

Docker. Основы. Образ. Cli. Dockerfile.

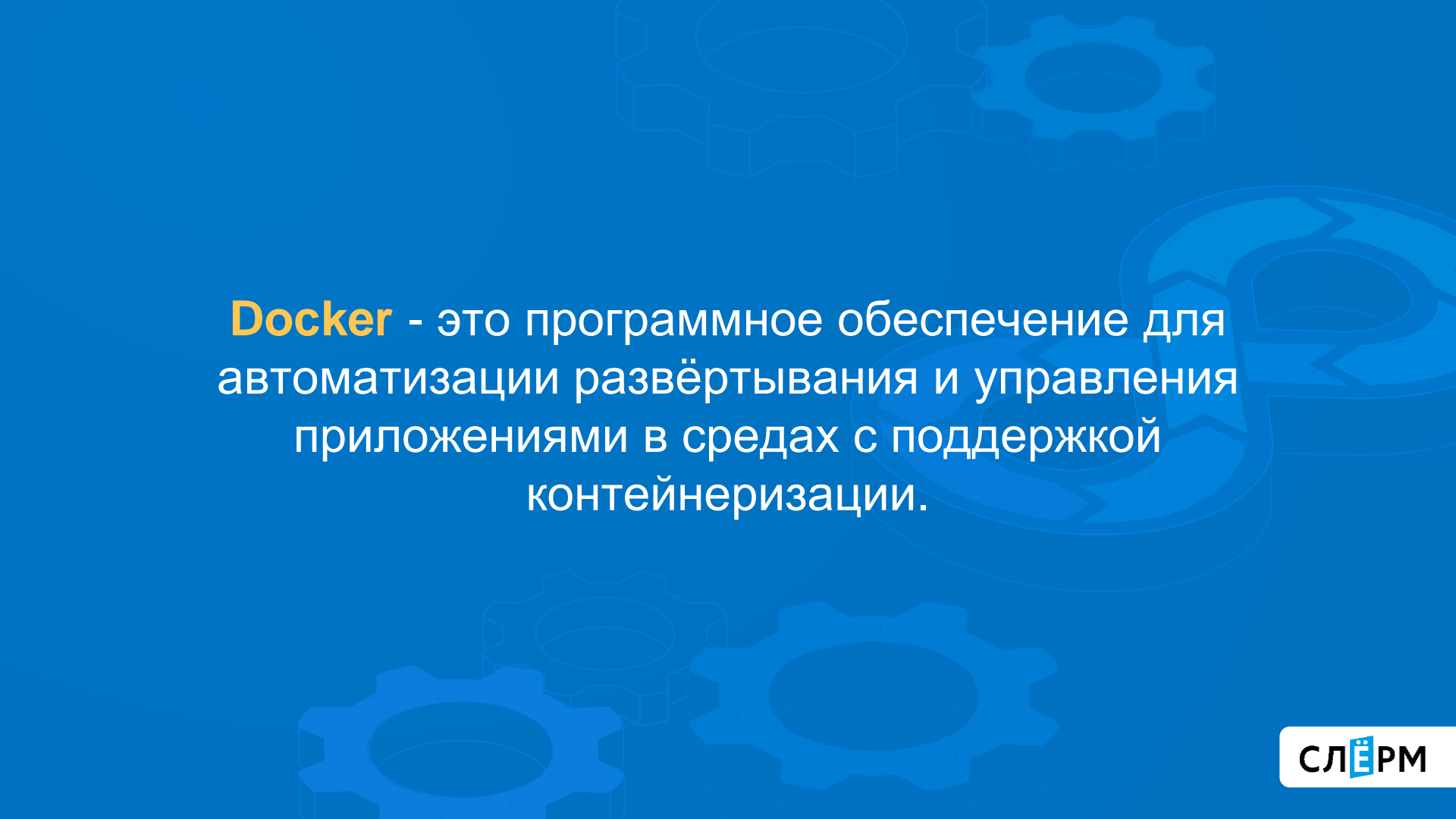


Марсель Ибраев

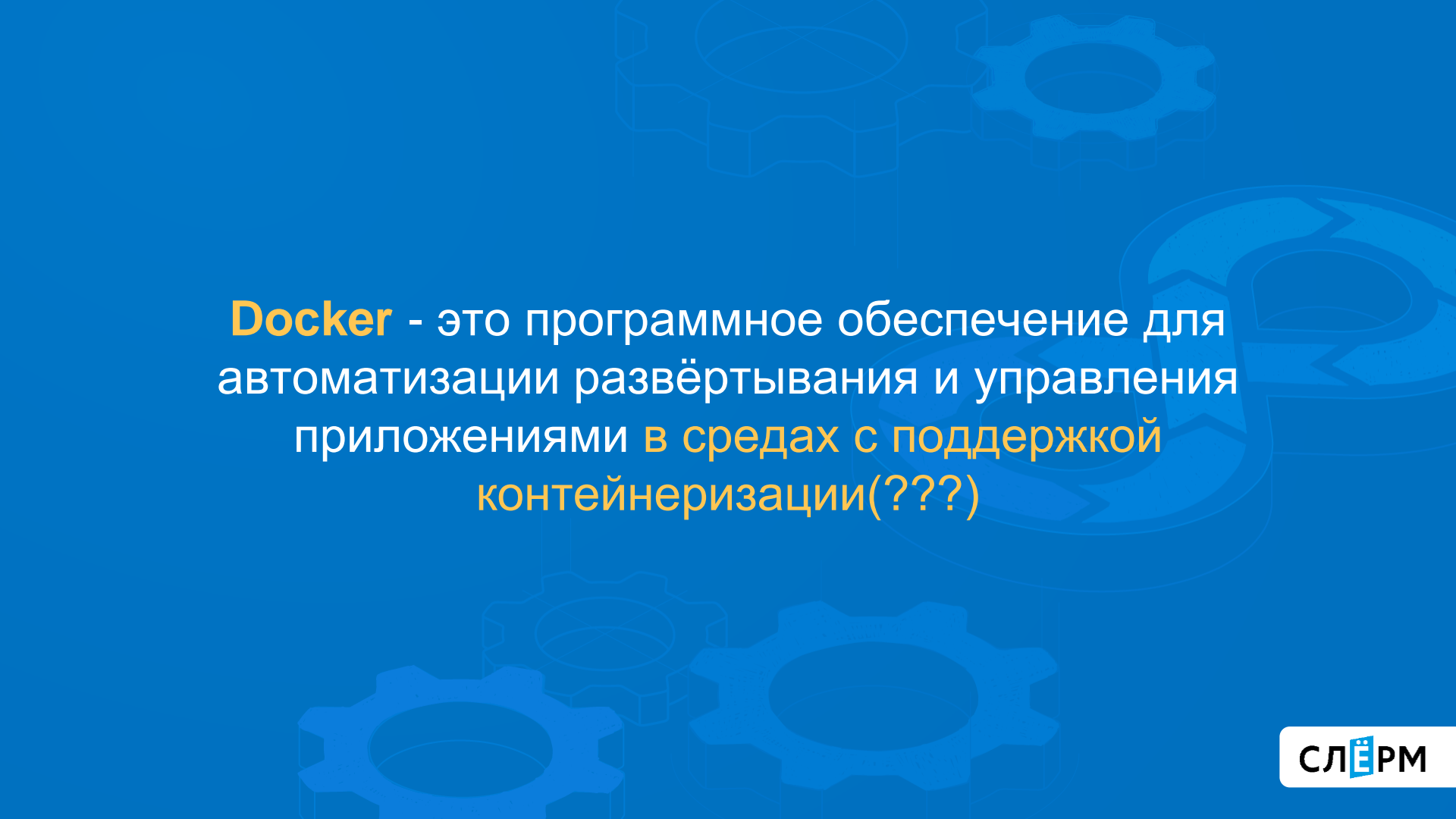
Certified Kubernetes Administrator

Практик, внедряю девопсы и кubernetes

Спикер и разработчик курсов Slurm



Docker - это программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.

