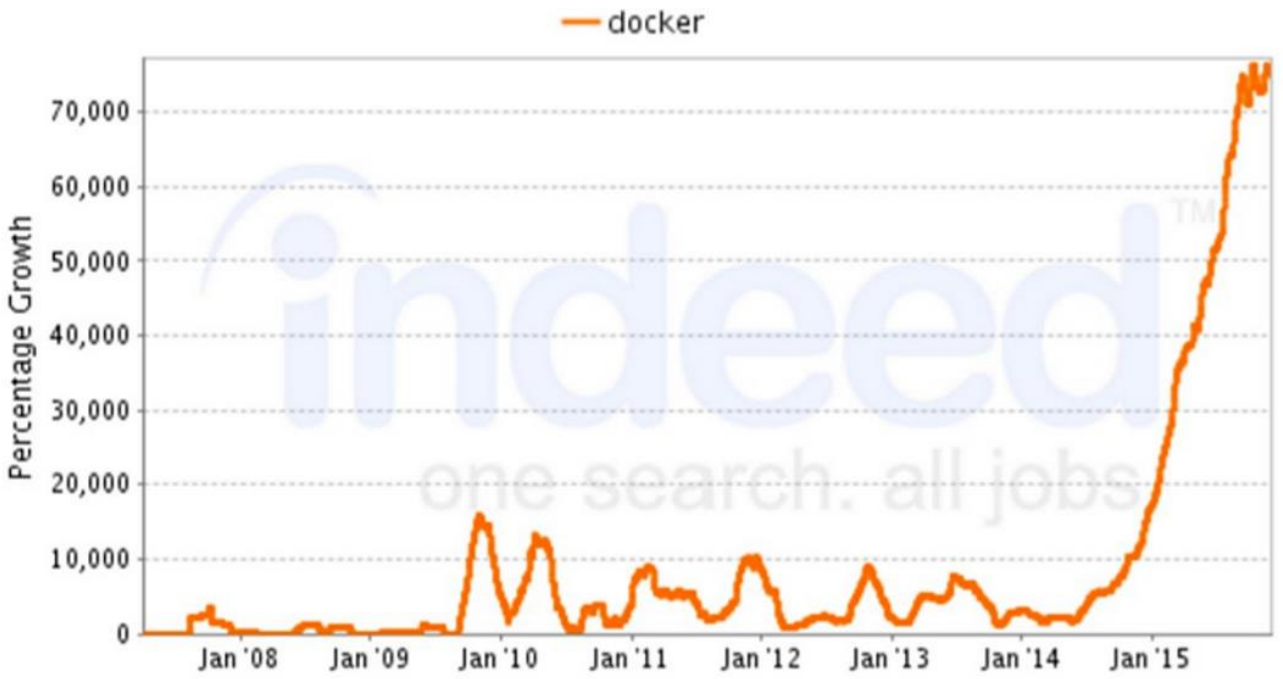
Введение в Docker

Для кого этот курс?

- Хочет научиться в Docker и контейнеризации
- Не знаком в Docker
- Знают что такое консоль и могут отличить bash от cmd
- Базовый опыт в программировании

