

# ОС UNIX, лекция 11: введение в контейнерную виртуализацию на примере Docker

Пожидаев М. С.

21 октября 2020 г.

Основные отличия контейнерной виртуализации:

1. Код выполняется нативно.
2. В гостевых системах ядро общее с хост-системой.

# Минусы контейнерной виртуализации

При сплошных плюсах технологии контейнерной виртуализации есть и некоторые минусы:

1. Слишком быстро копится overhead использования дискового пространства.
2. Сильная привязка к ядру хост-системы.
3. На Windows (равно как и в других ОС вообще) выполняется в виртуальной машине, что сводит на нет полезный эффект.
4. Не эмулирует недоступное оборудование.

1. Пространства имён (namespace).
2. Контрольные группы(cgroups).

Одна копия Linux может разделять на пространства имён следующие подсистемы:

- ▶ файловая система;
- ▶ идентификаторы процессов;
- ▶ сетевые атрибуты;
- ▶ элементы межпроцессного взаимодействия;
- ▶ идентификаторы пользователей.

Контрольные группы — Механизм ядра Linux, позволяющий лимитировать интенсивность использования таких ресурсов как:

- ▶ память;
- ▶ сетевой обмен;
- ▶ ввод/вывод;
- ▶ процессорное время.

Docker — система управления контейнерами на базе пространств имён и контрольных групп Linux.

1. Автоматически конфигурирует сеть для гостевых систем и хост-системы.
2. Использует каскадное объединение томов хранения файлов.
3. Поддерживает публичный репозиторий образов контейнеров.
4. Написана на языке Go.

Docker Hub — публичный репозиторий образов Docker для построения контейнеров. Имеет платные возможности для размещения частных образов и поддерживает автоматизированную сборку на основе репозитория в GitHub.

*Browse over 100,000 container images from software vendors, open-source projects, and the community.*



- ▶ *Image*: заготовка, на основе которого создаётся серия однотипных контейнеров.
- ▶ *Network*: группа адресов для сетевых соединений между контейнерами.
- ▶ *Volume*: файловая система для хранения данных, в том числе разделяемых между контейнерами.

## Пример Dockerfile

```
FROM ubuntu:latest
WORKDIR /app
COPY build /app/
RUN apt-get update
RUN DEBIAN_FRONTEND=noninteractive apt-get install -y \
openjdk-11-jre git g++ cmake curl bison
RUN git clone https://github.com/foo/bar/
RUN /app/build
```

Когда контейнеров становится много, то управление ими требует применение дополнительных инструментов. Существует ряд утилит, которые решают задачу так называемой оркестрации контейнеров: запуск необходимого количества контейнеров и поддержания их работоспособности. Среди них наиболее популярны:

- ▶ Kubernetes;
- ▶ Docker Swarm.

Спасибо за внимание!

Веб-сайт: <http://marigostra.ru/>

E-mail: [mSP@luwrain.org](mailto:mSP@luwrain.org)