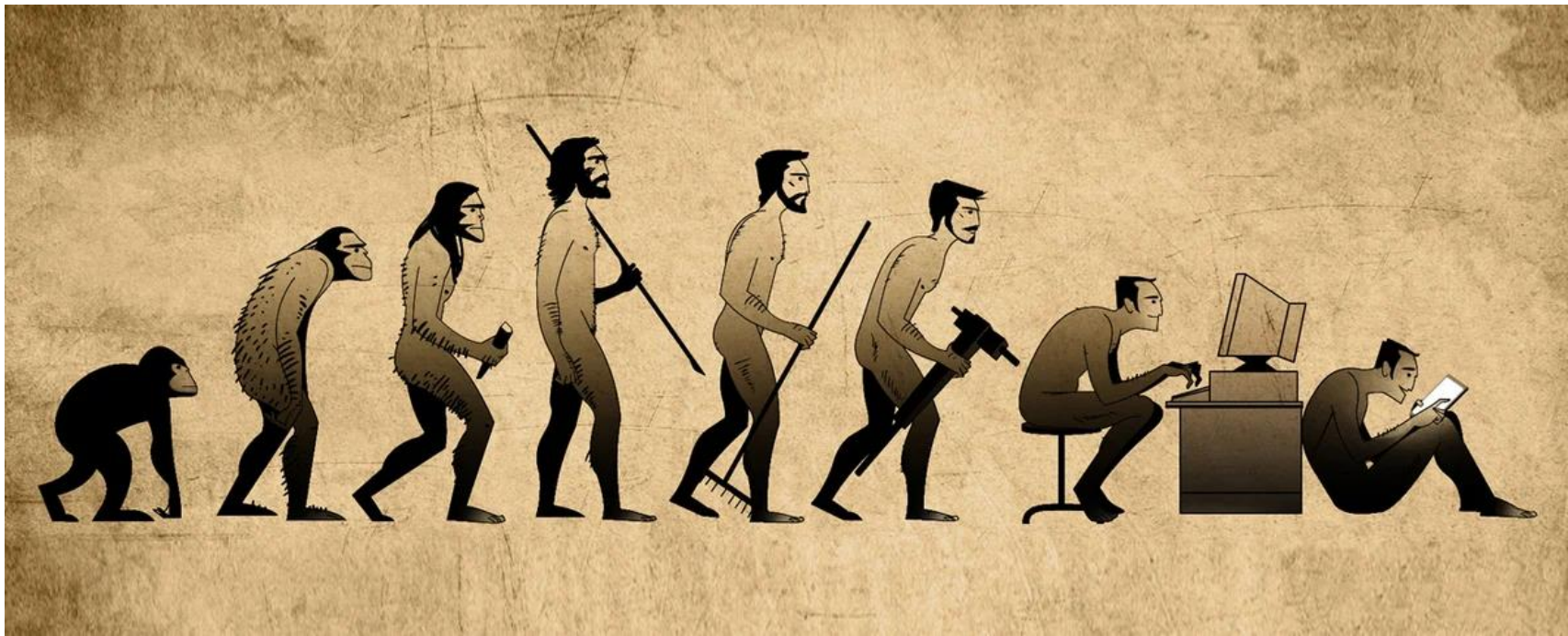


Docker - это программное обеспечение для автоматизации развёртывания и управления приложениями **в средах с поддержкой контейнеризации(???)**





Вернемся в прошлое...





Монолитная эра

- Приложения монолитные
- Куча зависимостей
- Долгая разработка до релиза
- Все инстансы знаем по именам
- Используем виртуализацию
 - Один сервер – несколько VM
 - Resource Management
 - Изоляция окружений





