

Контейнеризация

SKILLFACTORY



Содержание

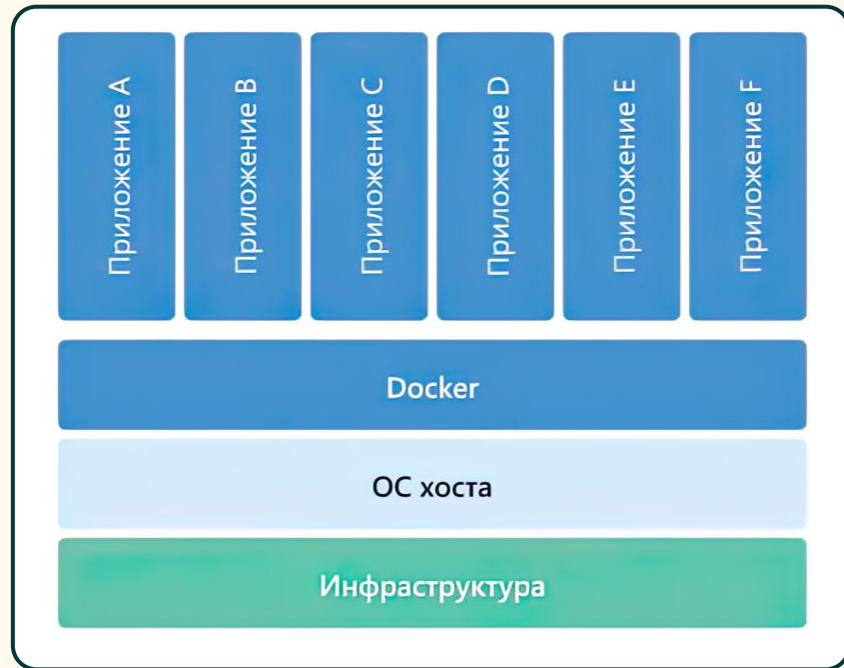
- Контейнеры — LXD vs Docker
- Архитектура — Image, Volume, Registry, Network
- Extra — Dockerfile, docker-compose, команды

Контейнеризация

Изолированное пространство процессов внутри операционной системы.

Использует функции ядра «Пространство имён» и «Контрольные группы».

Уровень изоляции: Ядро операционной системы.



LXD vs Docker

LXD

- Системный контейнер.
- Сохраняет состояние.
- Запуск полной операционной системы.
- Подходит для использования вместо VM.

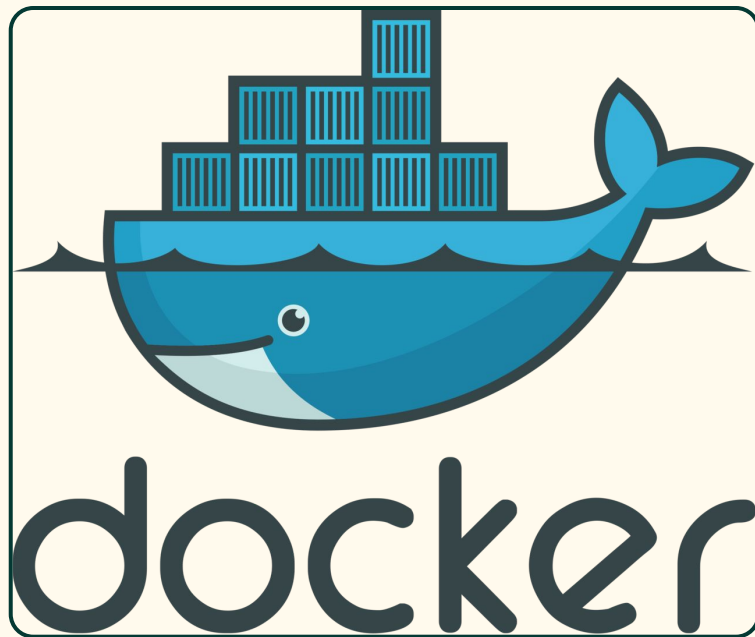
Docker

- Контейнер процесса.
- Не сохраняет состояние.
- Запуск только приложения/процесса.
- Подходит для реализации мультисервисного приложения.

