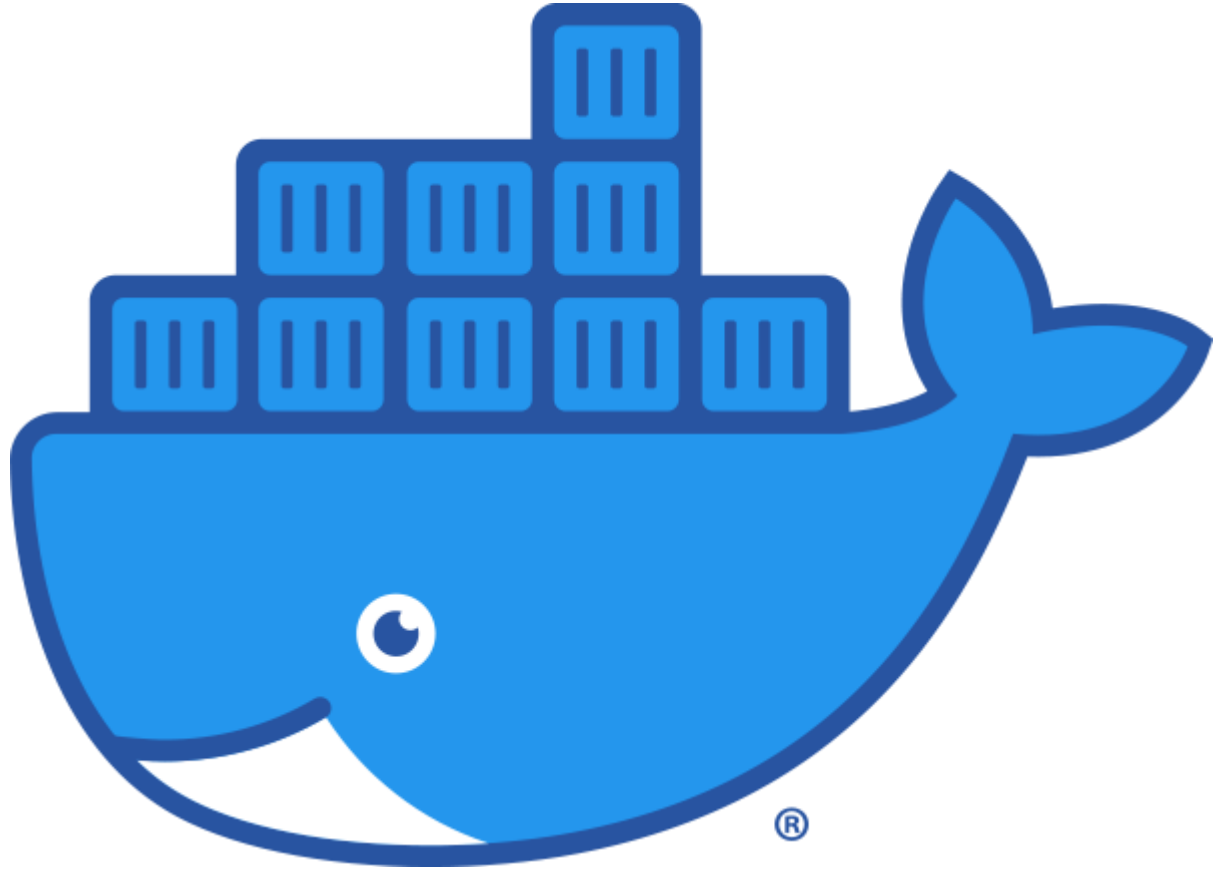




# Введение в Docker

Спикер: Елизавета Михно

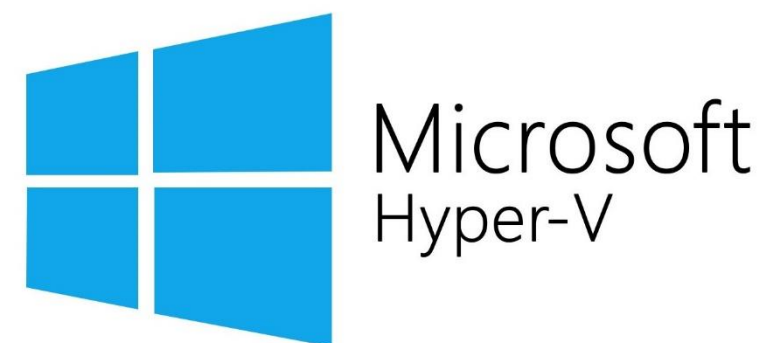


# Виртуализация

vmware®

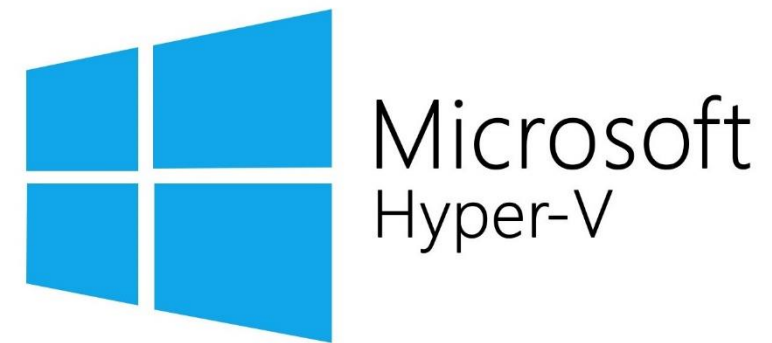
# Виртуализация

vmware®



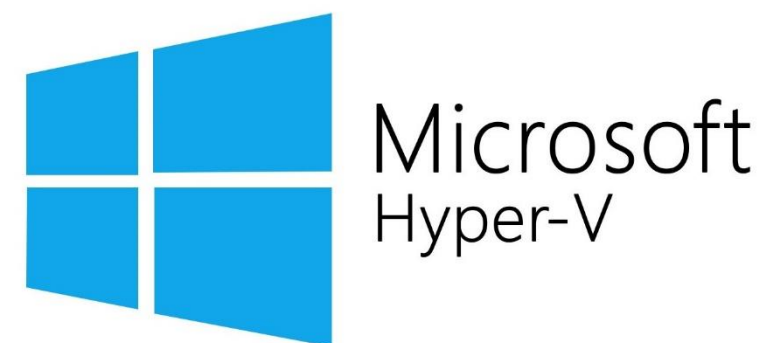
# Виртуализация

vmware®



# Виртуализация

vmware®



# Контейнеры

- Openvz



# Контейнеры

- Openvz
- Systemd-nspawn

[ ● ◀ ] **systemd**



# Контейнеры

- Openvz
- Systemd-nspawn
- Lxc



# Отличия

- Виртуальная машина подразумевает виртуализацию железа для запуска гостевой ОС

# Отличия

- Виртуальная машина подразумевает виртуализацию железа для запуска гостевой ОС

- Контейнер использует ядро хостовой системы

# Отличия

- Виртуальная машина подразумевает виртуализацию железа для запуска гостевой ОС

- В виртуальной машине может быть любая ОС

- Контейнер использует ядро хостовой системы

# Отличия

- Виртуальная машина подразумевает виртуализацию железа для запуска гостевой ОС

- В виртуальной машине может быть любая ОС

- Контейнер использует ядро хостовой системы

- В контейнере только Linux (недавно Windows)

# Отличия

- Виртуальная машина подразумевает виртуализацию железа для запуска гостевой ОС

- В виртуальной машине может быть любая ОС

- Виртуальная машина хороша для изоляции

- Контейнер использует ядро хостовой системы

- В контейнере только Linux (недавно Windows)