Системы виртуализации

vmware®

Microsoft
Hyper-V



 **QEMU**



Системы виртуализации

vmware®



Microsoft
Hyper-V

Оверхед на гипервизор,
большие образы, медленно



QEMU



Системы виртуализации на уровне ядра



OpenVZ

Systemd-nspawn

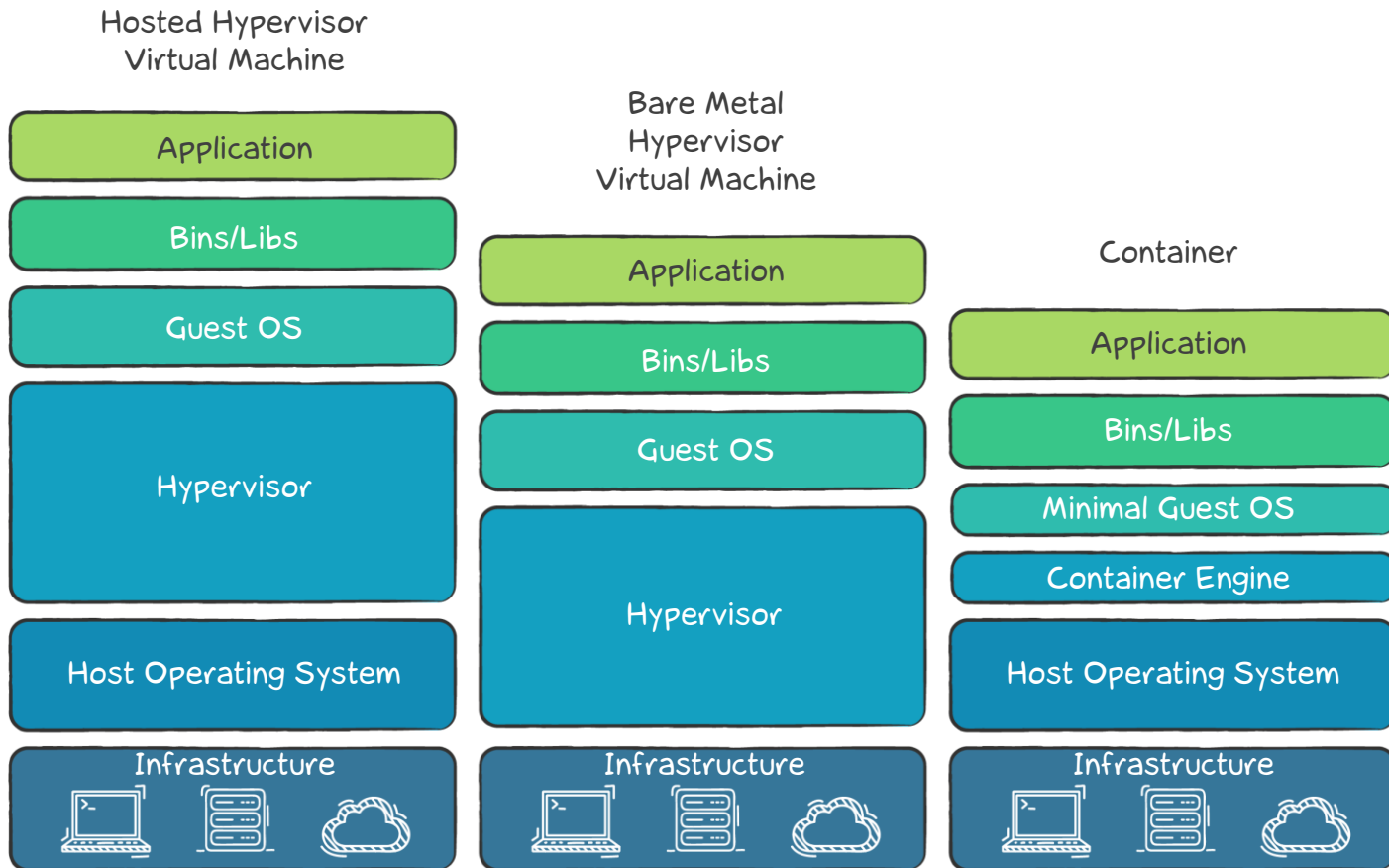


LXC



Разница виртуалки и контейнера

- Виртуальная машина подразумевает виртуализацию железа для запуска гостевой ОС
- Контейнер использует ядро хостовой системы
- В виртуалке может работать любая ОС
- В контейнере только Linux (недавно и Windows)
- Виртуалка хороша для изоляции
- Контейнер для изоляции - плохо





Что используется для контейнеризации?

Namespaces

- PID
- Networking
- Mount
- User

Control Groups

- Memory
- CPU
- Block I/O
- Network



Что используется для контейнеризации?

Namespaces

- PID
- Networking
- Mount
- User

Control Groups

- Memory
- CPU
- Block I/O
- Network

А также:

- Capabilities
- Copy-on-Write
- И т.д.

