rm hub4sabpek/mediaservicecreatedatabase "User Id=postgres;Password=baguvix;Host=postgres;Database=mediaservice;Port=5432;Integrated Security=False"**
- **sudo docker run -it --rm hub4sabpek/authenticationcreatedatabase "User Id=postgres;Password=baguvix;Host=postgres;Database=authservice;Port=5432;Integrated Security=False"**

Установка pgAdmin:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера (необходимо подставить значения логина и пароля):

```
#PGADMIN
pgadmin:
  container_name: pgadmin
```

```
image: dpage/pgadmin4
environment:
  PGADMIN_DEFAULT_EMAIL: ${PGADMIN_DEFAULT_EMAIL:-
sabpek_blazor@sabpek.com}
  PGADMIN_DEFAULT_PASSWORD:
${PGADMIN_DEFAULT_PASSWORD:-baguvix}
  PGADMIN_CONFIG_SERVER_MODE: 'False'
volumes:
  - /home/developer/configs/pgadmin:/var/lib/pgadmin
ports:
  - "${PGADMIN_PORT:-5050}:80"
restart: always
```

2. Создаем директорию pgadmin по указанному в volumes пути;
3. Меняем право доступа к созданной папке
sudo chmod a+wrx {папка указанная в volumes}
4. Нажимаем 'Update the stack' в portainer;
5. Заходим в pgAdmin и подключаемся к бд.

Поднимаем nginx:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
nginx:
  hostname: nginx
  image: nginx:stable-alpine
  container_name: nginx
  environment:
    - "TZ=Asia/Barnaul"
  ports:
    - "80:80"
    - "443:443"
    - "8080:8080"
  volumes:
    - /var/wwwroot:/usr/share/nginx/html/
    - /home/developer/configs/nginx.conf:/etc/nginx/nginx.conf
    - /home/developer/certs:/etc/nginx/certs
  restart: always
```

2. Копируем соответствующий конфигурационный файл по пути указанному в блоке volumes и создаем wwwroot директорию;
3. Нажимаем ‘Update the stack’ в portainer.
Внимание! Он может вставать и падать до тех пор, пока оставшиеся сервисы не будут подняты

Установка Seq:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
#SEQ
seq:
  hostname: seq
  container_name: seq
  image: datalust/seq:latest
  environment:
    - ACCEPT_EULA=Y
  volumes:
    - /var/seqdata:/data
  ports:
    - "5433:80"
    - "5434:5341"
  restart: always
```

2. Создаем папку по указанному в volumes пути;
3. Меняем право доступа к созданной папке
sudo chmod a+wx {папка указанная в volumes}
4. Нажимаем ‘Update the stack’ в portainer.

Установка jaeger:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
#jaeger
jaeger:
  image: jaegertracing/all-in-one:1.22
  container_name: jaeger
  restart: always
  ports:
```



```
- 16686:16686  
environment:  
- JAEGER_AGENT_HOST=jaeger  
- COLLECTOR_ZIPKIN_HOST_PORT=:9411
```

2. Создаем папку по указанному в volumes пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис авторизации:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
#сервис авторизации  
auth-service:  
  hostname: auth-service  
  image: hub4sabpek/authentication-service:latest  
  container_name: auth-service  
  volumes:  
    - /home/developer/configs/auth.json:/app/appsettings.json  
    - /home/developer/configs/auth_key.pub:/app/key.pub  
    - /home/developer/configs/auth_key:/app/key  
  restart: always
```

2. Копируем соответствующий конфигурационный файл по пути, указанному в блоке volumes;
3. Нажимаем 'Update the stack' в portainer.

Поднимем клиент авторизации:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
#Клиент сервиса авторизации  
auth-client:  
  hostname: auth-client  
  image: hub4sabpek/authentication-client:latest  
  container_name: auth-client  
  ports:  
    - "9500:80"  
  volumes:
```

```
- /home/developer/configs/auth_client_config.js:/var/www/web/settings.js  
restart: always
```

2. Копируем соответствующий конфигурационный файл по пути, указанному в блоке volumes;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем медиасервис:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
#media сервис - хранилище  
mediaservice:  
  image: hub4sabpek/mediaserviceservice:secure  
  container_name: mediaservice  
  volumes:  
    - /home/developer/configs/mediaservice.json:/app/appsettings.json  
    - /var/share/media:/storage  
    - /home/developer/configs/mediaservice_key.pub:/app/key.pub  
  restart: always
```

