РАЗВЕРТЫВАНИЕ ПРИЛОЖЕНИЙ И САЙТОВ С ПОМОЩЬЮ DOCKER

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ганецкий А.О.

Сацук С.М. – к.т.н., доцент

В настоящее время существует ряд трудностей связанных с развертыванием новых сайтов или приложений. Довольно проблематично перенести сайт или приложение на другой сервер, в следствии необходимости установить такое программное окружение, которое было на компьютере разработчиков. Для решения этой проблемы есть несколько способов: установочный скрипт, облачные сервисы и виртуализация. У данных решений есть, кроме прочих, один очень существенный недостаток- маштабируемоть. Разработчикам придется переустанавливать заново программы для каждого из серверов. Мной предложено решение, которое решает данную проблему.

Использование Docker`а упрощает задачу тем, что разработчикам необходимо один раз написать скрипт для развертывания программного окружения, и использовать его для любого нового сервера. Docker позволяет разворачивать окружения полностью идентичные оригиналу, а так же используя один раз написанный скрипт, можно развернуть неограниченное количество одинаковых серверов.

Подобно виртуальной машине Docker запускает свои процессы в собственной, заранее настроенной операционной системе. Но при этом все процессы Docker работают на физическом-host сервере деля все процессоры и всю доступную память с другими процессами, запущенными в host-системе. Подход, используемый Docker, находится посередине между запуском всего на физическом сервере и полной виртуализацией, предлагаемой виртуальными машинами.

Для развертывания приложения или сайта на php, необходим стек LAMP. Мной было использовано в качестве исходного образа для Docker- Ubuntu 16.04.

FROM ubuntu:16.04

Использование официального образа ubuntu в качестве основы необходимо с точки зрения безопасности. Так для данного стека будет необходимо настроить таймзону без использования интерактивного режима.

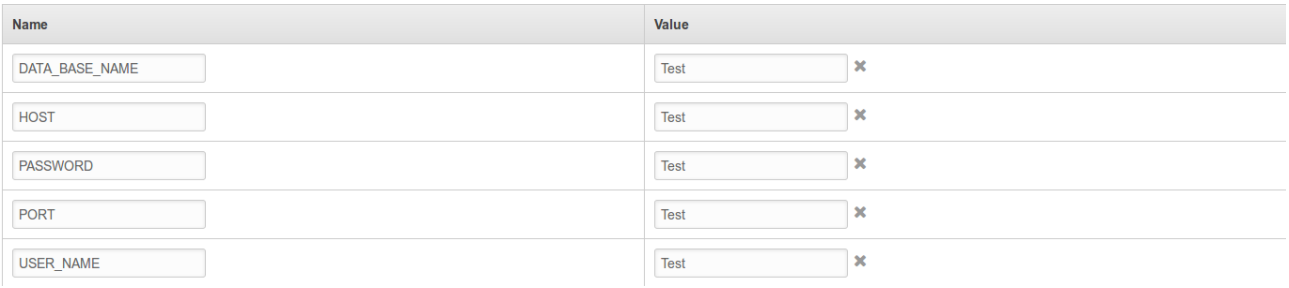
```
ENV                 TZ
'Europe/Minsk'

RUN echo $TZ > /etc/timezone && \
apt-get update && apt-get install -y tzdata && \
rm /etc/localtime && \
ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && \
dpkg-reconfigure -f noninteractive tzdata && \
apt-get clean
```

Для настройки php в режиме fpm обычно нужно переписывать некоторые настройки в конфигах php. Я создал файл настройки зарание и после установки удаляется старый файл и копирую созданный мной на его место, после этого php начнет работать в режиме fpm.

Для поддержания работы данного стека необходим сервис, который будет следить, чтобы корректно работали apache и php-fpm. В случаи ошибки в данных программах, этот сервис должен будет в кратчайшие сроки перезапустить программы, и продолжать отслеживать их работаспособность. Отлично подходит сервис supervisor. Установка “демона” этого сервиса в качестве рабочего процесса, позволяет быть уверенным что контейнер будет работать вместе со всеми сервисами. Для добавления apache и php-fpm в процессы, за которыми следит supervisor, необходимо было прописать два файла настройки, по одному для каждого, и положить их в корневую папку сервиса: /etc/supervisor/conf.d/

В качестве базы данных для php был выбран mysql, размещенный на Amazon RDS. Данный сервис позволяет достаточно безопасно хранить данные, быстро обеспечивать к ним доступ, и производить постоянное резервное копирование информации. При необходимости перезапуска контейнера (при появлении ошибок в контейнере, он сразу будет выключаться и на его место развернут точно такой же, а потом будет произведен поиск ошибок и исправление их в дальнейшем) база данных, расположенная вне контейнера не “умрет” вместе с ним. Данное решение позволяет быстрее перезапускать контейнеры, в случаи необходимости, без потери информации из базы данных. Ключи от базы данных передаются через переменные окружения, их нет в репозиториях разработки. Данный сервис разрабатывался для развертывания на Amazon Elastic Beanstalk. И заходя в аккаунт, если у данного разработчика есть доступ, он может просмотреть, добавить или удалить ключи от базы данных.



Для передачи статических данных проект помещается в git, откуда каждый разработчик\тестировщик проекта, может скачать проект и развернуть его локально, если этого будет достаточно, или глобально, при наличии доступа к аккаунту AWS EBS.

Данный Dockerfile соберет стек LAP-fpm (Linux(Ubuntu 16.04), Apache2, PHP(в режиме fpm)), и позволит развернуть за несколько минут новый блог на WordPress, сайт на Drupal, а так же огромное количество приложений, которым необходим данный стек.

Данная настройка контейнера находится в открытом доступе, и каждый, при желании или необходимости, может просмотреть и изучить все более подробно.

GitHub: https://github.com/nubochistka/prod

Таким образом с помощью Docker были решены следующие задачи:

1. удобная передача серверного проекта клиенту
2. обеспечение тиражируемости серверов
3. обеспечение переиспользуемости ранее созданных серверных конфигураций

Необходимо отметить, что Docker также крайне удобен для обновления ранее установленных версий продукта и для создания тестовых серверов, полностью идентичных «натуральным».

Список использованных источников:

1. https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create_deploy_docker.html Документация AWS для работы с Docker.
2. https://docs.docker.com/ Официальная документация Docker.