



# Inception

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

*Версия: 1.1*

# Содержание

<b>I</b>	<b>Преамбула</b>	<b>2</b>
<b>II</b>	<b>Введение</b>	<b>3</b>
<b>III</b>	<b>Общие указания</b>	<b>4</b>
<b>IV</b>	<b>Обязательная часть</b>	<b>5</b>
<b>V</b>	<b>Бонусная часть</b>	<b>9</b>
<b>VI</b>	<b>Представление и экспертная оценка</b>	<b>10</b>

# Глава I

## Преамбула



## Глава II

### Введение

Этот проект направлен на расширение ваших знаний о системном администрировании с помощью Docker. Вы будете виртуализировать несколько образов Docker, создав их на новой персональной виртуальной машине.

# Глава III Общие указания

- Этот проект должен быть выполнен на виртуальной машине.
- Все файлы, необходимые для конфигурации вашего проекта, должны быть помещены в каталог `srcs` папка.
- Также необходим `Makefile`, который должен быть расположен в корне вашего каталога. Он должен настроить все ваше приложение (т.е. собрать образы Docker с помощью `docker-compose.yml`).
- Этот предмет требует применения на практике концепций, которые, в зависимости от вашей подготовки, вы, возможно, еще не изучали. Поэтому мы советуем вам не стесняться читать много документации, связанной с использованием Docker, а также все остальное, что вы найдете полезным для выполнения этого задания.



## Глава IV

# Обязательная часть

Этот проект заключается в том, что вам необходимо создать небольшую инфраструктуру, состоящую из различных сервисов по определенным правилам. Весь проект должен быть выполнен на виртуальной машине. Вы должны использовать `docker compose`.

Каждый образ Docker должен иметь то же имя, что и соответствующий ему сервис. Каждая служба должна работать в отдельном контейнере.

Для повышения производительности контейнеры должны быть собраны либо из предпоследней стабильной версии Alpine Linux, либо из Debian Buster. Выбор за вами.

Вы также должны написать свои собственные Dockerfiles, по одному на каждый сервис. Dockerfiles должны быть вызваны в вашем `docker-compose.yml` вашим Makefile.

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

Затем необходимо выполнить настройку:

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

Ваши контейнеры должны перезапускаться в случае сбоя.



Контейнер Docker - это не виртуальная машина . Поэтомуне рекомендуется использовать какие-либо хакерские патчи, основанные на 'tail -f' и т.п., при попытке запустить его. Читайте о том, как работают демоны и стоит ли их использовать или нет.







Конечно, используя `network: host` или `--link` или `links:` запрещено. Строка `network` должна присутствовать в вашем файле `docker-compose.yml`. Ваши контейнеры не должны быть запущены командой, выполняющей бесконечный цикл. Таким образом, это также относится к любой команде, используемой в качестве точки входа или используемой в скриптах точки входа.

Ниже приведены несколько запрещенных хаки-патчей: `tail -f`, `bash`, `sleep infinity`, `while true`.



Читайте о PID 1 и лучших практиках написания Docker-файлов.

- В базе данных WordPress должно быть два пользователя, один из которых - администратор. Имя пользователя администратора не может содержать `admin/Admin` или `admin-istrator/Administrator` (например, `admin`, `administrator`, Администратор, `admin-123` и так далее).



Ваши тома будут доступны в папке `/home/login/data` хост-машины, использующей Docker. Конечно, вы должны заменить логин на свой.

Чтобы упростить задачу, необходимо настроить 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 перенаправлен на IP-адрес, указывающий на веб-сайт `wil`.



Последний тег запрещен.