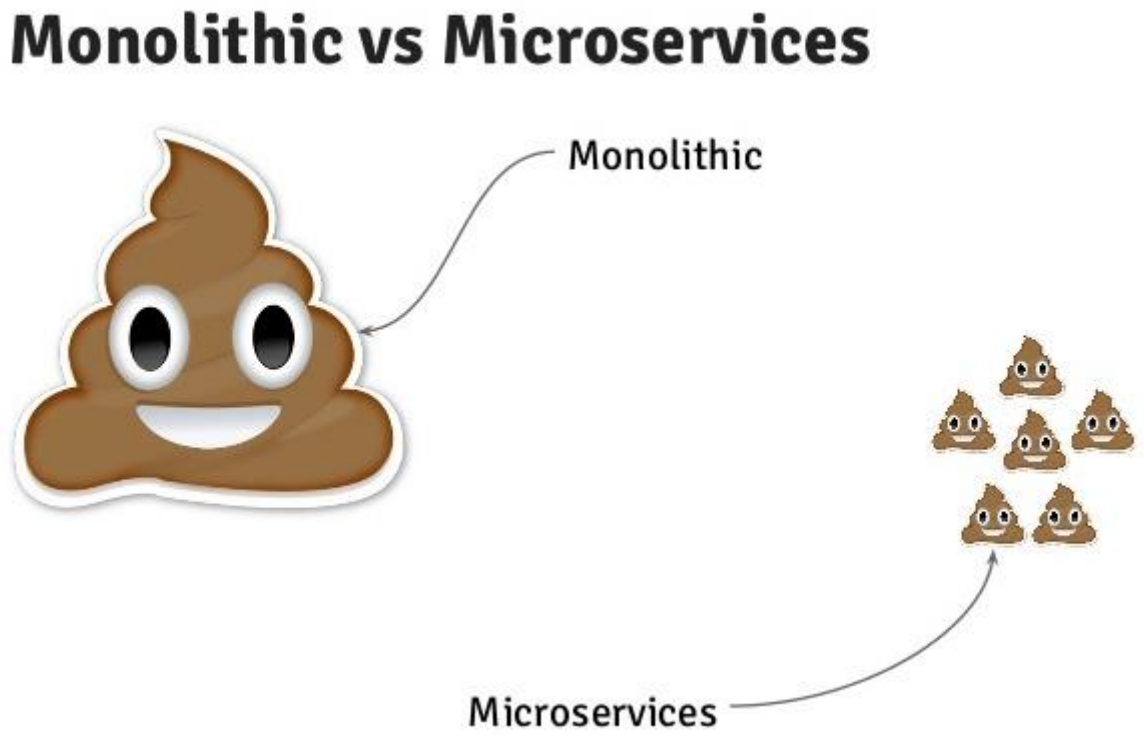
Почему контейнеризация?



Количество вакансий, для которых в качестве требований указано знание Docker, за последний год выросло на 70 000%, сообщает официальный блог разработчиков этого програмного решения, ссылаясь на данные крупнейшего в мире поисковика по сайтам вакансий indeed.com.

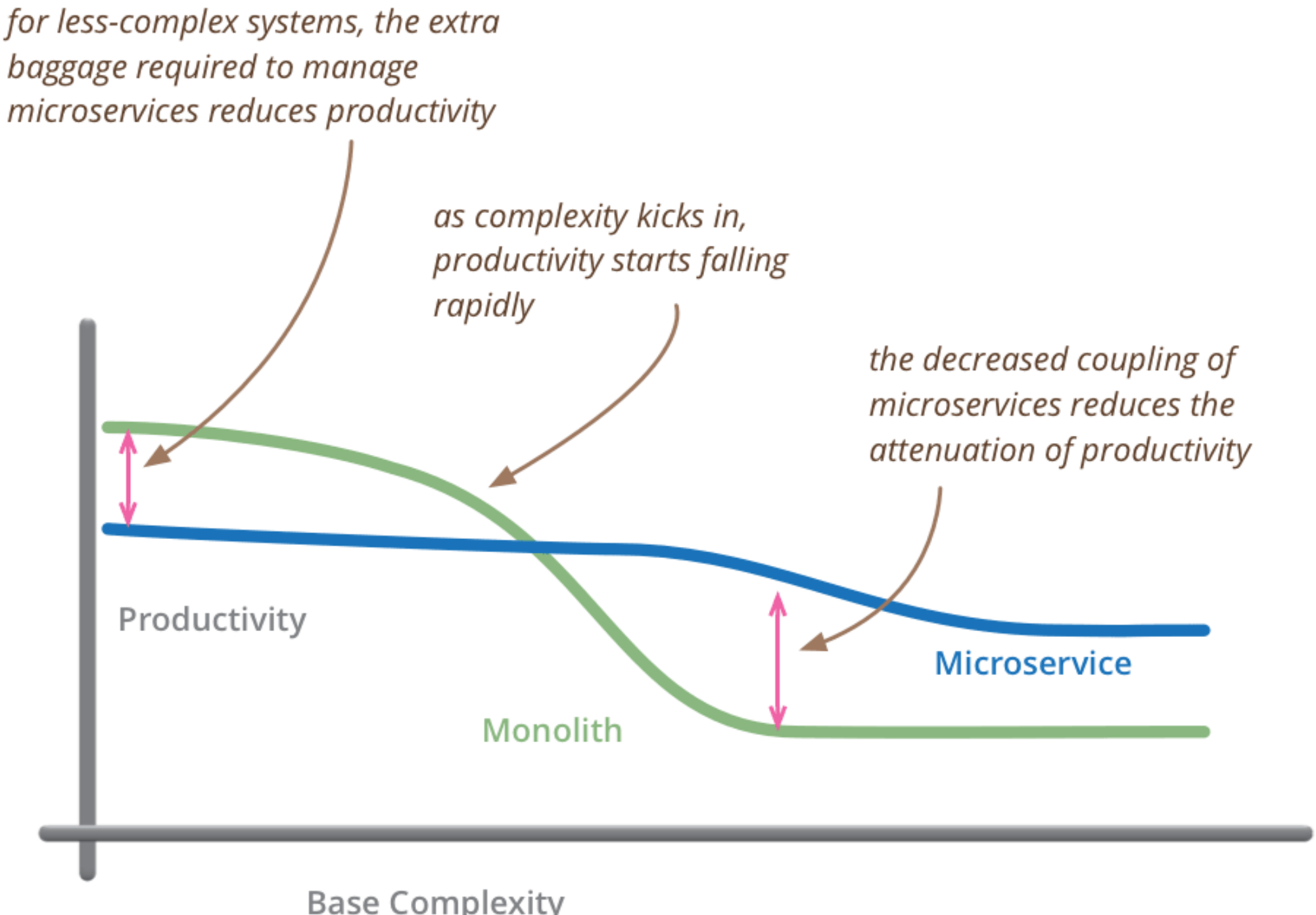
Почему рынок хочет Docker?

