

Введение в Docker

Linux

Вопросы по практической работе и предыдущему уроку

План урока

1. Установка docker
2. Обзор docker
3. Управление образами и контейнерами
4. Управление сетями в docker
5. Обзор docker-compose

Установка docker

Контейнеризация — метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя вместо одного.

Эти экземпляры с точки зрения пользователя полностью идентичны отдельному экземпляру операционной системы. Простыми словами, контейнеризация позволяет виртуализировать процесс.

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации

ТЕМА

Подключаем репозиторий

01

1. `apt-get install
apt-transport-https
ca-certificates curl gnupg-agent
software-properties-common -y`
2. `curl -fsSL
https://download.docker.com/li
nux/ubuntu/gpg | apt-key add -`
3. `add-apt-repository "deb
[arch=amd64]
https://download.docker.com/li
nux/ubuntu $(lsb_release -cs)
stable"`

ТЕМА

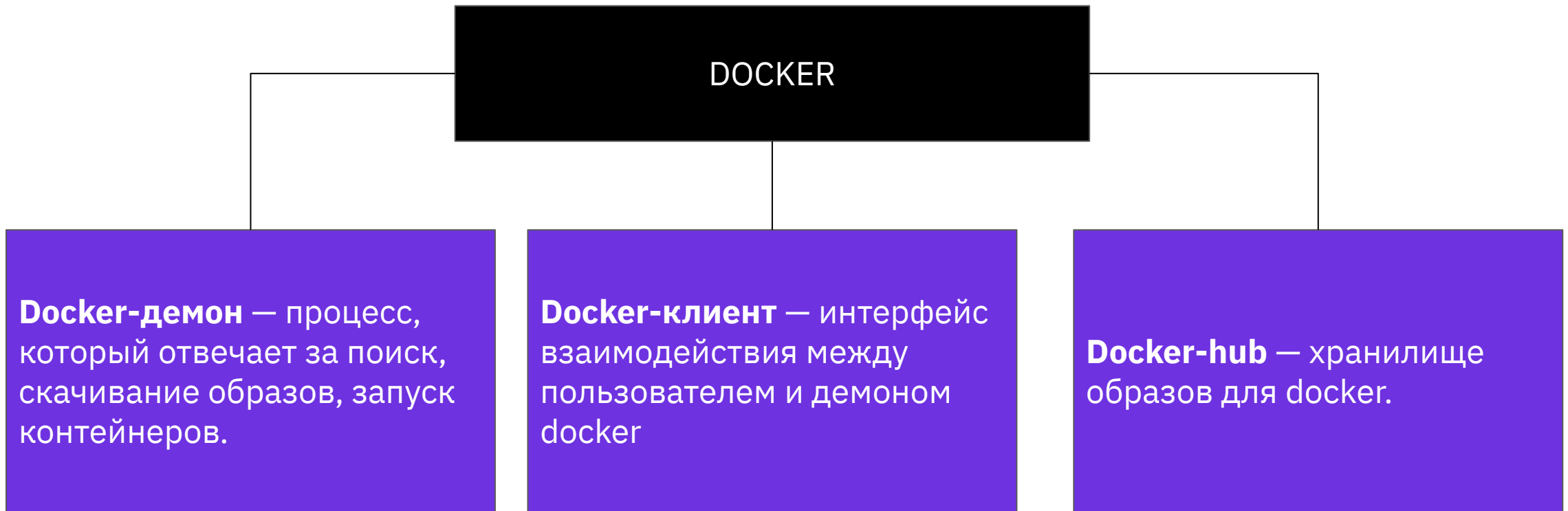
Установка

02

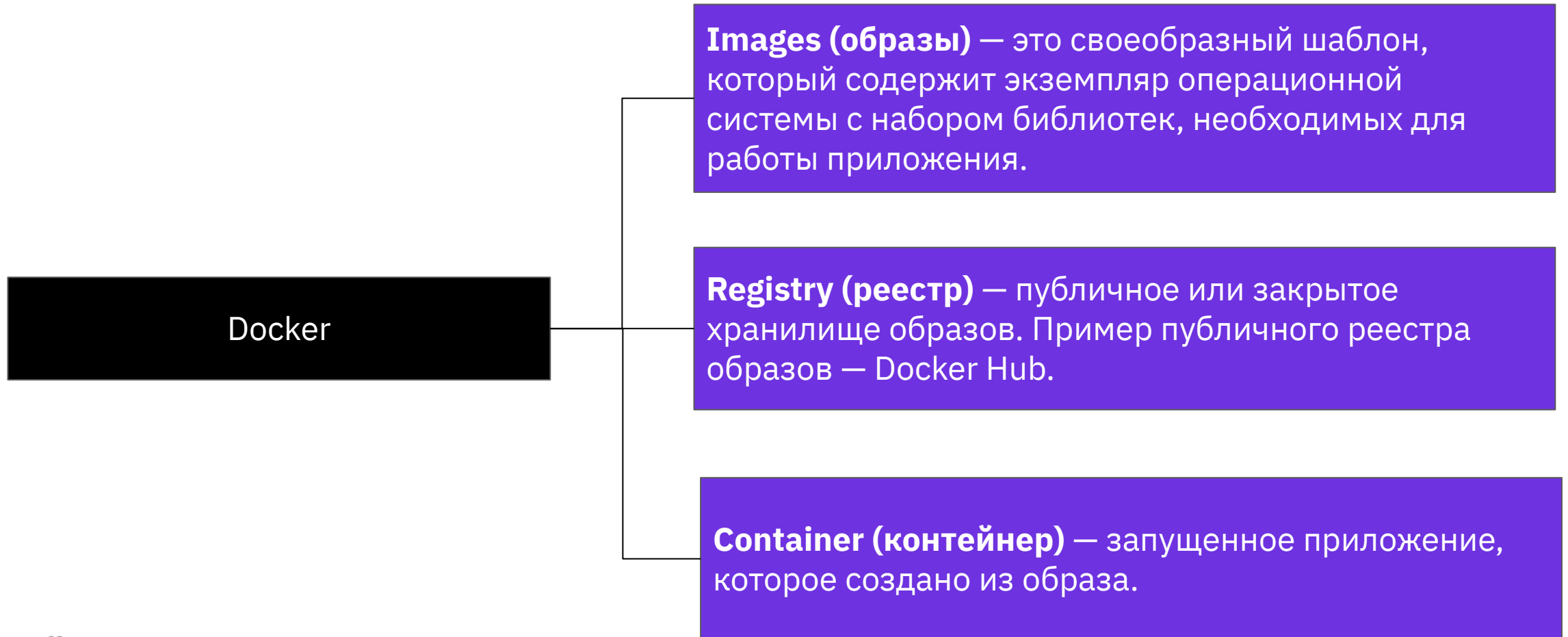
```
apt update; apt install docker-ce -y.
```


Обзор docker

Docker с точки зрения ОС



Docker с точки зрения архитектуры



Управление образами и контейнерами

Управление образами и контейнерами

**docker search
image_name —
поиск образа в
реестре.
Например, docker
search nginx
найдёт все
образы, которые
содержат веб-
сервис nginx.**

**docker pull
image_name
скачает диск из
реестра,
например, docker
pull nginx.**

**docker run --name
container_name
image_name
запустить
контейнер из
скачанного
образа**

**docker rm
container_name
удалит
контейнер**

Dockerfile

Dockerfile — сценарий, в котором будут описаны все шаги по сборке нашего приложения.

Структура Dockerfile:

FROM — определит базовый образ, из которого будет собираться контейнер.

MAINTAINER — сообщит контейнеру имя автора создаваемого образа.

RUN — запустит команду внутри образа.

ADD — берёт файлы с хоста и кладёт внутрь образа.

VOLUME — директория, которая будет подключена в контейнер.

EXPOSE — задаст порт, через который контейнер будет общаться с внешним миром.

