

Основы Docker

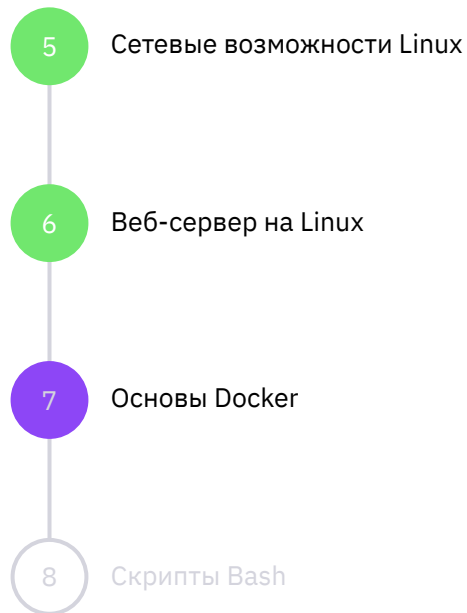
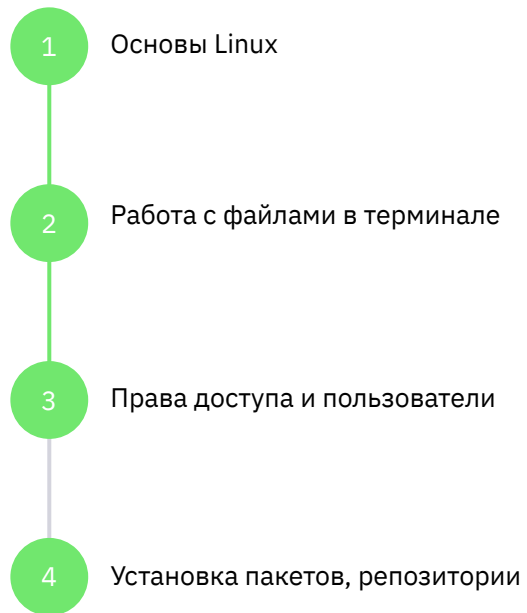




Содержание урока








План курса





Что будет на уроке сегодня

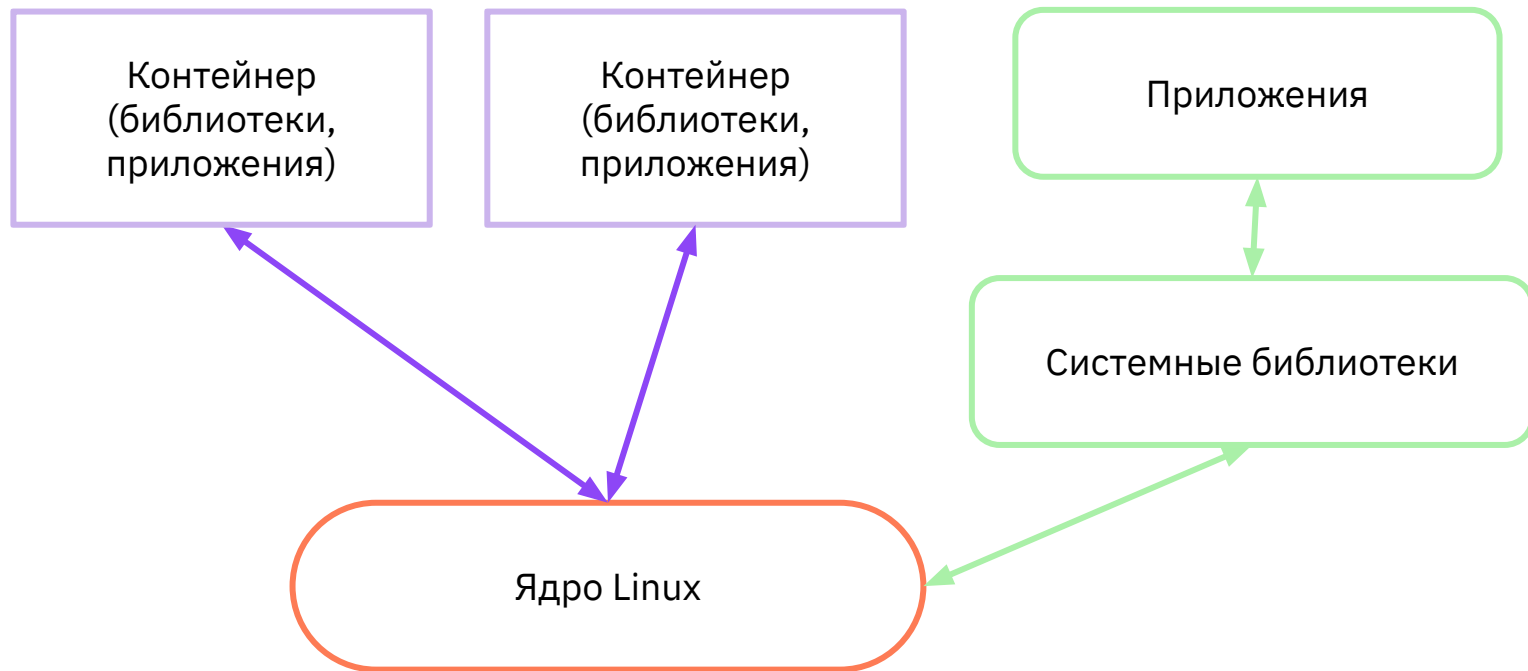
-  Разберём основы технологии контейнеризации
-  Изучим инфраструктуру Docker
-  Разберём схему команды управления контейнерами
-  Научимся запускать контейнеры из образов
-  Познакомимся с запуском систем контейнеров