История развития проектов



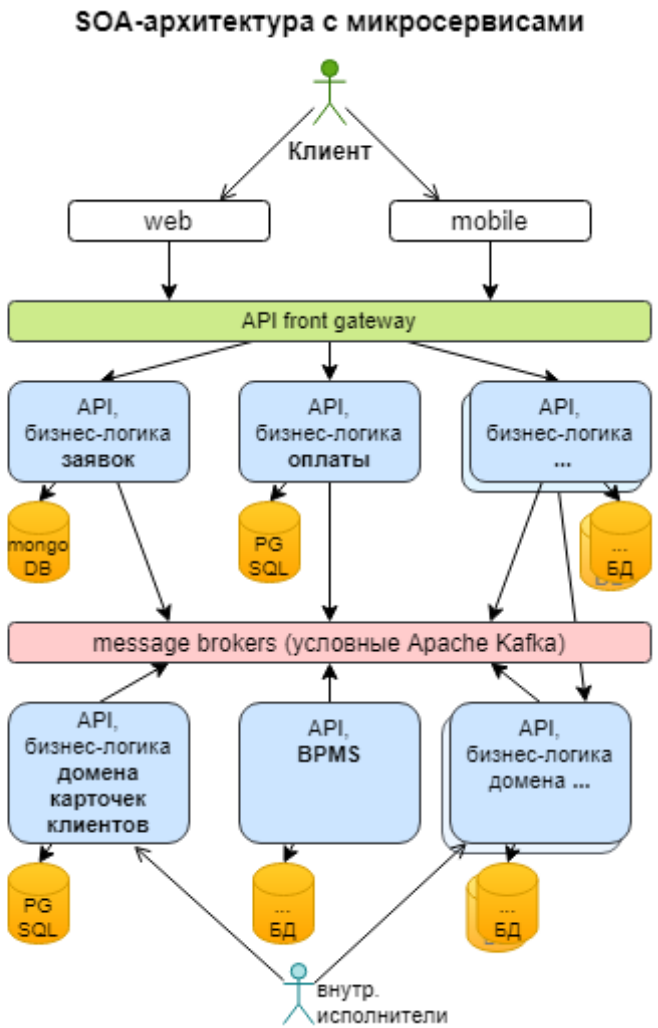
Отличия

Монолит	Микросервисы
Для MVP лучше взять монолит, а дальше, с ростом проекта, дело за микросервисами.	Многократное использование микросервисов позволило легче адаптироваться под новые требования стартапа и масштабировать его.
Монолитные куски стали большие, тяжелые, твердые, трескаются от любого неловкого движения.	Внедрять новый функционал без дополнительных затрат и упрощать интеграцию последующих приложений.
	Использовать разные языки программирования под микросервисы и разные виды протоколов для общения между ними.



Martin Fowler

Что именно хочет бизнес?