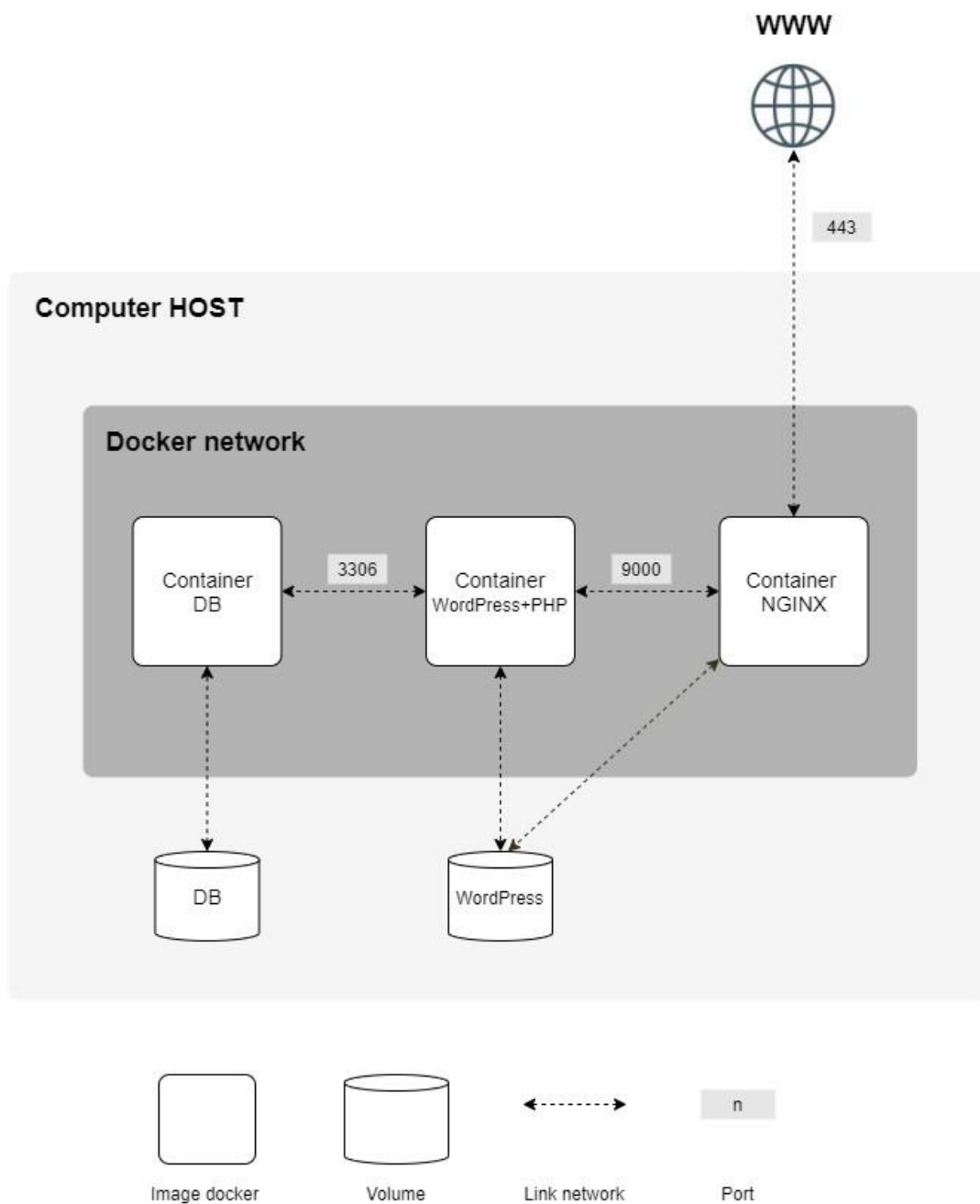
Пароль не должен присутствовать в ваших Dockerfiles. Обязательно использовать переменные окружения.

Кроме того, настоятельно рекомендуется использовать файл `.env` для хранения переменных окружения. Файл `.env` должен быть расположен в корне каталога `srcs`.

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



Вот пример диаграммы ожидаемого результата:



Ниже приведен пример предполагаемой структуры каталогов:

```
$> ls -alR
total XX
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 .
drwxrwxrwt 17 wil wil 4096 avril 42 20:42 ...
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Makefile
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 srcs

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

./srcs/requirements:
всего XX
drwxrwxr-x 5 wil wil 4096 аври 42 20:42 .
    л
drwxrwxr-x 3 wil wil 4096 аври 42 20:42 ..
    л
drwxrwxr-x 4 wil wil 4096 аври 42 20:42 бонус
    л
drwxrwxr-x 4 wil wil 4096 аври 42 20:42 mariadb
    л
drwxrwxr-x 4 wil wil 4096 аври 42 20:42 nginx
    л
drwxrwxr-x 4 wil wil 4096 аври 42 20:42 инструм
    л          енты
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 wordpress

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

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

$> cat srcs/.env
DOMAIN_NAME=wil.42.fr
# сертификаты
CERTS_=/XXXXXXXXXXXX
XX# НАСТРОЙКА
MYSQL
MYSQL_ROOT_PASSWORD=XXXXXXXXXX
MYSQL_USER=XXXXXXXXXX
MYSQL_PASSWORD=XXXXXXXXXX
l...l-
$>
```



# Глава V

## Бонусная часть

Для этого проекта бонусная часть должна быть простой.

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

Список бонусов:

- Настройте кэш redis для вашего сайта WordPress, чтобы правильно управлять кэшем.
- Установите контейнер FTP-сервера, указывающий на том вашего сайта WordPress.
- Создайте простой статический сайт на выбранном вами языке, кроме PHP (да, PHP исключен!). Например, сайт-витрина или сайт для представления своего резюме.
- Настройте администратора.
- Создайте услугу по своему выбору, которую вы считаете полезной. Во время защиты вы должны будете 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 оценивается только в том случае, если обязательная часть выполнена безупречно. Совершенство означает, что обязательная часть выполнена полностью и работает без сбоев.

Если вы не выполнили ВСЕ обязательные требования, ваша бонусная часть не будет оцениваться вообще.



## Глава VI

# Представление и экспертная оценка

Сдайте задание в свой Git-репозиторий, как обычно. Во время защиты будет оцениваться только та работа, которая находится в вашем репозитории. Не стесняйтесь дважды проверять названия папок и файлов, чтобы убедиться в их правильности.