2. Добавим файл конфигураций по указанному пути;
3. Добавим директорию media по пути указанному в volumes;
4. Поменяем права на директорию
sudo chmod a+wrx {папка указанная в volumes}
5. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис подразделений:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
# -- Сервис ПОДРАЗДЕЛЕНИЙ ----  
divisionsservice:  
  image: hub4sabpek/divisionsservice:latest  
  container_name: divisionsservice  
  restart: always  
  depends_on:  
    - postgres  
  volumes:  
    - /home/developer/configs/divisionsservice.json:/app/appsettings.json  
    - /home/developer/configs/divisionsservice_key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис СОТРУДНИКОВ-ИСПОЛНИТЕЛЕЙ:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
# -- Сервис СОТРУДНИКОВ-ИСПОЛНИТЕЛЕЙ ----
employeesservice:
  image: hub4sabpek/employeesservice:latest
  container_name: employeesservice
  restart: always
  depends_on:
    - postgres
  volumes:
    - /home/developer/configs/employeesservice.json:/app/appsettings.json
    - /home/developer/configs/employeesservice_key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис СХЕМЫ СЕТИ:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
# -- Сервис СХЕМЫ СЕТИ ----
energoobjectsservice:
  image: hub4sabpek/energoobjectsservice:latest
  deploy:
    resources:
      limits:
        cpus: '0.50'
        memory: 1024M
  container_name: energoobjectsservice
  restart: always
  depends_on:
    - postgres
  volumes:
    - /home/developer/configs/energoobjectsservice.json:/app/appsettings.json
    - /home/developer/configs/key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис linkeddatastorage:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
linkeddatastorage:
  image: hub4sabpek/referenceslinkeddatastorageservice:latest
  container_name: linkeddatastorage
  restart: always
  depends_on:
    - postgres
  volumes:
    - /home/developer/configs/linkeddatastorage.json:/app/appsettings.json
    - /home/developer/configs/key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис managementservice:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
managementservice:
  image: hub4sabpek/referencesmanagementservice:dev
  container_name: managementservice
  restart: always
  depends_on:
    - postgres
  volumes:
    - /home/developer/configs/managementservice.json:/app/appsettings.json
    - /home/developer/configs/key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис ТОЧЕК УЧЕТА И СЧЕТЧИКОВ абонентов:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
# -- Сервис ТОЧЕК УЧЕТА И СЧЕТЧИКОВ абонентов ----
accountingpointsservice:
```

```
image: hub4sabpek/accountingpointsservice:latest
deploy:
  resources:
    limits:
      cpus: '0.50'
      memory: 1024M
  container_name: accountingpointsservice
  restart: always
  depends_on:
    - postgres
  volumes:
    - /home/developer/configs/accountingpointsservice.json:/app/appsettings.json
    - /home/developer/configs/key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис exportdocxservice:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
exportdocxservice:
  image: hub4sabpek/exportdocxservice:latest
  container_name: exportdocxservice
  restart: always
  volumes:
    - /home/developer/configs/exportdocxservice.json:/app/appsettings.json
    - /home/developer/configs/key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис excelexport:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
excelexport:
  image: hub4sabpek/exportspreadsheetservice:latest
  container_name: excelexport
```

```
restart: always
depends_on:
  - postgres
volumes:
  - /home/developer/configs/excelexport.json:/app/appsettings.json
  - /home/developer/configs/key.pub:/app/key.pub
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем сервис САБПЭК. Тех. нарушения:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
techalerts-service:
  hostname: techalerts-service
  image: hub4sabpek/techalertsservice:latest
  container_name: techalerts-service
  volumes:
    - /home/developer/configs/techalerts-service.json:/app/appsettings.json
  ports:
    - "9700:80"
  restart: always
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Поднимаем клиента САБПЭК. Тех. нарушения:

1. Добавляем в compose файл (блок services) конфигурацию для контейнера

```
techalerts-client:
  hostname: techalerts-client
  image: hub4sabpek/techalertsclient:latest
  container_name: techalerts-client
  depends_on:
    - postgres
  volumes:
    - /home/developer/configs/techalerts-
client.json:/var/www/web/appsettings.json
```

```
ports:  
  - "9800:80"  
restart: always
```

2. Добавим файл конфигураций по указанному пути;
3. Нажимаем 'Update the stack' в portainer.

Настраиваем DNS:

Большинство сервисов общаются по внутренней сети docker, но веб-интерфейс вынужден обращаться к хосту по имени. Для того, чтобы избежать лишнего конфигурирования, было использовано решение с DNS.

В своем dns необходимо добавить 4 имени для ip сервера на котором развернут Тех. нарушение:

- techalerts-sabpek_blazor.sabpek.com
- authapi-sabpek_blazor.sabpek.com
- auth-sabpek_blazor.sabpek.com
- sabpek_blazor.sabpek.com

Если нет возможности внести записи во внутренний DNS, то можно внести записи в hosts файл (C:\Windows\System32\drivers\etc\hosts). Тогда система будет работать только на том рабочем месте, где настроен hosts.

```
# The IP address and the host name should be separated by at least one  
# space.
```

```
IP address techalerts-sabpek_blazor.sabpek.com
```

```
IP address authapi-sabpek_blazor.sabpek.com
```

```
IP address auth-sabpek_blazor.sabpek.com
```

```
IP address sabpek_blazor.sabpek.com
```