Docker

Возможности:

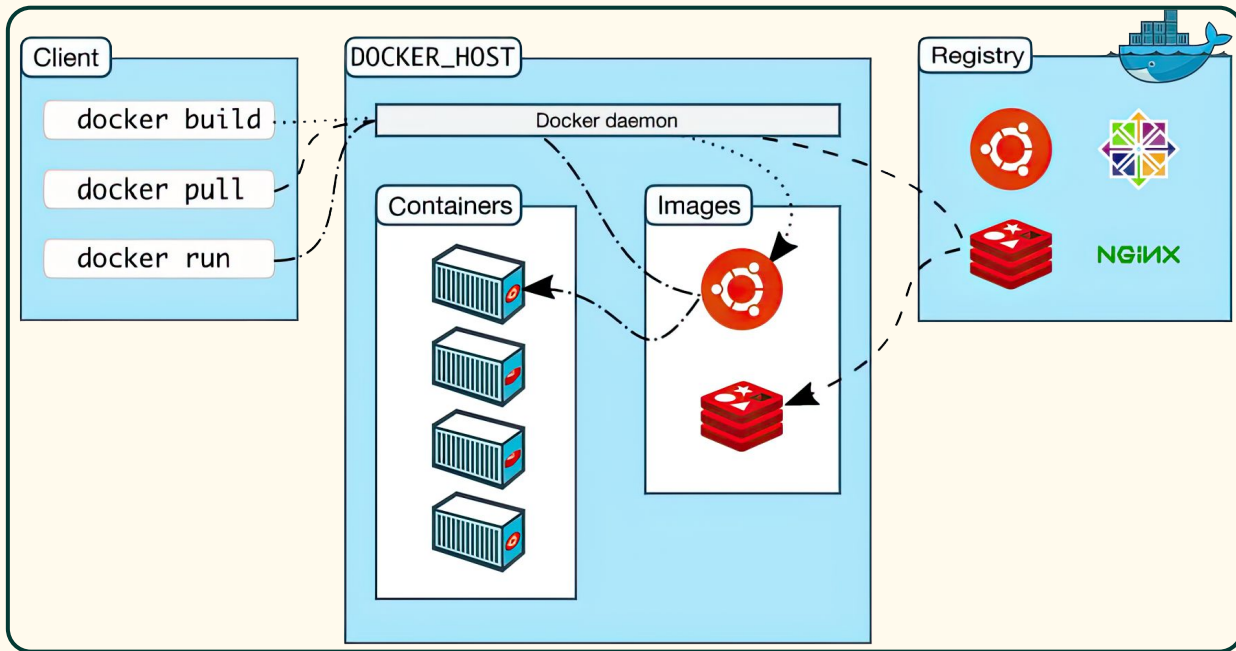
- запуск контейнера из образа;
- свободное распространение контейнеров с помощью dockerhub;
- миграция;
- развертывание нескольких приложений с использованием нескольких контейнеров;
- собственная реализация демона containerd;
- управление через DockerAPI и CLI.



Содержание

- Контейнеры — LXD vs Docker
- Архитектура — Image, Volume, Registry, Network
- Extra — Dockerfile, docker-compose, команды

Архитектура



Образ (Image)

Образ – снимок изолированной среды.

Образы Docker неизменны и хранятся в репозитории (локальный, публичный и приватный).

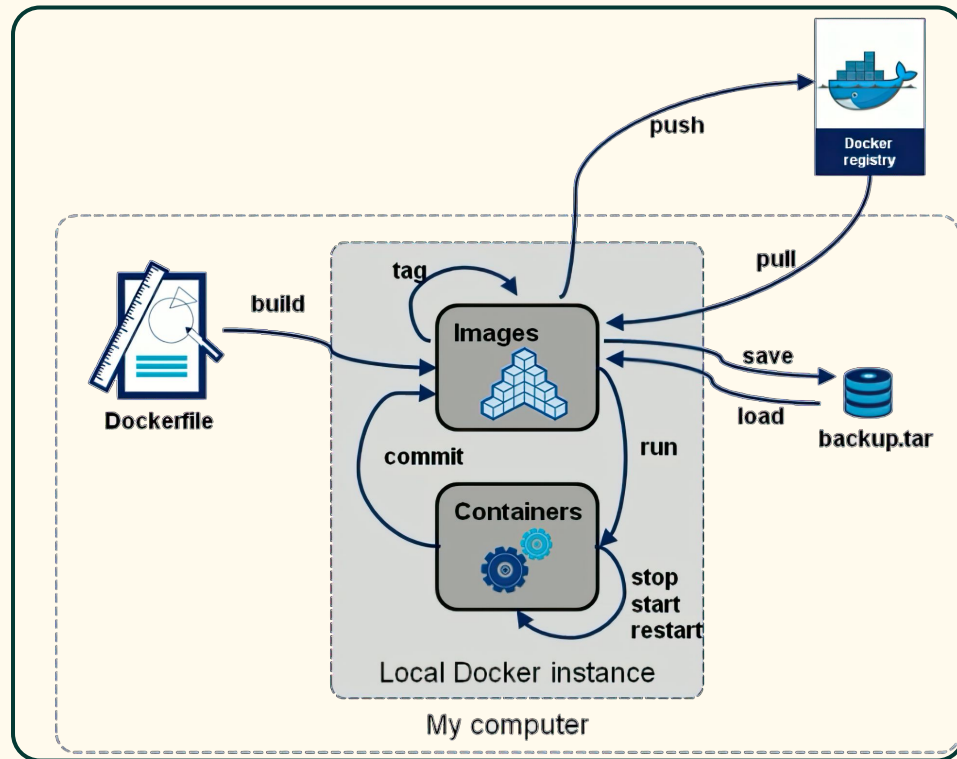
Структура образа:

