

Если вы видите что-то необычное, просто сообщите мне.

# Docker - инструмент создания инфраструктуры.

Docker (Докер) — программное обеспечение с открытым исходным кодом, применяемое для разработки, тестирования, доставки и запуска веб-приложений в средах с поддержкой контейнеризации. Он нужен для более эффективного использования системы и ресурсов, быстрого развертывания готовых программных продуктов, а также для их масштабирования и переноса в другие среды с гарантированным сохранением стабильной работы.

## Преимущества использования Docker

- Минимальное потребление ресурсов
- Скоростное развертывание
- Удобное скрывание процессов
- Работа с небезопасным кодом
- Простое масштабирование
- Удобный запуск
- Оптимизация файловой системы

## Компоненты Docker

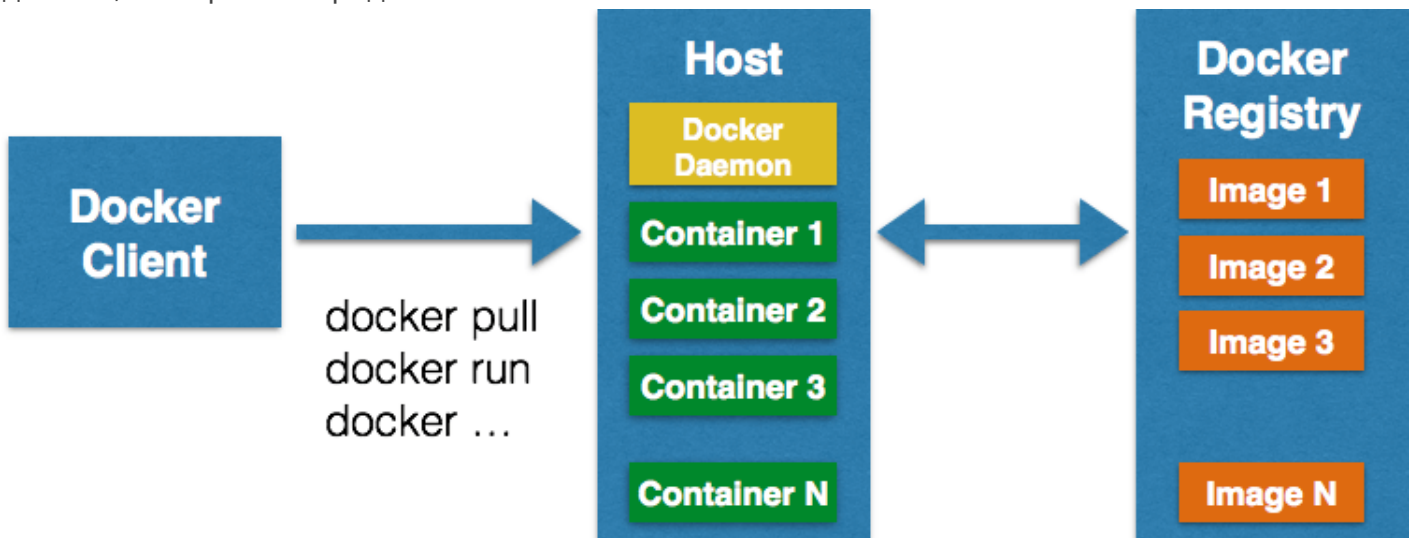
- Docker-демон (Docker-daemon) — сервер контейнеров, входящий в состав программных средств Docker. Демон управляет Docker-объектами (сети, хранилища,

образы и контейнеры). Демон также может связываться с другими демонами для управления сервисами Docker.

- Docker-клиент (Docker-client / CLI) — интерфейс взаимодействия пользователя с Docker-демоном. Клиент и Демон — важнейшие компоненты «движка» Докера (Docker Engine). Клиент Docker может взаимодействовать с несколькими демонами.
- Docker-образ (Docker-image) — файл, включающий зависимости, сведения, конфигурацию для дальнейшего развертывания и инициализации контейнера.
- Docker-файл (Dockerfile) — описание правил по сборке образа, в котором первая строка указывает на базовый образ. Последующие команды выполняют копирование файлов и установку программ для создания определенной среды для разработки.
- Docker-контейнер (Docker-container) — это легкий, автономный исполняемый пакет программного обеспечения, который включает в себя все необходимое для запуска приложения: код, среду выполнения, системные инструменты, системные библиотеки и настройки.
- Docker-volume — эмуляция файловой системы для осуществления операций чтения и записи. Она создается автоматически с контейнером, поскольку некоторые приложения осуществляют сохранение данных.
- Реестр (Docker-registry) — зарезервированный сервер, используемый для хранения docker-образов. Примеры реестров:
- Docker-хост (Docker-host) — машинная среда для запуска контейнеров с программным обеспечением.
- Docker-сети (Docker-networks) — применяются для организации сетевого интерфейса между приложениями, развернутыми в контейнерах.

## Как работает Docker

Работа Docker основана на принципах клиент-серверной архитектуры, которая основана на взаимодействии клиента с веб-сервером (хостом). Первый отправляет запросы на получение данных, а второй их предоставляет.



## Схема работы

1. Пользователь отдает команду с помощью клиентского интерфейса Docker-демону, развернутому на Docker-хосте. Например, скачать готовый образ из реестра (хранилища Docker-образов) с помощью команды `docker pull`. Взаимодействие между клиентом и демоном обеспечивает REST API. Демон может использовать публичный (Docker Hub) или частный реестры.
2. Исходя из команды, заданной клиентом, демон выполняет различные операции с образами на основе инструкций, прописанных в файле `Dockerfile`. Например, производит их автоматическую сборку с помощью команды `docker build`.
3. Работа образа в контейнере. Например, запуск `docker-image`, посредством команды `docker run` или удаление контейнера через команду `docker kill`.

## Примеры применения

- Быстрая доставка приложений (команды `docker pull` и `docker push`) позволяет организовать коллективную работу над проектом. Разработчики могут работать удаленно на локальных компьютерах и выполнять пересылку фрагментов кода в контейнер для тестов.

- Развертывание и масштабирование — контейнеры работоспособны на локальных компьютерах, серверах, в облачных онлайн-сервисах. Их можно загружать на хостинг для дальнейшего тестирования, создавать (`docker run`), останавливать (`docker stop`), запускать (`docker start`), приостанавливать и возобновлять (`docker pause` и `docker unpause` соответственно).
- Множественные нагрузки — осуществление запуска большого количества контейнеров на одном и том же оборудовании, поскольку Docker занимает небольшой объем дисковой памяти.
- Диспетчер процессов — возможность мониторинга процессов в Docker посредством команд `docker ps` и `docker top`, имеющими схожий синтаксис с Linux.
- Удобный поиск — в реестрах Docker он осуществляется очень просто. Для этого следует использовать команду `docker search`.

## Установка докера

```
#!/bin/bash
apt update
apt install -y docker.io
groupadd docker
usermod -aG docker $USER
```

Две последние команды необходимы для того, чтобы текущий пользователь мог управлять контейнерами, не запрашивая повышенных привелегий.

## Управлением демоном

Запуск демона:

```
systemctl start docker
```

Остановка демона:

```
systemctl stop docker
```

Включение/отключение автозагрузки демона:

```
systemctl enable/disable docker
```

При добавлении ключа `--now` - выполняет команду не дожидаясь перезагрузки. В противном случае, команда выполнится только после перезагрузки.

---

Revision #9

Created 2021-09-22 11:53:02 UTC by gasick

Updated 2023-11-08 16:10:47 UTC by ivan.serov37@gmail.com