

Проверьте, какие storage и log драйвера используются в вашем Docker.

Куплено
благодаря
Skladchik.com

Для этого воспользуйтесь командой:

```
docker info
```

Скопируйте строчки Storage Driver и Logging Driver и отправьте их в качестве ответа на данный шаг.

Самостоятельная работа

Docker демон умеет работать внутри Docker контейнера. Это называется Docker-in-Docker (dind). Давайте попробуем запустить тестовый стенд и ближе познакомиться с работой Docker демона.

1. Запустите контейнер из образа `docker:18.09.8-dind`, с именем `docker` и с привилегированным режимом (`--privileged`).
2. Проверьте, что `docker ps` показывает запущенный контейнер.
3. Запустите еще один контейнер из образа `docker:latest`, с интерактивным режимом, командой `sh`, с линком на первый контейнер (`--link docker:docker`) и с переменной окружения `DOCKER_HOST=docker:2375`. Дальнейшие действия будут выполняться из этого контейнера.

`--link` позволяет связать несколько контейнеров по сети, таким образом из контейнера из шага 3 можно будет обращаться в контейнер из шага 1 по dns имени `docker` и порту 2375.

4. Проверьте, что внутри контейнера из шага 3 при выполнении команды `docker ps` вы больше не видите контейнер с Docker демоном. Также можно проверить, что нет ни одного image'a.

5. pull image `busybox:latest`

```
docker pull busybox:latest
```

6. Попробуйте обращение в API Docker демона напрямую. Например, так:

```
wget docker:2375/images/json -O - -q
```

7. Скопируйте json ответ демона с image'ем `busybox:latest` и отправьте как решение в данном шаге.

Теперь вы умеете запускать `dind`, что довольно часто используется в различных CI pipeline, и можете общаться с Docker демоном без помощи утилиты `docker`.

+2