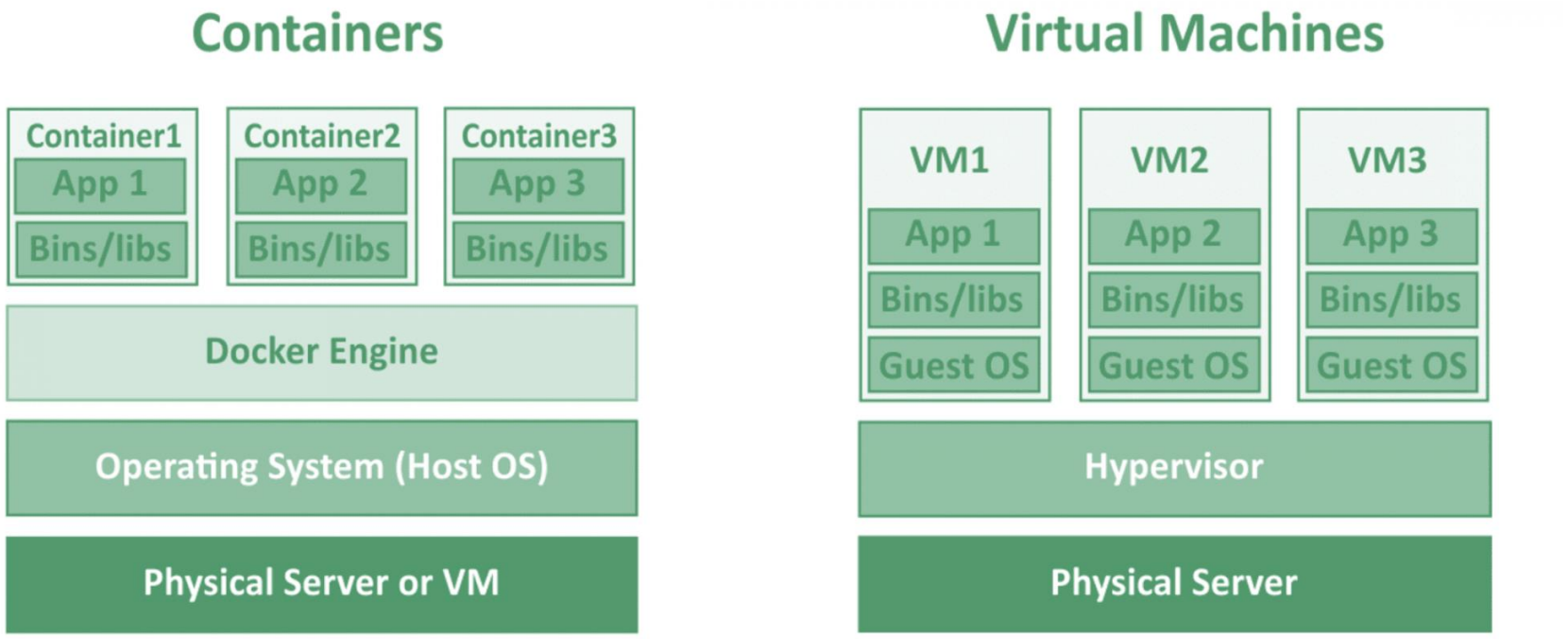


Как с этим работать?

Решение

Контейнер != Виртуальная машина

- Меньше ресурсов
- Контейнеры запускаются только на одной host машине
- Контейнеры менее изолированы, чем виртуальные машины