Контейнеризация

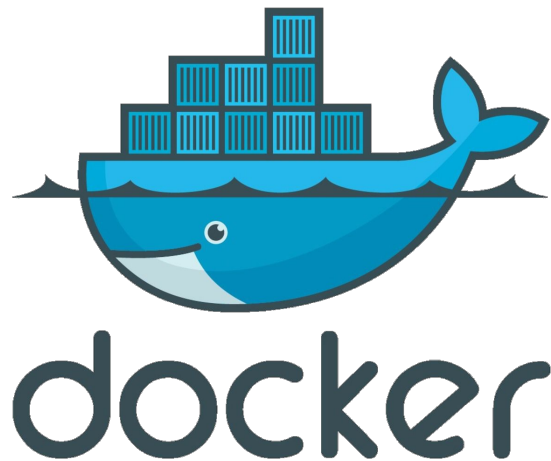


Архитектура контейнеров



Контейнеризация

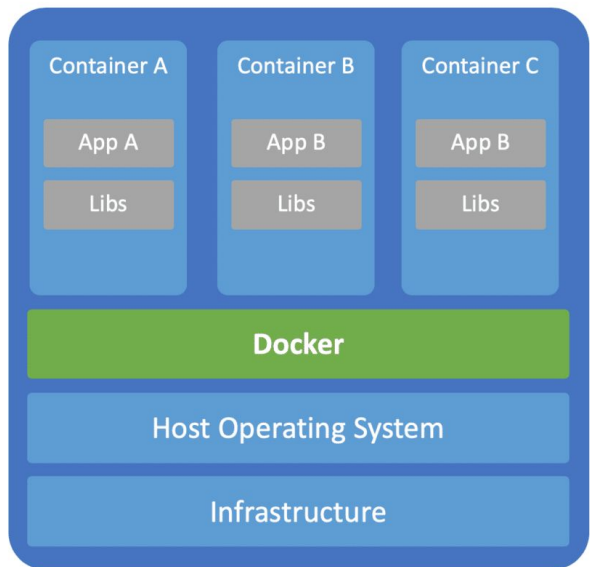
- Быстрое развёртывание приложений из образов
- Полная повторяемость при запуске (надёжность)
- Высокая скорость запуска и удаления контейнеров
- Низкие накладные расходы на виртуализацию
- Возможность преодоления несовместимости софта и ОС
- Автоматизация операций управления приложениями