- базовый образ;
- слои;
- слой контейнера;
- манифест (метаданные образа в формате json).

Реестр(Registry)

Реестр – высокомасштабируемое серверное приложение для хранения и распространения образов Docker.

Готовые образы: [Docker hub](https://hub.docker.com/).

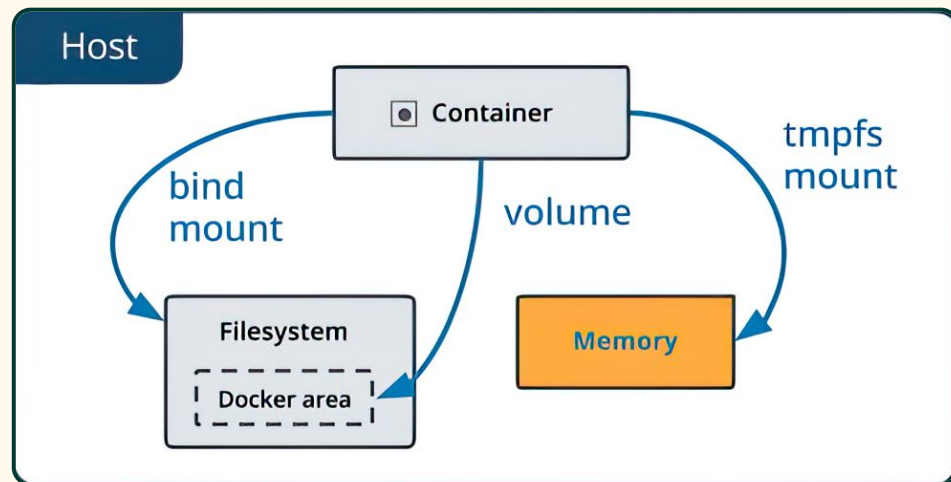


Volume

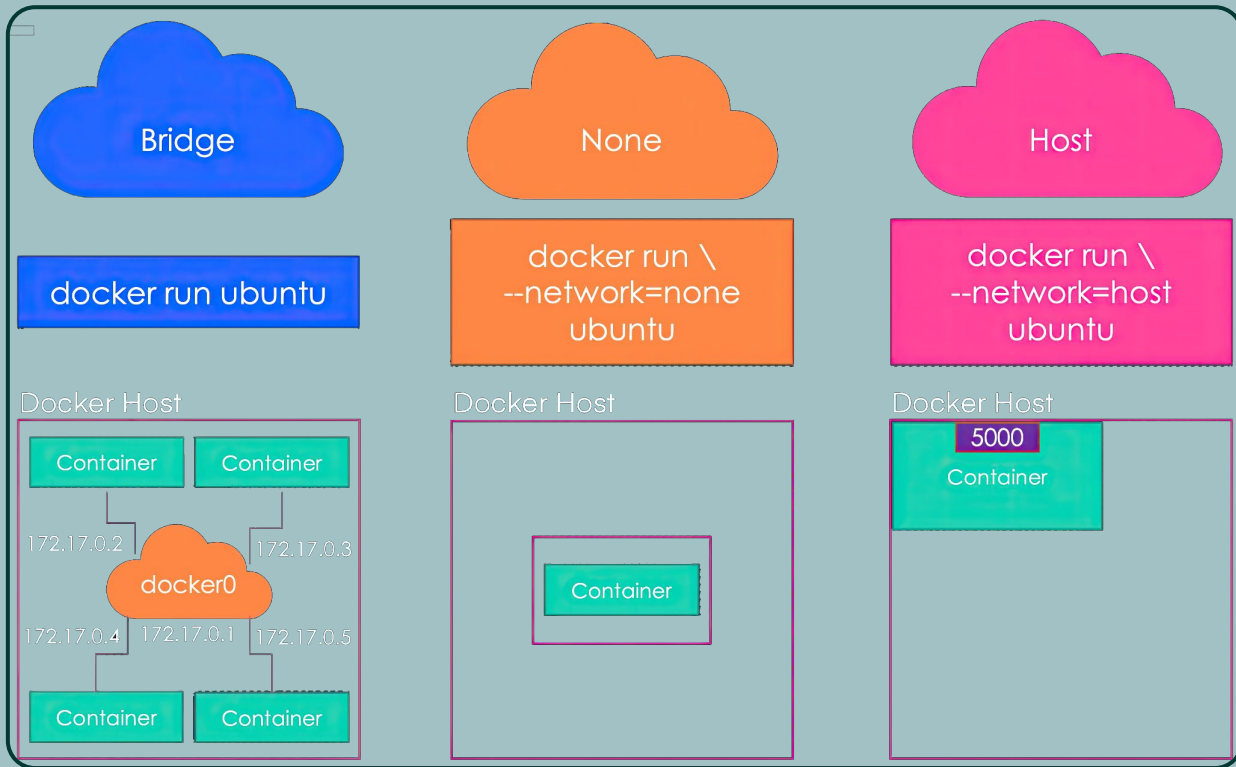
Том (Volume) – изолированная область постоянной памяти для хранения данных.