Основные принципы

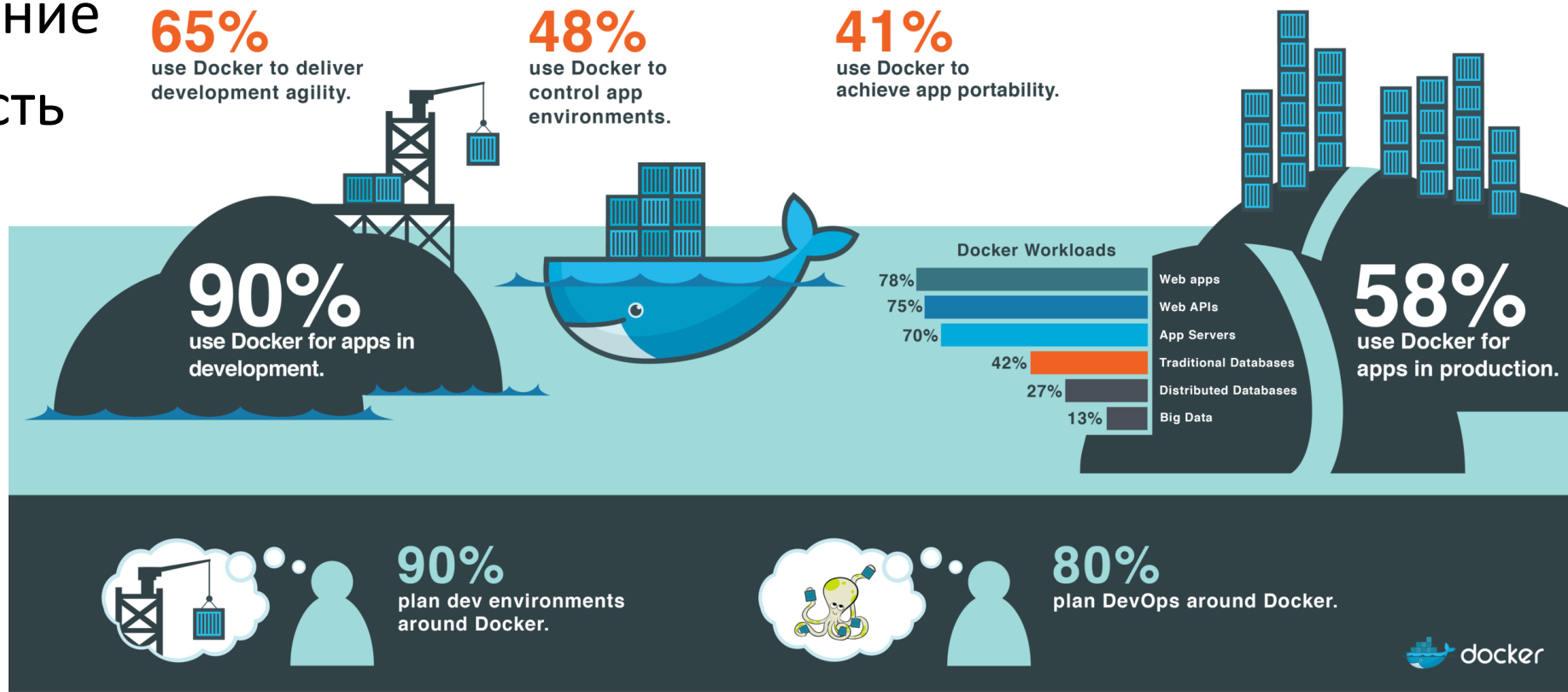
- 1 контейнер — 1 сервис
- Неизменность образа
- Утилизируемость контейнеров
- Отчётность
- Управляемость
- Самодостаточность
- Лимитирование ресурсов (опционально)

Хотим ли мы использовать контейнеры всегда?