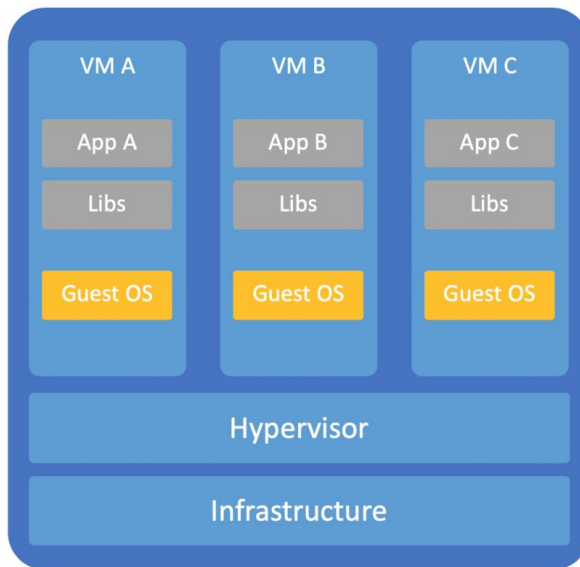


Виртуализация и контейнеризация

Container

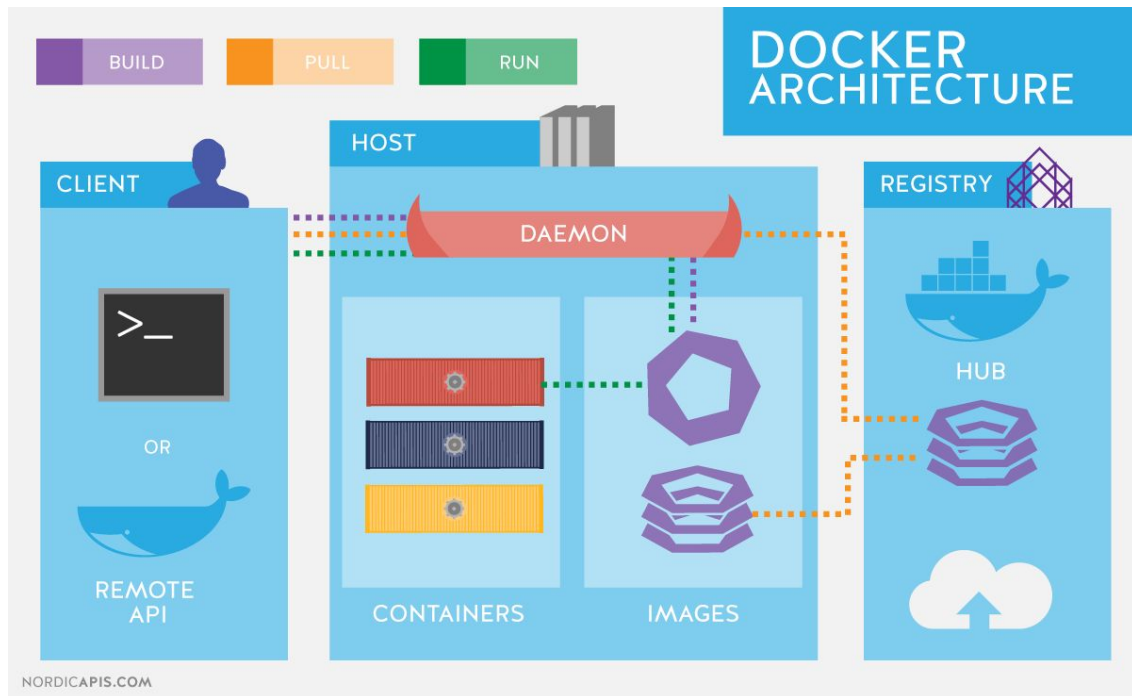


Virtual Machines





Архитектура Docker





Реестр образов Docker Hub

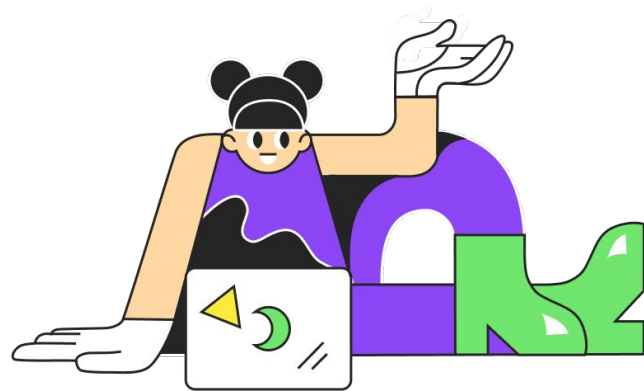
- Реестр по умолчанию в Docker
- Веб-интерфейс: <https://hub.docker.com/>
- Официальные образы (official image)
- Теги (latest, 0.4-alpine)
- Документация к образам





Запуск тестового контейнера

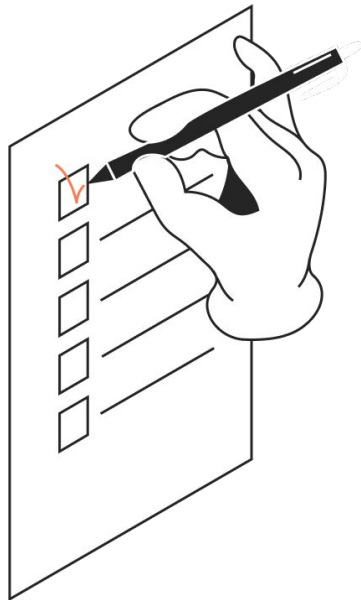
- Установка Docker: `apt install docker.io`
- Проверка: `sudo docker`