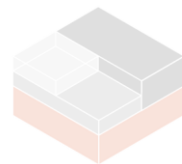


Системы виртуализации на уровне ядра



OpenVZ

Systemd-nspawn



LXC

Большие образы, нет стандарта упаковки,
проблема зависимостей



Эра контейнеров

- Другая философия
- Один процесс – один контейнер
- Все зависимости в контейнере
- Чем меньше образ – тем лучше
- Инстансы становятся эфемерными
- Расцвет Docker





Docker

- Меняет философию
- Стандартизирует упаковку приложения
- Решает вопрос зависимостей
- Гарантирует воспроизводимость
- Минимум (или совсем нет?) оверхеда



Docker

Docker
Daemon

Docker CLI

Dockerfile

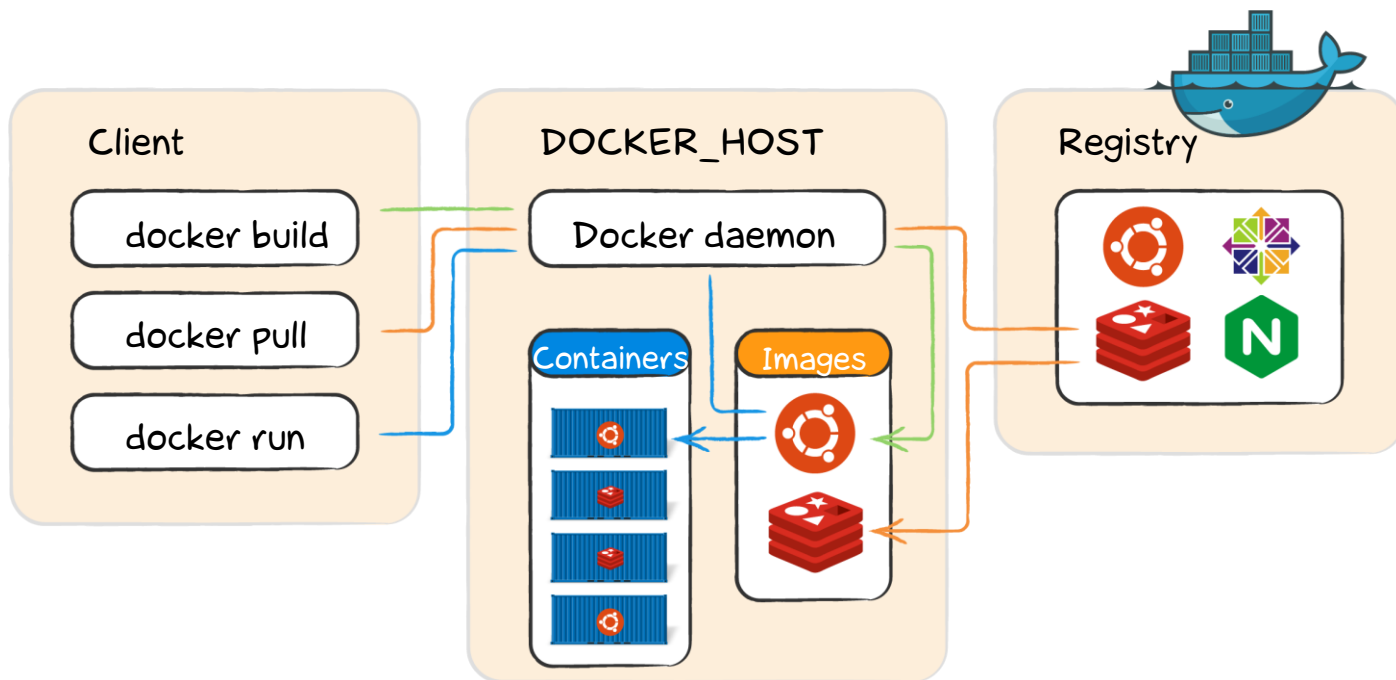
Image

Container

Docker
registry



Docker components





Docker Daemon

- Серверная часть
- Работает на хост-машине
- Скачивает образы и запускает из них контейнеры
- Создает сеть между контейнерами
- Собирает логи контейнеров
- Создает новый образ



Docker CLI

Консольная утилита
для работы с докер-демоном

Может работать не только локально,
но и по сети



Docker – основные команды:

- `docker search <<name>>` - поиск образа в регистри
- `docker pull <<name>>` - скачать образ из регистри на машину
- `docker build <</path/to/dir>>` - собрать образ
- `docker run <<name>>` - запустить контейнер
- `docker rm <<name>>` - удалить контейнер
- `docker ps` - список работающих контейнеров
- `docker logs <<name>>` - логи контейнера
- `docker start/stop/restart <<name>>` - работа с контейнером