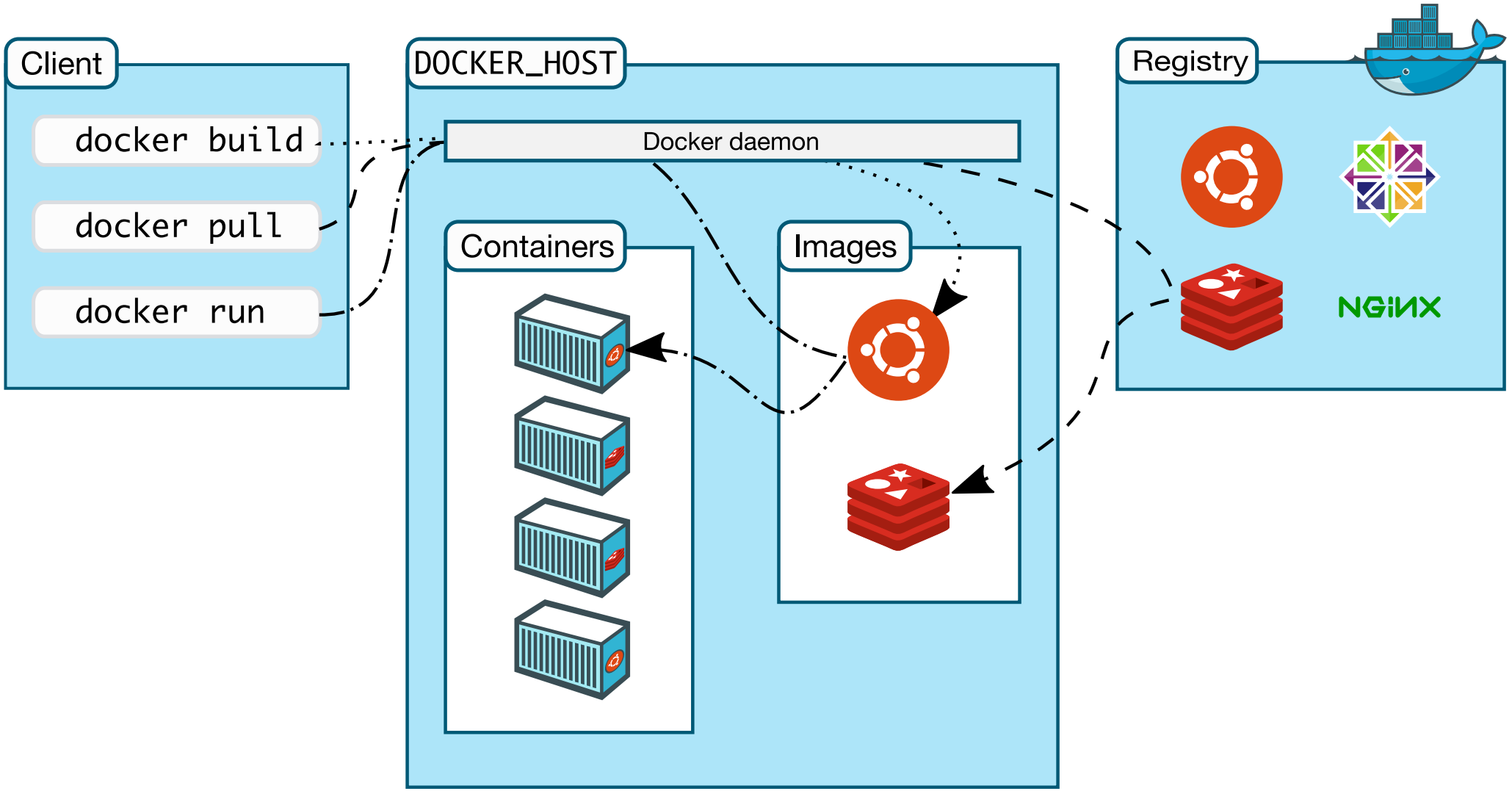
- Что с базой данных\файлами?
- Криптография\шифрование?

Почему Docker?

- Первое решение
- Популярность контейнеров на рынке
- Лёгкое обращение
- Высокая гибкость



Архитектура Docker



Архитектура Docker (2)

- **Docker-демон (Docker-daemon)** — сервер контейнеров, входящий в состав программных средств Docker. Демон управляет Docker-объектами (сети, хранилища, образы и контейнеры). Демон также может связываться с другими демонами для управления сервисами Docker.
- **Docker-клиент (Docker-client / CLI)** — интерфейс взаимодействия пользователя с Docker-демоном. Клиент и Демон — важнейшие компоненты «движка» Докера (Docker Engine). Клиент Docker может взаимодействовать с несколькими демонами.
- **Docker-образ (Docker-image)** — файл, включающий зависимости, сведения, конфигурацию для дальнейшего развертывания и инициализации контейнера.
- **Docker-контейнер (Docker-container)** — это легкий, автономный исполняемый пакет программного обеспечения, который включает в себя все необходимое для запуска приложения: код, среду выполнения, системные инструменты, системные библиотеки и настройки.
- **Реестр (Docker-registry)** — зарезервированный сервер, используемый для хранения docker-образов.

С чего начать?

