

Этот код и комментарии показывают, как можно работать с файлами и хранилищем данных в контексте использования Docker. Давайте разберем каждую строку:

Запуск контейнера
\\\

```
docker run -it -h GB --name gb-test ubuntu:22.04
```

- `docker run`: Запускает новый контейнер Docker.
- `-it`: Запускает контейнер в интерактивном режиме с подключенным терминалом (вы можете взаимодействовать с контейнером).
- `-h GB`: Устанавливает имя хоста внутри контейнера на "GB".
- `--name gb-test`: Дает контейнеру имя "gb-test".
- `ubuntu:22.04`: Использует образ Ubuntu версии 22.04.

Просмотр содержимого корневой директории
\\\

```
ls -l /
```

- `ls -l /`: Показывает список файлов и директорий в корневой директории `/`.

Создание директории
\\\

```
mkdir /example
```

- `mkdir /example`: Создает новую директорию `/example` в корневой директории.

Создание и редактирование файла
\\\

```
vi /example/passwords.txt
```

- `vi /example/passwords.txt`: Открывает (или создает) файл `/example/passwords.txt` с использованием текстового редактора `vi`.

Создание файла без редактора
\\\

```
touch /example/passwords.txt  
echo "123test" >> /example/passwords.txt
```

- `touch /example/passwords.txt`: Создает пустой файл `/example/passwords.txt`, если он не существует.
- `echo "123test" >> /example/passwords.txt`: Добавляет строку "123test" в файл `/example/passwords.txt`.

Перезапуск контейнера
\\\

```
docker start gb-test  
docker exec -it gb-test bash  
cat /example/passwords.txt
```

- `docker start gb-test`: Запускает остановленный контейнер с именем "gb-test".

- ``docker exec -it gb-test bash``: Запускает новую командную оболочку (``bash``) внутри работающего контейнера.
- ``cat /example/passwords.txt``: Выводит содержимое файла `/example/passwords.txt``.

Удаление и создание контейнера
```

```
docker stop gb-test
docker rm gb-test
```
```

- ``docker stop gb-test``: Останавливает контейнер "gb-test".
- ``docker rm gb-test``: Удаляет контейнер "gb-test".

Примонтирование внешнего хранилища
```

```
mkdir /test/folder
docker run -it -h GB --name gb-test -v /test/folder:/otherway
ubuntu:22.04
```
```

- ``mkdir /test/folder``: Создает директорию ``/test/folder`` на хост-машине.
- ``-v /test/folder:/otherway``: Примонтирует эту директорию к пути ``/otherway`` внутри контейнера.

Добавление данных в примонтированную директорию
```

```
echo "$HOSTNAME" >> /otherway.test.txt
```
```

- ``echo "$HOSTNAME" >> /otherway.test.txt``: Добавляет значение переменной окружения ``$HOSTNAME`` в файл ``/otherway.test.txt`` внутри контейнера.

Просмотр данных на хост-машине
```

```
cat /test/folder/test.txt
```
```

- ``cat /test/folder/test.txt``: Выводит содержимое файла ``test.txt``, который находится в примонтированной директории на хост-машине.

Этот пример показывает, как важно управлять данными в контейнерах. В идеале, никакие важные данные не должны храниться внутри контейнера: вместо этого их следует хранить в примонтированных томах или директориях.