Dockerfile

Инструкция для создания образа

Почти каждая команда инструкции – новый слой

Dockerfile

1. `From python:2.7`
2. `RUN mkdir -p /app`
3. `WORKDIR /app`
4. `COPY ./requirements.txt /app/`
5. `RUN pip install -r requirements.txt`
6. `CMD ["python", "main.py"]`

Layer 1 – Base Layer 🔒

Layer 2 🔒

Layer 3 🔒

Layer 4 🔒

Layer 5 🔒

Layer 6 🔒

Image



Image

- Упаковка нашего контейнера
- Из них запускаются контейнеры
- Хранятся в докер-реестрах (registry)
- Имеют hash, имя и tag
- Имеют «слоёную» структуру
- Создаются (build'ются) по инструкции (Dockerfile)



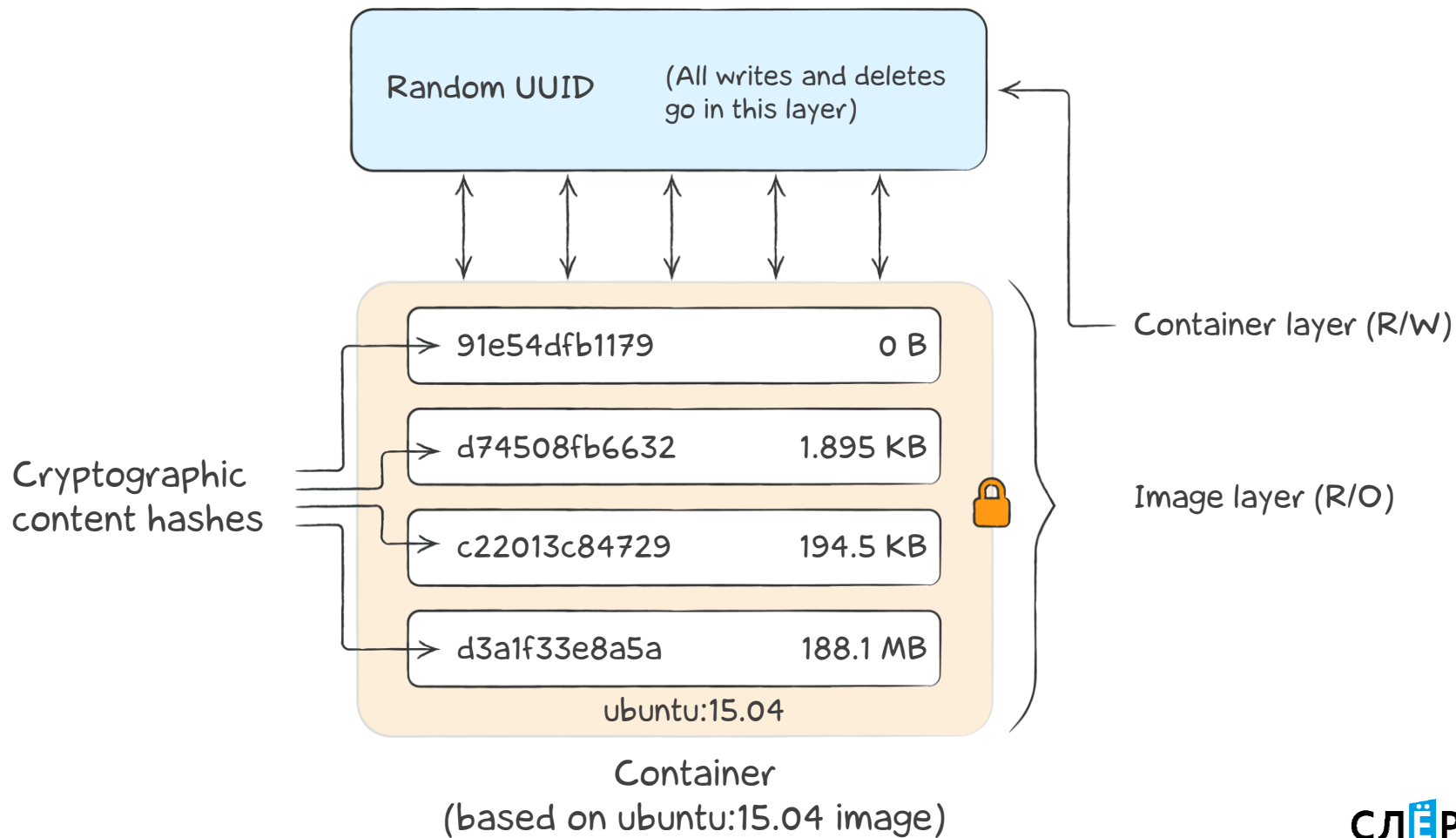
Docker registry

- Хранит образы докера
- Общедоступный стандартный реестр – dockerhub
- Но можно сделать свой