- https://docs.docker.com/

Немного практики

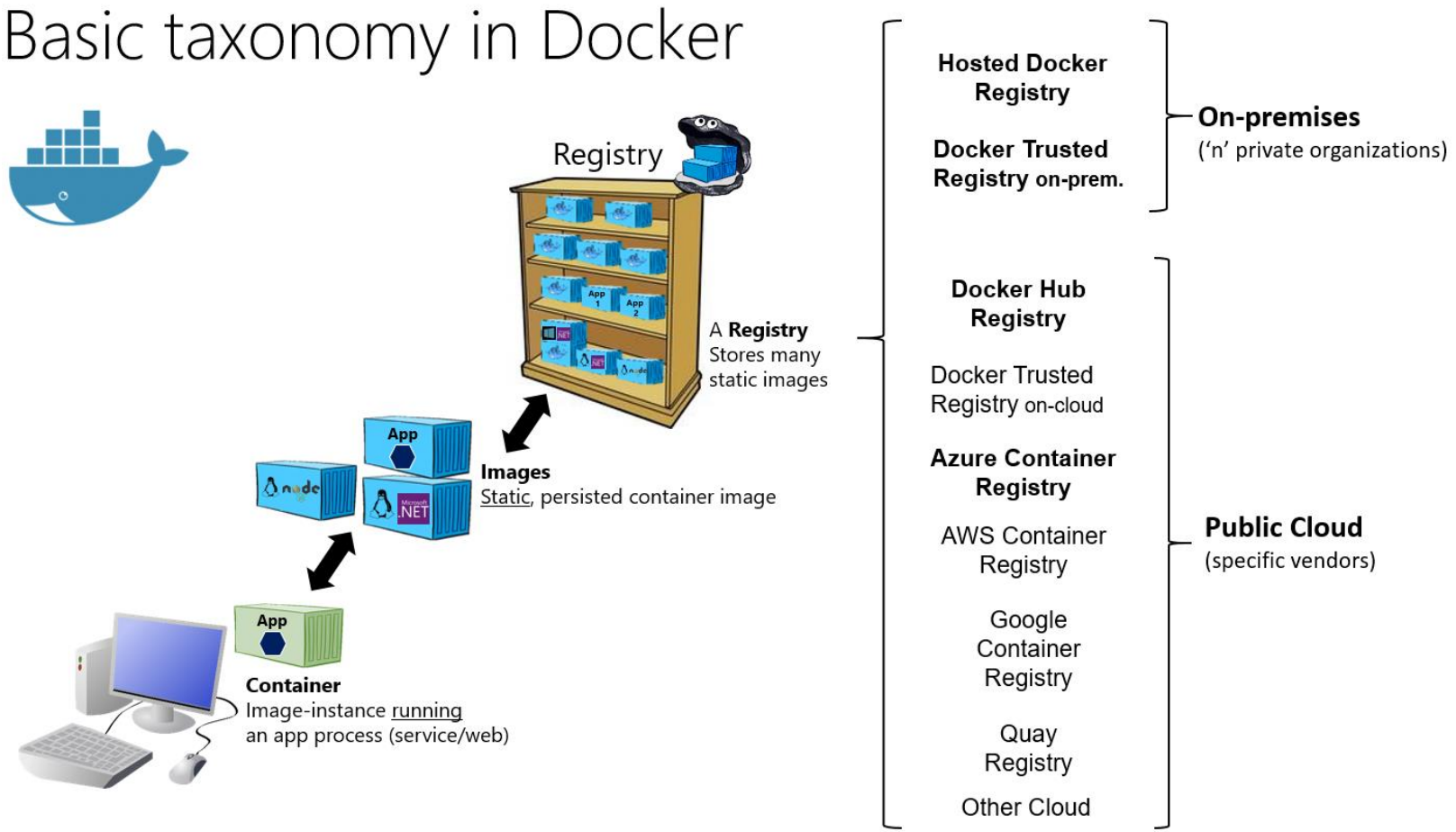
docker run --rm -it ubuntu /bin/bash

- Посмотреть версию ОС
- Установить пакет elinks
- Использовать elinks для доступа в Google