Альтернатива:

- монтирование каталогов;
- временная память (tmpfs).



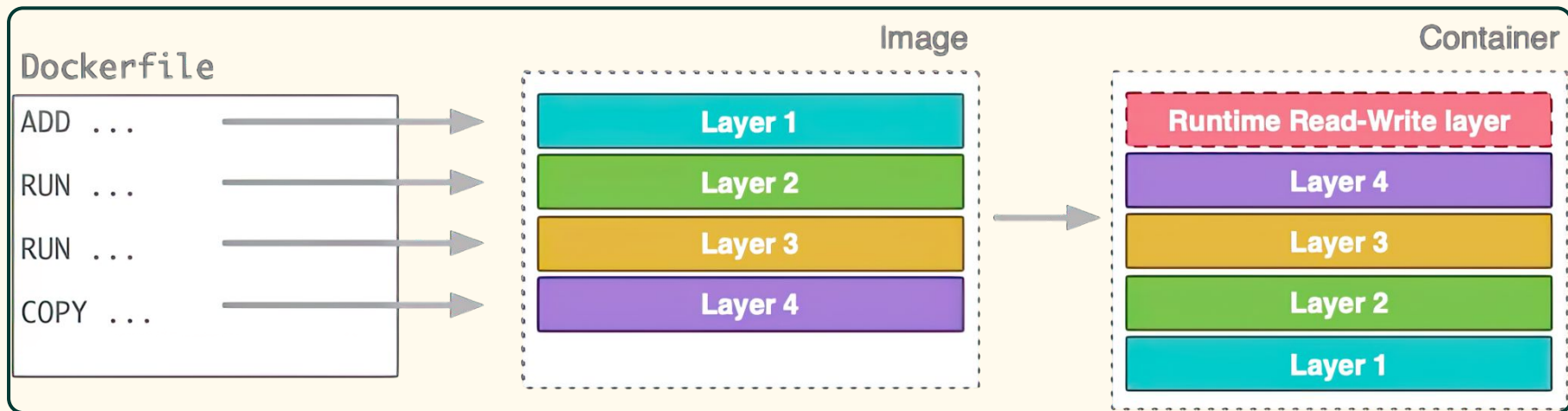
Networking



Содержание

- Контейнеры — LXD vs Docker
- Архитектура — Image, Volume, Registry, Network
- Extra — Dockerfile, docker-compose, команды