Container

- Запускается из образа
- Изолирован
- Должен содержать в себе всё для работы приложения
- 1 процесс – 1 контейнер





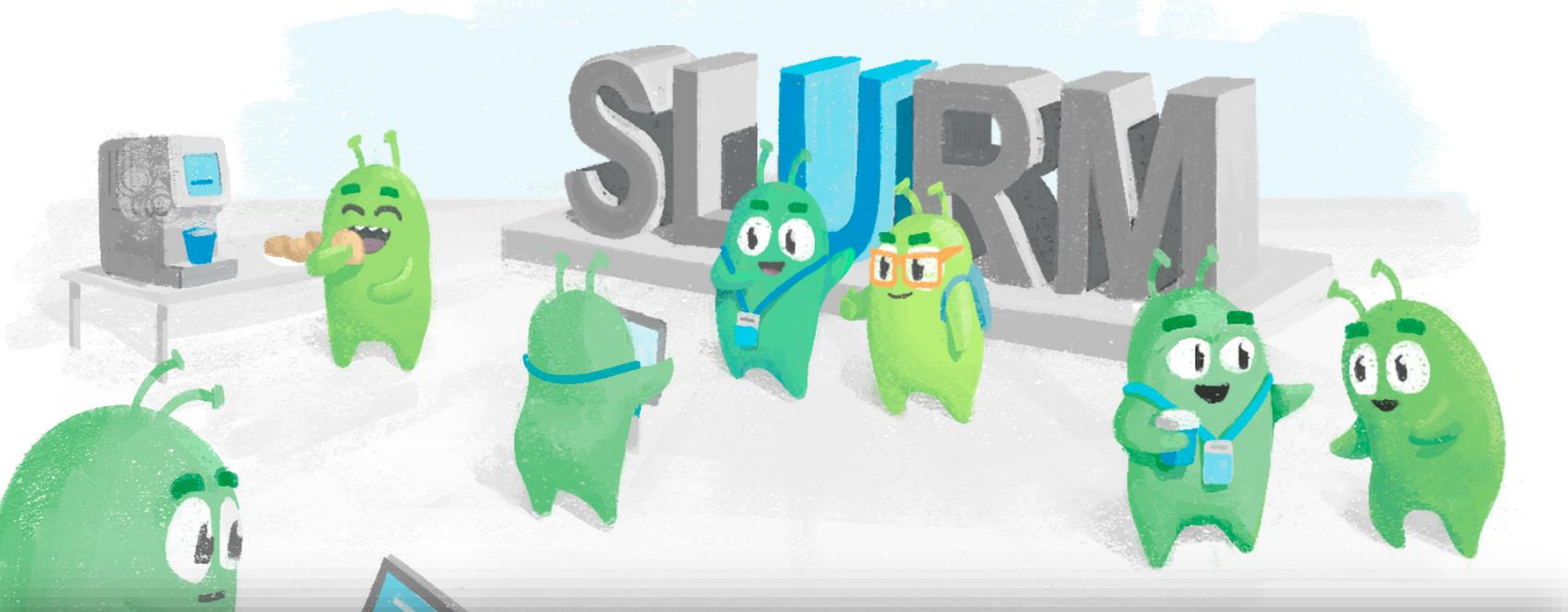
Домашнее задание:

- Посмотреть тома Docker (docker volume)
- Познакомится ближе с механизмами контейнеризации (unshare, netns, systemd-cgls)
- Как ограничить ресурсы контейнеру (cpu/ram)?



Полезные ссылки:

- <https://docs.docker.com/storage/volumes/>
- <https://fabiokung.com/2014/03/13/memory-inside-linux-containers/>
- https://docs.docker.com/config/containers/resource_constraints/
- <https://habr.com/ru/company/selectel/blog/279281/>



southbridge.io

Спасибо!

СЛЁРМ

slurm.io