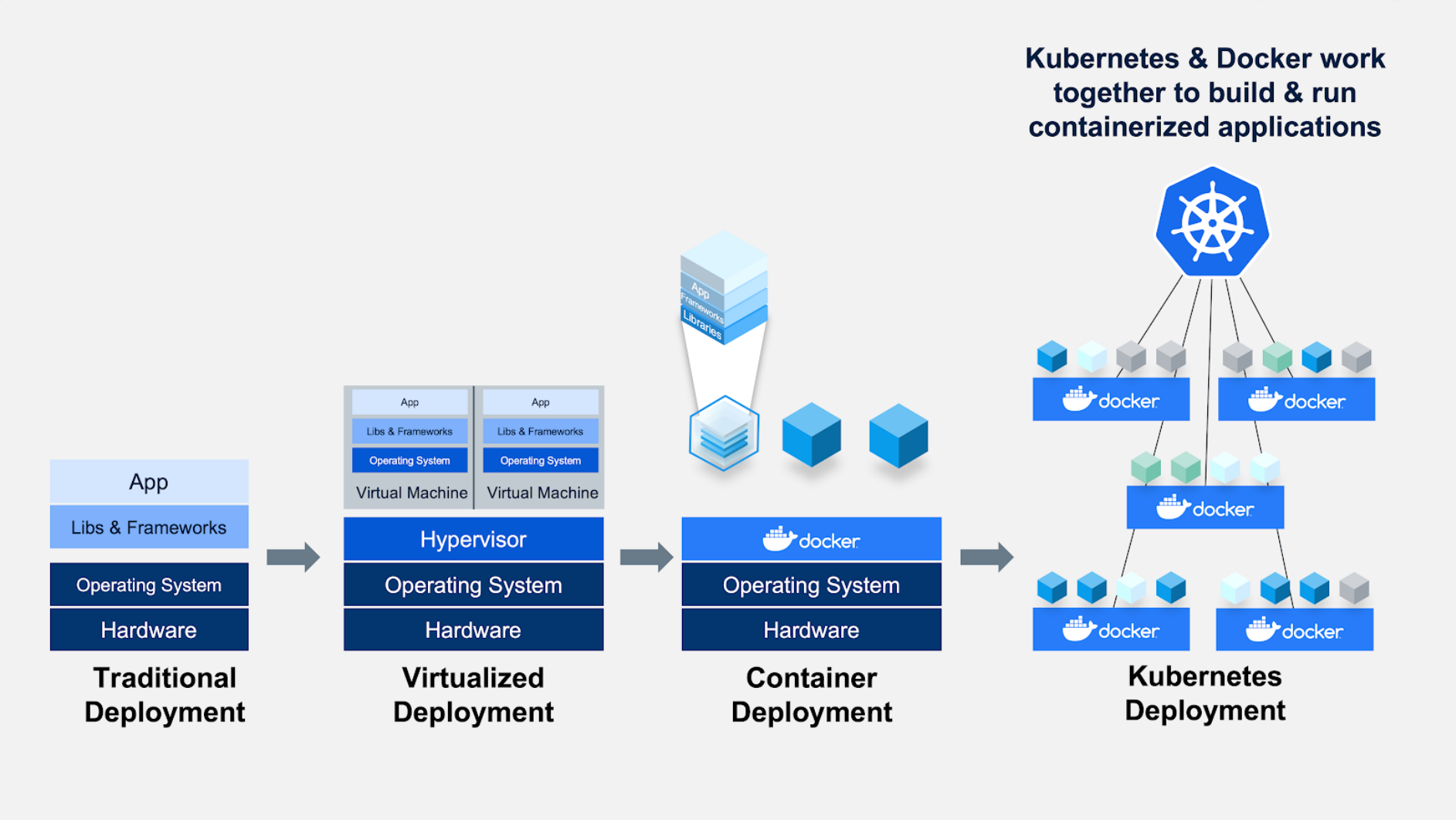
- **Задание**: повторить на образе alpine

Где находятся image?

- Глобальный hub: https://hub.docker.com/

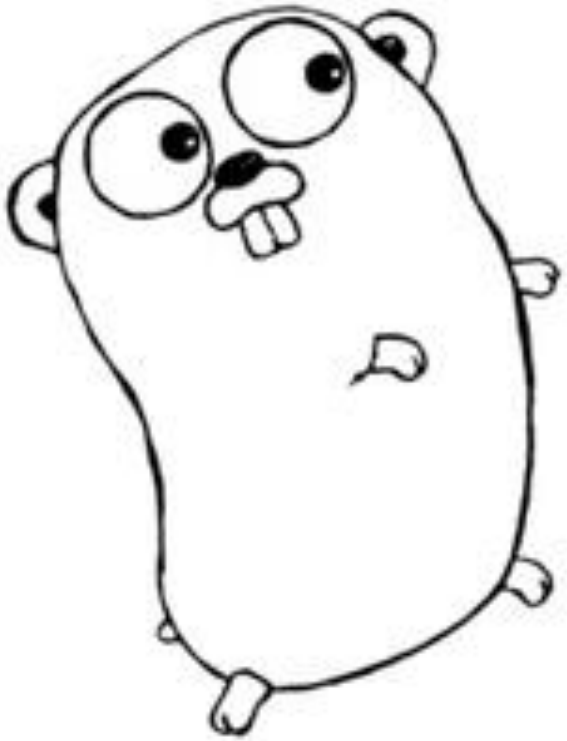


Дальнейшее развитие



Fun facts

- Docker написан на Golang



+

docker

= <3

Аналоги Docker

- https://github.com/containerd/containerd
- https://github.com/cri-o