Обновление промежуточного ПО (образа)

Промежуточное ПО, например, PHP или Apache также требуют обновления, что связано с их поддержкой, выпуском исправлений безопасности и т.п.

Самый простой способ сделать это в случае развертывания ALM при помощи Docker. Достаточно будет обновить необходимый образ по примеру ниже

```
docker pull devprom/alm-app:latest

cd /home/devprom/docker
git fetch
git merge

docker-compose down
docker-compose up -d
```

Данные команды выполняются в каталоге, из которого в первый раз устанавливалось ПО.

Миграция с предыдущей версии Docker-образа

Возможно, у вас используется предыдущая версия Docker-образа. Название контейнера по умолчанию - alm. Данный контейнер содержит как приложение, так и СУБД вместе с базой данных.

Необходимо выполнить миграцию на текущий вариант развертывания.

Предварительные шаги

Выполните резервное копирование в административном разделе приложения и дождитесь, пока система отобразит размер резервной копии в списке резервных копий.

Установите на ОС пакет docker-compose.

Выключение контейнера alm

Сперва необходимо корректно остановить СУБД, выполнив для этого следующие команды (ваш пароль devprom-пользователя может отличаться, пропишите его):

```
docker exec alm mysql --user=devprom --password= -e "set global
innodb_fast_shutdown=0"
docker exec alm service mysql stop
docker stop alm
```

Сохраните данные контейнера alm

Подготовьте скрипты для развертывания приложения:

```
git clone https://github.com/devprom-dev/docker.git
cd docker
```

Скопируйте базу из старого контейнера на хост:

```
docker cp alm:/var/lib/mysql ./dbdata
chown -R 101:101 ./dbdata
```

Скопируйте актуальные конфигурационные файлы Apache из старого контейнера на хост:

```
docker cp alm:/etc/apache2/sites-available/devprom.conf apache2/devprom.conf
docker cp alm:/etc/apache2/sites-available/ldap.conf apache2/ldap.conf
```

Измените имя хоста, где будет расположена СУБД, константе DB_HOST нужно установить значение db:

```
vi /var/www/devprom/htdocs/settings_server.php
```

Установка новых контейнеров

```
docker-compose up -d
```

Замените содержимое каталога с базой данных на базу из старого контейнера:

```
docker stop alm-db

rm -fr /var/lib/docker/volumes/docker_dbdata/_data/*
cp -R dbdata/* /var/lib/docker/volumes/docker_dbdata/_data/

docker start alm-db
```

Предоставьте доступ приложению для подключения к БД. Возможно, пароль devprom_pass потребуется заменить на тот, что записан в /var/www/devprom/htdocs/settings_server.php

```
docker exec alm-db mysql_upgrade -u root
docker exec alm-db mysql -u root --password= -e "CREATE USER 'devprom'@'%' IDENTIFIED
BY 'devprom_pass'"
docker exec alm-db mysql -u root --password= -e "GRANT ALL PRIVILEGES ON *.* TO
'devprom'@'%' WITH GRANT OPTION"
```

Приложение готово в работе.

Сохраняйте контейнер alm до тех пор, пока не убедитесь в полной работоспособности приложения. Вы можете вернуться к прежней конфигурации при помощи команд:

```
docker-compose down
docker start alm
```