- Создание и запуск контейнера: `docker run hello-world`





Базовые операции

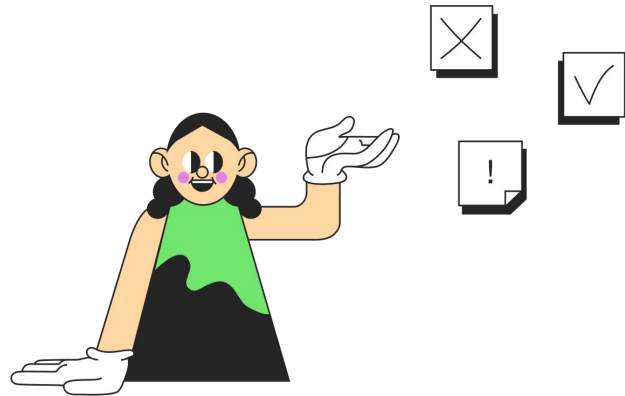
- `docker ps` – просмотр активных контейнеров
- `docker ps -a` – просмотр всех контейнеров
- `docker images` – список образов
- `docker search nginx` – поиск образа
- `docker pull nginx` – скачивание образа
- `docker start|restart|stop nginx` – операции с контейнером
- `docker rm 9cbf7c3230d0` – удаление контейнера
- `docker rmi hello-world` – удаление образа
- `docker logs nginx1` – просмотр логов контейнера





