



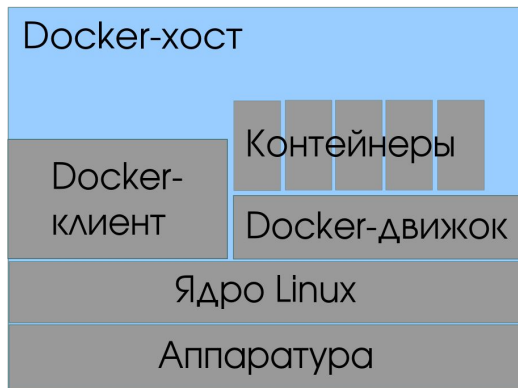
Docker

Tinkoff.ru



- Что такое docker
- Docker vs KVM
- Что docker мог бы заменить, но...

# Что такое docker



Программное обеспечение функционирует в среде Linux с ядром, поддерживающим cgroups и изоляцию пространств имён (namespaces).

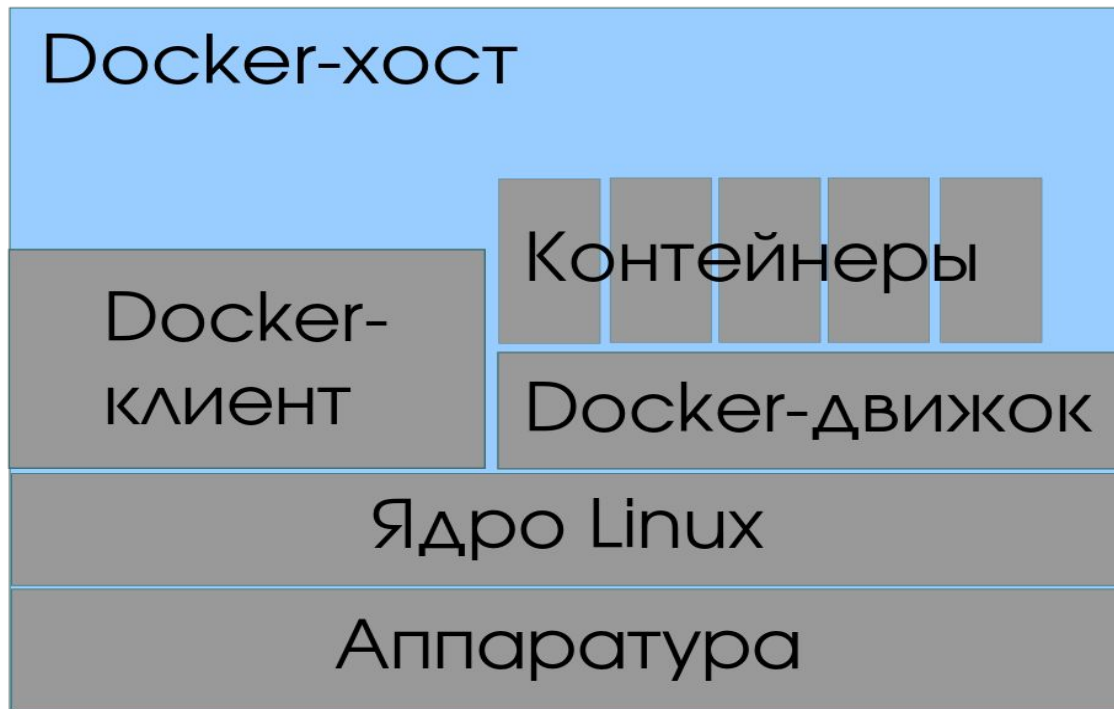
Для экономии дискового пространства проект использует файловую систему с поддержкой технологии каскадно-объединённого монтирования: контейнеры используют образ базовой операционной системы, а изменения записываются в отдельную область.

В состав программных средств входит демон — сервер контейнеров (запускается командой `docker -d`), клиентские средства, позволяющие из интерфейса командной строки управлять образами и контейнерами, а также API, позволяющий в стиле REST управлять контейнерами программно.

Демон обеспечивает полную изоляцию запускаемых на узле контейнеров на уровне файловой системы (у каждого контейнера собственная корневая файловая система), на уровне процессов (процессы имеют доступ только к собственной файловой системе контейнера, а ресурсы разделены средствами `libcontainer`), на уровне сети (каждый контейнер имеет доступ только к привязанному к нему сетевому пространству имён и соответствующим виртуальным сетевым интерфейсам).

Набор клиентских средств позволяет запускать процессы в новых контейнерах (`docker run`), останавливать и запускать контейнеры (`docker stop` и `docker start`), приостанавливать и возобновлять процессы в контейнерах (`docker pause` и `docker unpause`). Серия команд позволяет осуществлять мониторинг запущенных процессов (`docker ps` по аналогии с `ps` в Unix-системах, `docker top` по аналогии с `top` и другие). Новые образы возможно создавать из специального сценарного файла (`docker build`, файл сценария носит название `Dockerfile`), возможно записать все изменения, сделанные в контейнере в новый образ (`docker commit`). Все команды могут работать как с `docker`-демоном локальной системы, так и с любым сервером Docker, доступным по сети. Кроме того, в интерфейсе командной строки встроены возможности по взаимодействию с публичным репозиторием Docker Hub, в котором размещены предварительно собранные образы контейнеров, например, команда `docker search` позволяет осуществить поиск образов среди размещённых в нём, образы можно скачивать в локальную систему (`docker pull`), возможно также отправить локально собранные образы в Docker Hub (`docker push`).

