



Зарождение

Резюме: этот документ представляет собой упражнение, связанное с системным администрированием.

Версия: 1

СОДЕРЖАНИЕ

I	Преамбула	2
II	Вступление	3
III	Общие рекомендации	4
IV	Обязательная часть	5
V	Бонусная часть	9
VI	Представление и экспертная оценка	10

Глава I

Преамбула



Глава II.

Вступление

Этот проект направлен на расширение ваших знаний в области системного администрирования с помощью Docker. Вы виртуализируете несколько образов Docker, создав их на своей новой персональной виртуальной машине.

Глава III.

Общие рекомендации

- Этот проект нужно выполнить на виртуальной машине.
- Все файлы, необходимые для конфигурации вашего проекта, должны быть помещены в `srcs` папка.
- А `Makefile` также является обязательным и должен находиться в корне вашего каталога. Он должен настроить все ваше приложение (т.е. он должен создавать образы Docker, используя `docker-compose.yml`).
- Этот предмет требует применения на практике концепций, которые, в зависимости от вашего опыта, вы, возможно, еще не усвоили. Поэтому мы советуем вам не стесняться читать много документации, связанной с использованием Docker, а также всего остального, что вы сочтете полезным, чтобы выполнить это задание.

Глава IV.

Обязательная часть

Этот проект заключается в том, что вы настраиваете небольшую инфраструктуру, состоящую из различных служб по определенным правилам. Весь проект должен быть выполнен на виртуальной машине. Вы должны использовать докер-сочинять.

У каждого образа Docker должно быть то же имя, что и у соответствующей службы. Каждая служба должна работать в выделенном контейнере.

По соображениям производительности контейнеры должны быть созданы либо из предпоследней стабильной версии Alpine Linux, либо из Debian Buster. Выбор остается за вами.

Вы также должны написать свой собственный Dockerfiles, по одному на каждую услугу. В Dockerfiles должен быть вызван в ваш docker-compose.yml По вашему Makefile.

Это означает, что вы должны сами создавать образы Docker для своего проекта. Затем запрещается извлекать готовые образы Docker, а также использовать такие службы, как DockerHub (Alpine / Debian исключены из этого правила).

Затем вам необходимо настроить:

- Контейнер Docker, содержащий только NGINX с TLSv1.2 или TLSv1.3.
- Контейнер Docker, содержащий WordPress + php-fpm (он должен быть установлен и настроен) только без nginx.
- Контейнер Docker, содержащий MariaDB только без nginx.
- Том, содержащий вашу базу данных WordPress.
- Второй том, содержащий файлы вашего сайта WordPress.
- А докер-сеть который устанавливает связь между вашими контейнерами.

В случае сбоя ваши контейнеры должны перезапускаться.



Контейнер Docker - это не виртуальная машина. Таким образом, не рекомендуется использовать какие-либо хакерские патчи, основанные на 'tail -f' и т. Д., При попытке его запустить. Прочтите о том, как работают демоны и стоит ли их использовать.



Конечно, использование `network: host` или `--link` или `links:` запрещено. Сетевая строка должна присутствовать в вашем файле `docker-compose.yml`. Ваши контейнеры не должны запускаться командой, выполняющей бесконечный цикл. Таким образом, это также относится к любой команде, используемой в качестве точки входа или в сценариях точки входа. Ниже приведены несколько запрещенных хакерских патчей: `tail -f`, `bash`, `sleep infinity`, `while true`.



Прочтите о PID 1 и лучших методах написания файлов Docker.

- В вашей базе данных WordPress должно быть два пользователя, один из которых является администратором. Имя пользователя администратора не может содержать `admin` / `Admin` или `administrator` / `Administrator` (например, `admin`, `administrator`, `Administrator`, `admin-123` и т. Д.).



Ваши тома будут доступны в папке `/ home / login / data` на хост-машине с помощью Docker. Разумеется, вам придется сменить логин на свой.

Чтобы упростить задачу, вы должны настроить свое доменное имя так, чтобы оно указывало на ваш локальный IP-адрес.

Это доменное имя должно быть `login.42.fr`. Опять же, вы должны использовать свой собственный логин.

Например, если ваш логин - `Wil`, `wil.42.fr` будет перенаправлять на IP-адрес, указывающий на веб-сайт `Wil`.



Последний тег запрещен.