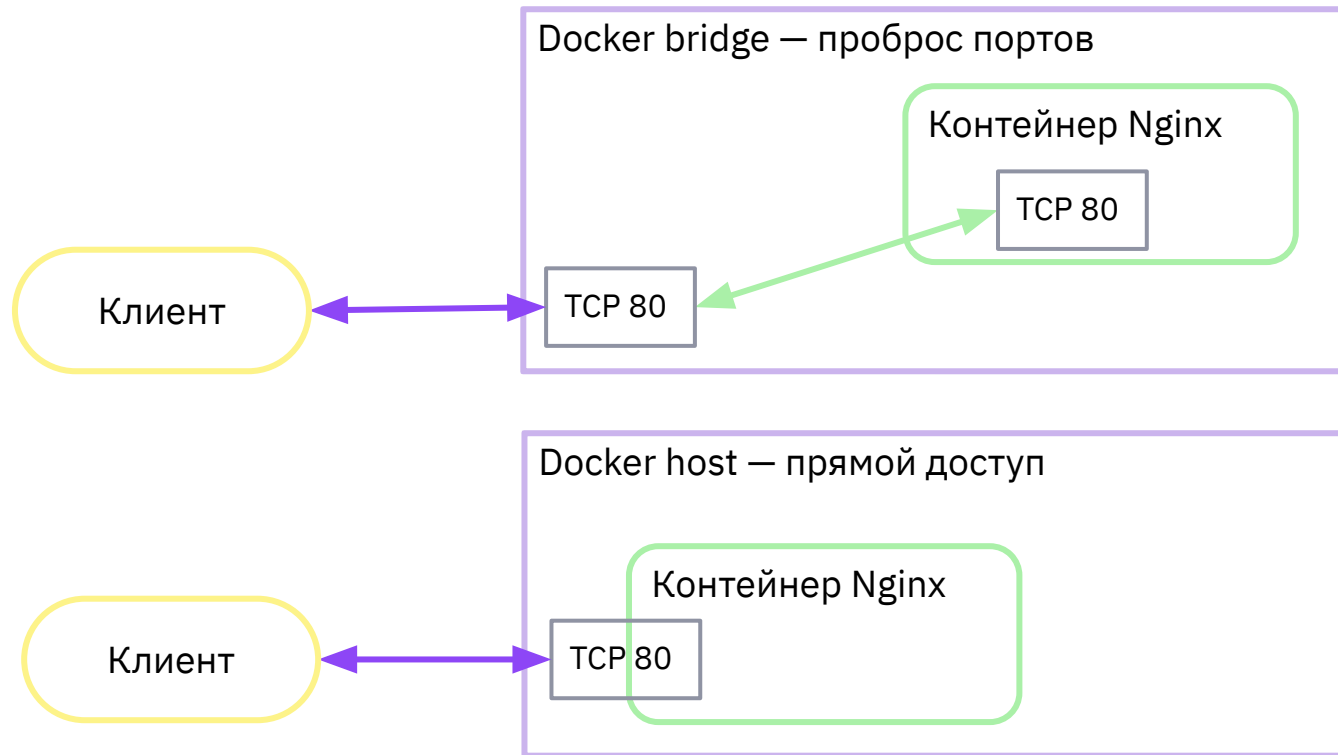
Типы сетей в Docker

- **Bridge** — сети по умолчанию, аналог типа подключения NAT в VirtualBox.
- **Host** — с помощью этого драйвера контейнер получает доступ к собственному интерфейсу хоста. Аналог подключения «Мост» в VirtualBox.
- **Macvlan** даёт контейнерам прямой доступ к интерфейсу и суб-интерфейсу (VLAN) хоста.
- **Overlay** позволяет строить сети на нескольких хостах с Docker.





Проброс портов и host-сеть в Docker



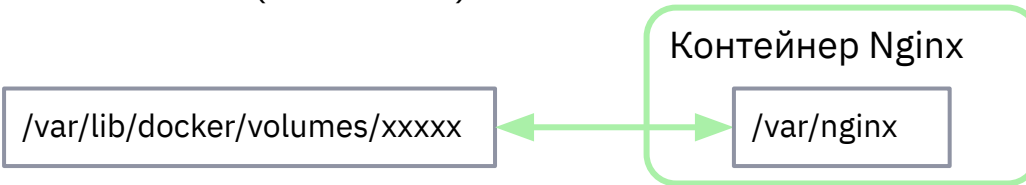


Томы и монтирование в Docker

Bind mount (монтирование)



Docker volume (том Docker)