# Архитектура docker





- `docker cli` - консольная утилита для управления образами и контейнерами
- `docker daemon` - процесс на хост машине и умеющий скачивать и заливать образы, запускать из них контейнеры, следить за запущенными контейнерами, собирать логи и настраивать сеть между контейнерами. Именно демон создает образы контейнеров, хоть и может показаться, что это делает `docker-client`.
- `docker hub` - хранилище образов. Аналогично `github`, но для `docker` образов.
- `docker-compose` - консольная утилита для работы с `docker-compose.yml`. В `docker-compose.yml` как и `Dockerfile`, используется декларативный подход, что позволяет определить весь технологический стек, включая требования к сети и хранению. В нём описываются сервисы запускаемые с помощью `docker`.

KVM (Kernel-based Virtual Machine) — технология аппаратной виртуализации, позволяющая создать на хост-машине полный виртуальный аналог физического сервера. KVM позволяет создать полностью изолированный от «соседей» виртуальный сервер с собственным ядром ОС, который пользователь может настраивать и модифицировать под собственные нужды без ограничений.

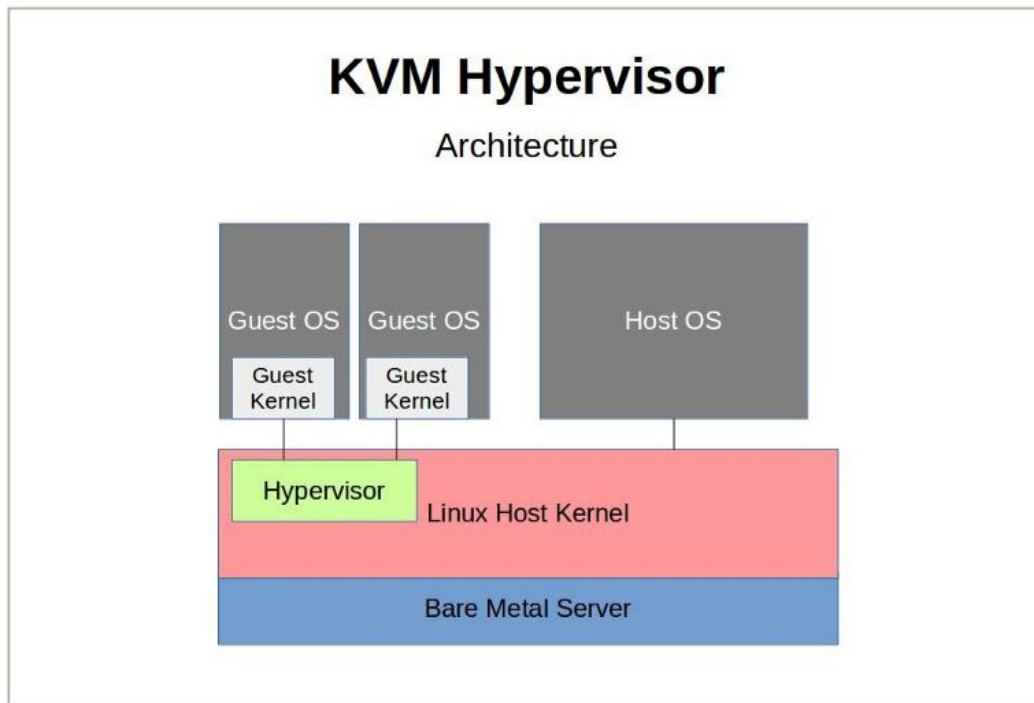
Возможна установка любой операционной системы на выбор (Debian, CentOS, Ubuntu, FreeBSD, Windows Server), либо установка собственного дистрибутива.

KVM требует наличия x86-совместимого процессора с поддержкой одной из технологий аппаратной виртуализации — Intel VT либо AMD SVM.

Программное обеспечение KVM состоит из загружаемого модуля ядра (называемого `kvm.ko`), предоставляющего базовый сервис виртуализации, процессорно-специфического загружаемого модуля `kvm-amd.ko` либо `kvm-intel.ko`, и компонентов пользовательского режима (модифицированного QEMU).

Сам по себе KVM не выполняет эмуляции. Вместо этого программа, работающая в пространстве пользователя, использует интерфейс `/dev/kvm` для настройки адресного пространства гостя виртуальной машины, через него же эмулирует устройства ввода-вывода и видеоадаптер.

KVM позволяет виртуальным машинам использовать немодифицированные образы дисков QEMU, VMware и других, содержащие операционные системы. Каждая виртуальная машина имеет своё собственное виртуальное аппаратное обеспечение: сетевые карты, диск, видеокарту и другие устройства.



# Что docker могу бы заменить, но...



- Системы управления конфигурациями
- Системы запуска сервисов