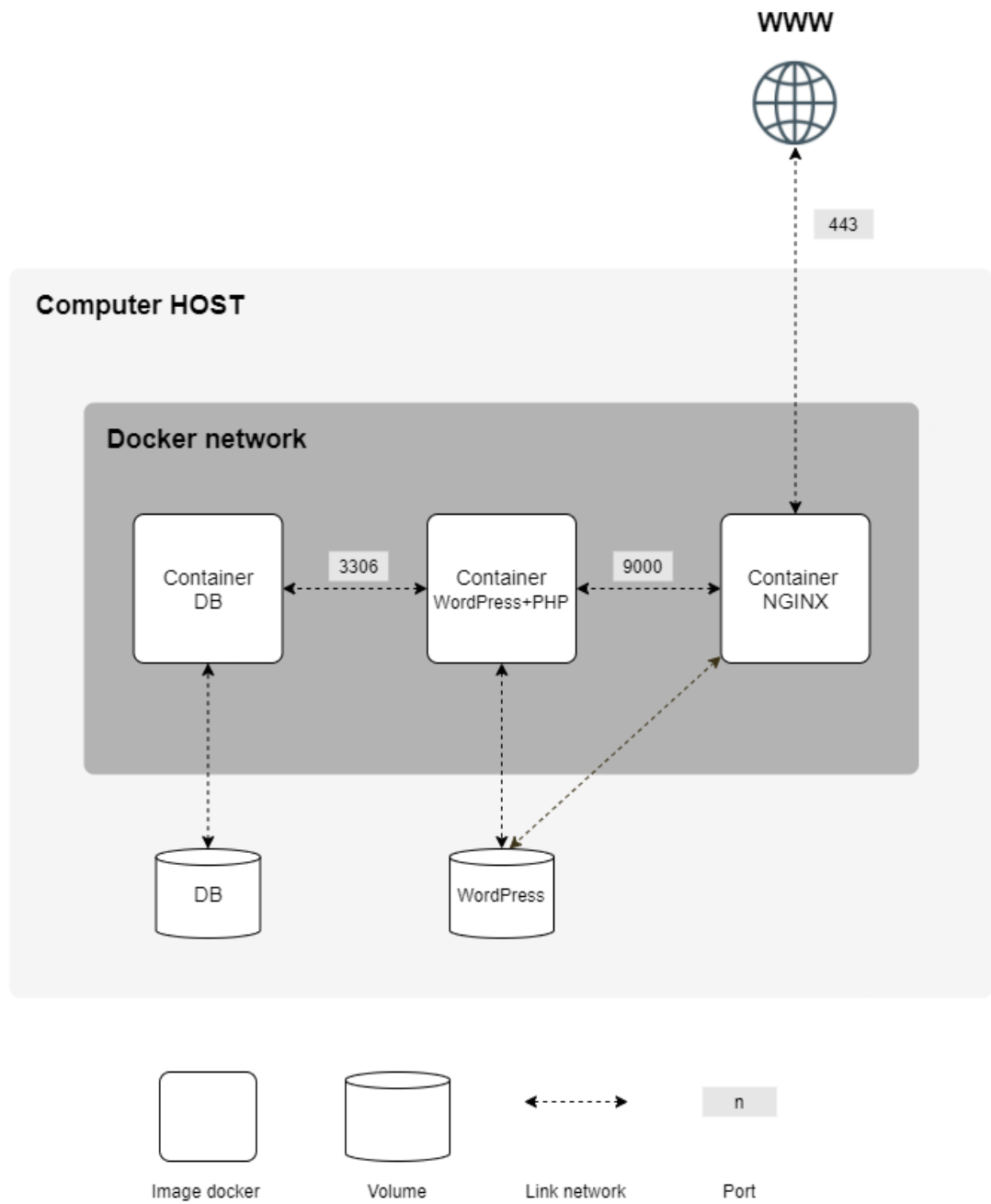
В ваших файлах `Dockerfiles` не должно быть пароля.

Обязательно использовать переменные среды.

Также настоятельно рекомендуется использовать файл `.env` для хранения переменных среды. Файл `.env` должен находиться в корне каталога `srcs`.

Контейнер NGINX должен быть единственной точкой входа в вашу инфраструктуру только через порт 443, используя TLSv1.2 или TLSv1.3. протокол.