Запуск Nginx в контейнере



Запуск Nginx – проброс портов и директории

- Команда создания и запуска: `docker run -d --restart always --name nginx1 \`
`-p 80:80 -v /var/www/html:/usr/share/nginx/html nginx`
- Учитываем порты на хосте (`ss -ntlp`)
- Монтируем директории
- Активируем автозапуск
- Контейнер работает в режиме демона
- Сеть по умолчанию (`bridge`)

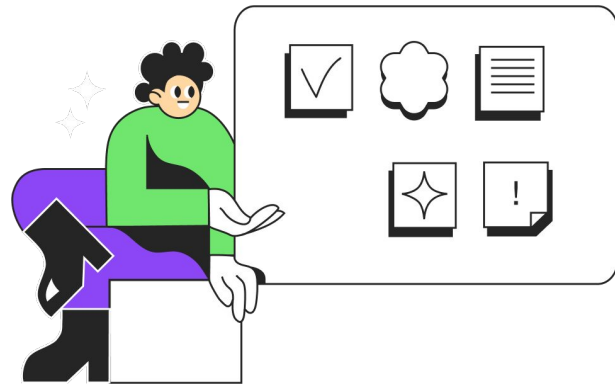


NGINX[®]
Part of F5



Работа внутри контейнера

- **Не рекомендуется в реальной работе**
- Заходим: `docker exec -ti nginx1 bash`
- Смотрим настройки: `ls -al /etc/`
- Версия базового дистрибутива: `cat /etc/os-release`
- Настройки nginx: `ls -al /etc/nginx/`
- Директория сайта: `ls -al /usr/share/nginx/html`



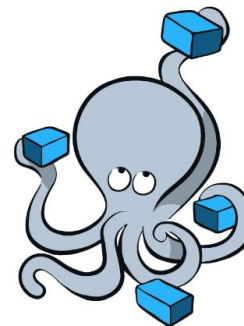


Docker Compose



Docker Compose

- Запуск нескольких контейнеров одной командой
- Декларативная конфигурация
- Формат YAML (отступы, вложенность)
- Автоматизация запуска систем контейнеров



docker
Compose



Синтаксис docker-compose.yml

```
version: '3'
```

```
services:
```

```
  wordpress:
```

```
    depends_on:
```

```
      - db
```

```
    image: wordpress:5.3.2-fpm-alpine
```

```
    container_name: wordpress
```

```
    environment:
```

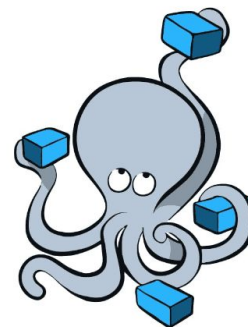
```
      - WORDPRESS_DB_HOST=db:3306
```

```
    volumes:
```

```
      - wordpress:/var/www/html
```

```
    networks:
```

```
      - app-network
```

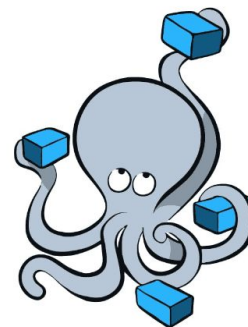


docker
Compose



Использование Docker Compose

- `apt install docker-compose` – установка
- Проверка yml: `apt install yamllint`
- `docker-compose up -d` – создание и запуск
- `docker-compose ps` – список контейнеров
- `docker-compose down` – остановить и удалить
- `docker-compose stop` – остановить
- `docker-compose start` – запустить
- `docker-compose rm` – удалить остановленные



docker
Compose



Итоги занятия



На этом уроке мы

- 📌 Познакомились понятием контейнеризации
- 📌 Узнали различия по сравнению с виртуализацией
- 📌 Изучили архитектуру Docker
- 📌 Запустили базовый контейнер
- 📌 Разобрали работу с сетью
- 📌 Создали полноценный контейнер с Nginx
- 📌 Познакомились с работой docker-compose





Спасибо за внимание!