CMD — команда, которая будет запущена при старте контейнера из образа.

Структура Dockerfile:

`docker build -t image_name .` - сборка образа

`docker run --name container_name image_name` -
запуск контейнера из созданного образа

Управление сетями в docker

Управление сетями в docker

Сеть Docker

Bridge — сети по умолчанию, аналог типа подключения NAT в VirtualBox. Связь устанавливается через Bridge-интерфейс, который поднимается в операционной системе при установке Docker и носит название docker0.

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.

Управление сетями в docker

1. `docker network ls` - просмотреть доступные сети
2. `docker network inspect network_name` - просмотреть участников сети
3. Доступ к приложениям, запущенным в контейнере, осуществляется через `iptables`.

Обзор docker-compose

Обзор **docker-compose**

docker-compose повторяет весь функционал Docker, за исключением одного: если Docker применяется для управления одним конкретным сервисом, то docker-compose позволяет управлять несколькими контейнерами, входящими в состав приложения.

Файл docker-compose.yml:

```
1 version: '3'
2 services:
3   nginx:
4     image: nginx:latest
5     ports:
6       - 80:80
7     volumes:
8       - /var/www/html
9
10
```

Обзор docker-compose

1. version '3' говорит об использовании третьей версии формата файлов для docker-compose.
2. Директива service описывает службу, которую мы будем запускать, дальше идёт имя nginx.
3. Собираем контейнер из последней стабильной версии nginx, доступной на Docker Hub: image: nginx:latest и пробрасываем 80-й порт хост-машины и каталог /var/www/html, используя директивы ports и volumes.

Обзор docker-compose

4. Обратите внимание, что файл `docker-compose.yml` для каждого контейнера должен лежать в отдельной папке.
5. Запускаем наш проект, используя команду `docker-compose up -d --build`, — говорим, что `docker-compose` должен запустить контейнер в оперативной памяти, выполнив сборку из образа.

Спасибо!
Каждый день
вы становитесь
лучше :)