Вот пример диаграммы ожидаемого результата:



Ниже приведен пример ожидаемой структуры каталогов:

```
$> ls -alR
всего XX
drwxrwxr-x 3 и будет 4096 апреля 42 20:42.
drwxrwxrwt 17 иля 4096 апреля 42 20:42 ..
- rw-rw-r-- 1 будет XXXX аврил 42 20:42 Makefile drwxrwxr-
x 3 будет иметь 4096 аврил 42 20:42 srcs

./ srcs:
всего XX
drwxrwxr-x 3 и будет 4096 апреля 42 20:42.
drwxrwxr-x 3 ив 4096 апреля 42 20:42 ..
- rw-rw-r-- 1 будет XXXX апрель 42 20:42 docker-compose.yml
- rw-rw-r-- 1 wil XXXX avril 42 20:42 .env drwxrwxr-x 5 wil 4096
avril 42 20:42 требования

./ srcs / требования:
всего XX
drwxrwxr-x 5 и будет 4096 апреля 42 20:42. drwxrwxr-x 3
wil 4096 avril 42 20:42 .. drwxrwxr-x 4 wil 4096 avril 42 20:42
бонус drwxrwxr-x 4 wil 4096 avril 42 20:42 mariadb
drwxrwxr-x 4 wil 4096 avril 42 20 : 42 nginx drwxrwxr-x 4 wil
4096 avril 42 20:42 инструменты drwxrwxr-x 4 wil 4096 avril
42 20:42 wordpress

./ srcs / requirements / mariadb:
всего XX
drwxrwxr-x 4 и будет 4096 апреля 42 20:45.
drwxrwxr-x 5 wil 4096 avril 42 20:42 .. drwxrwxr-x 2 wil
4096 avril 42 20:42 conf
- rw-rw-r-- 1 будет XXXX апрель 42 20:42 Dockerfile
- rw-rw-r-- 1 будет XXXX аврил 42 20:42 .dockerignore drwxrwxr-
x 2 wil 4096 avril 42 20:42 инструменты [...]

./ srcs / requirements / nginx:
всего XX
drwxrwxr-x 4 и будет 4096 апреля 42 20:42.
drwxrwxr-x 5 wil 4096 avril 42 20:42 .. drwxrwxr-x 2 wil
4096 avril 42 20:42 conf
- rw-rw-r-- 1 будет XXXX апрель 42 20:42 Dockerfile
- rw-rw-r-- 1 будет XXXX аврил 42 20:42 .dockerignore drwxrwxr-
x 2 wil 4096 avril 42 20:42 инструменты [...]

$> кошка srcs / .env
DOMAIN_NAME = wil.42.fr
# сертификата
ЦЕРТЫ _ = . / XXXXXXXXXXXXX
# НАСТРОЙКА MYSQL
MYSQL_ROOT_PASSWORD = XXXXXXXXXXXXX
MYSQL_USER = XXXXXXXXXXXXX
MYSQL_PASSWORD =
XXXXXXXXXXXX [...]
$>
```

Глава V

Бонусная часть

В этом проекте бонусная часть должна быть простой.

Dockerfile должен быть написан для каждой дополнительной службы. Таким образом, каждый из них будет работать в своем собственном контейнере и, при необходимости, будет иметь свой выделенный том.

Список бонусов:

- Настраивать кеш Redis для вашего сайта WordPress, чтобы правильно управлять кешем.
- Настроить FTP сервер контейнер, указывающий на объем вашего сайта WordPress.
- Создайте простой статический веб-сайт на любом языке по вашему выбору, кроме PHP (да, PHP исключен!). Например, сайт-витрина или сайт для презентации вашего резюме.
- Настраивать Администратор.
- Настройте услугу по вашему выбору, которая, по вашему мнению, будет полезна. Во время защиты вам придется обосновать свой выбор.



Для завершения бонусной части у вас есть возможность настроить дополнительные услуги. В этом случае вы можете открыть больше портов в соответствии с вашими потребностями.



Бонусная часть будет оцениваться только в том случае, если обязательная часть **ИДЕАЛЬНА**. Идеально означает, что обязательная часть выполнена полностью и работает без сбоев. Если вы не прошли **ВСЕ** обязательные требования, ваша бонусная часть вообще не будет оцениваться.

Глава VI.

Представление и экспертная оценка

Сдавайте свое задание в Git репозиторий как обычно. Во время защиты будет оцениваться только работа внутри вашего репозитория. Не стесняйтесь дважды проверять имена ваших папок и файлов, чтобы убедиться, что они верны.