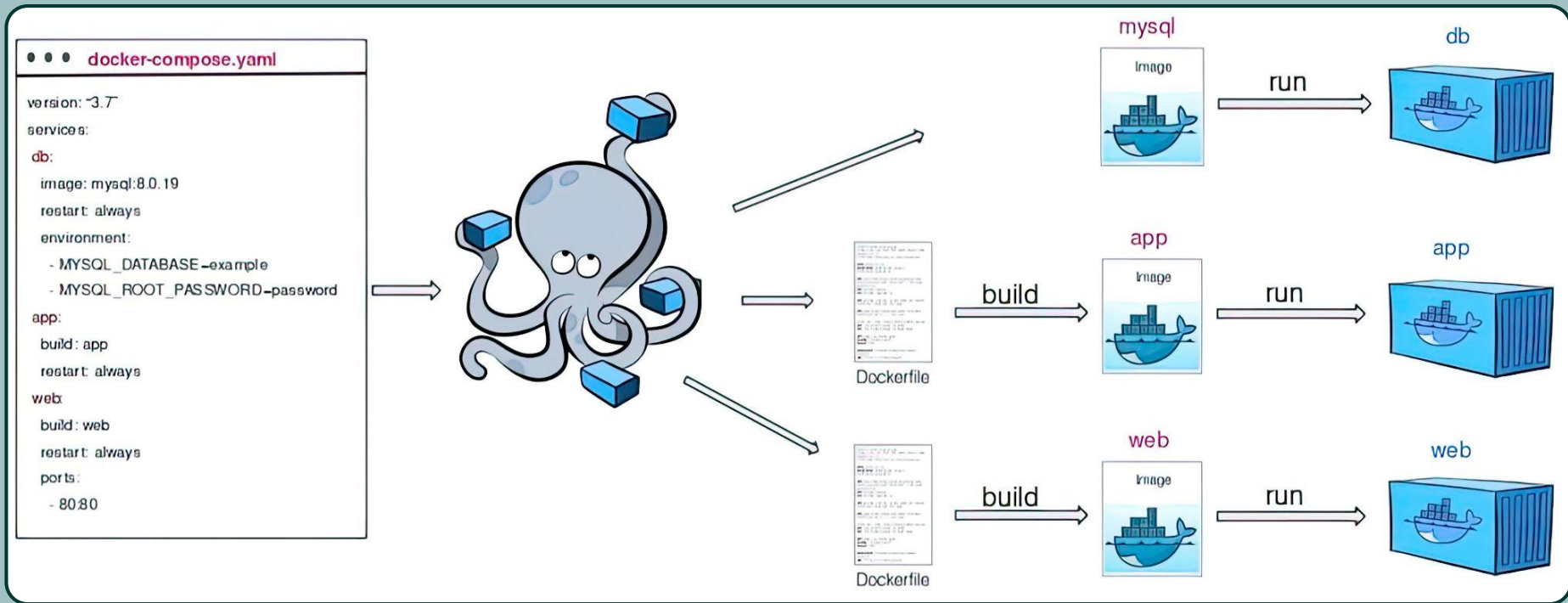
Dockerfile



Dockerfile - директивы

- FROM – инициализация образа.
- RUN – запуск команды в контейнере.
- ADD – копирование файлов в контейнер (url, tar).
- ENTRYPOINT – утилита по умолчанию.
- ENV – задание переменных окружения.
- LABEL – задание метаданных контейнера.
- EXPOSE – открытие портов.
- CMD – запуск команды по умолчанию в контейнере.
- USER – пользователь по умолчанию.
- VOLUME – создание точки монтирования в контейнере.
- COPY – копирование файлов в контейнер.
- WORKDIR- рабочая директория.

docker-compose.yml



Команды - docker

- **docker build** - сборка контейнера;
- **docker cp** – копирование файлов между контейнером и хостом;
- **docker exec** – запустить команду в работающем контейнере;
- **docker export** – экспортировать файловую систему контейнера;
- **docker import** – импортировать файловую систему контейнера;
- **docker history** – показывать историю образа;
- **docker image** – управление образами;
- **docker inspect** – получение информации об объектах docker;
- **docker kill** – остановка одного или более контейнеров;
- **docker ps** – список работающих контейнеров;
- **docker run** – запустить команду в новом контейнере.

Команды - docker compose

docker compose:

- **build** - сборка контейнера;
- **exec** – запустить команду в работающем контейнере;
- **up** – создать и запустить контейнеры;
- **down** – остановить и удалить контейнеры;
- **start** – запустить сервисы;
- **stop** – остановить сервисы;
- **restart** – перезапустить сервисы;
- **port** – вывести таблицу портов;
- **kill** – остановка контейнеров;
- **rm** – удалить неработающие контейнеры;
- **images** – список образов, используемых созданными контейнерами;
- **ps** – список работающих контейнеров;
- **run** – запустить команду.

Ссылки

- [Введение в Docker](#)
- [Документация докер](#)
- [Создание оптимальных образов](#)
- [LXD Ubuntu](#)
- [Создание приватного репозитория](#)
- [Команды докер](#)
- [Команды docker compose](#)
- [Dockerfile](#)
- [Команды docker-compose.yml](#)
- [Лучшие практики Dockerfile](#)