# Отличия

- Виртуальная машина подразумевает виртуализацию железа для запуска гостевой ОС

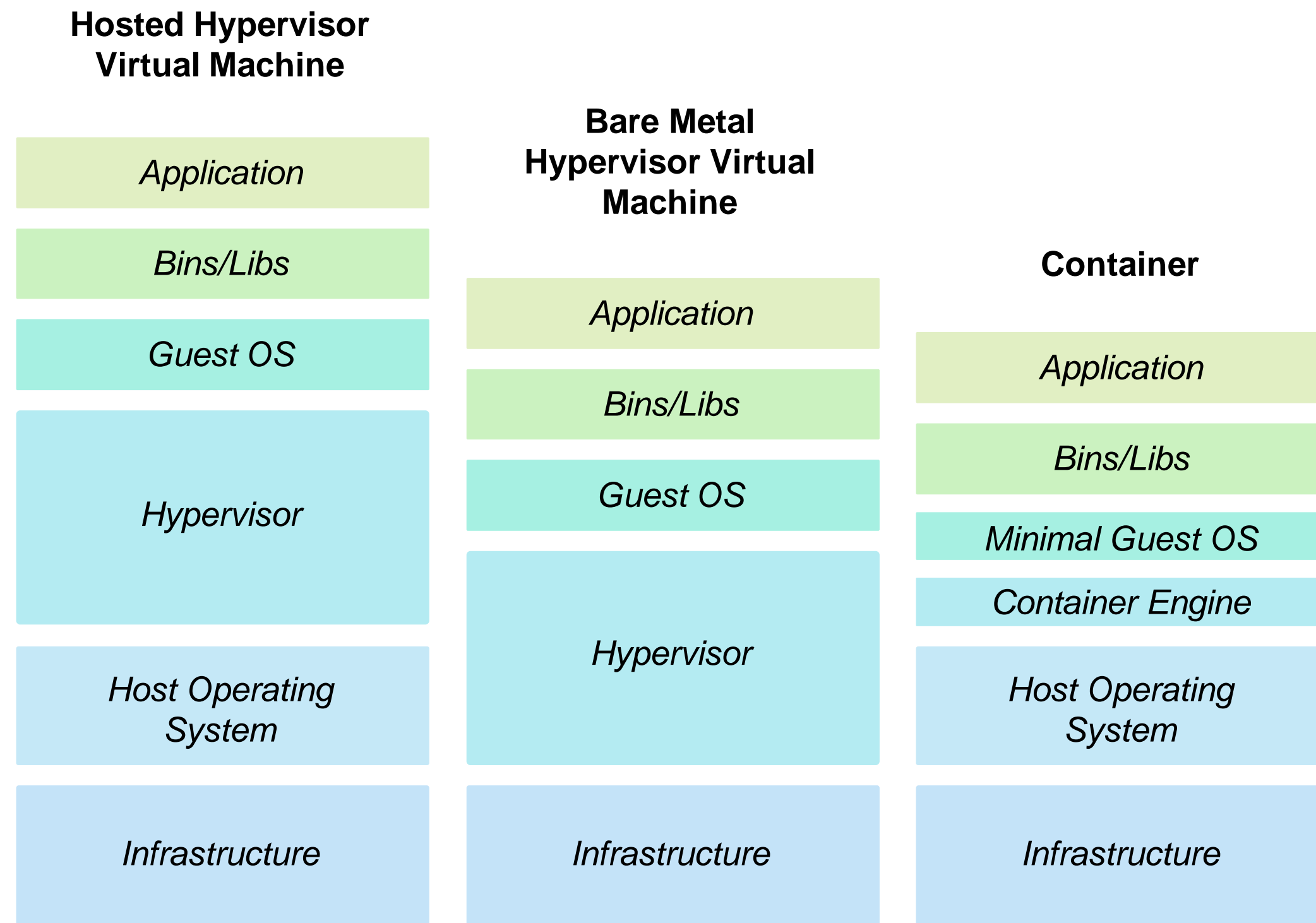
- В виртуальной машине может быть любая ОС

- Виртуальная машина хороша для изоляции

- Контейнер использует ядро хостовой системы

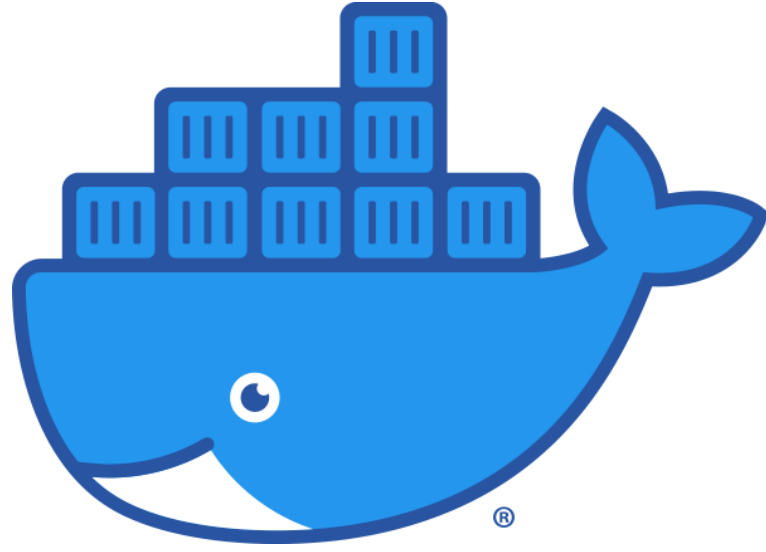
- В контейнере только Linux (недавно Windows)

- Контейнер для изоляции - плохо



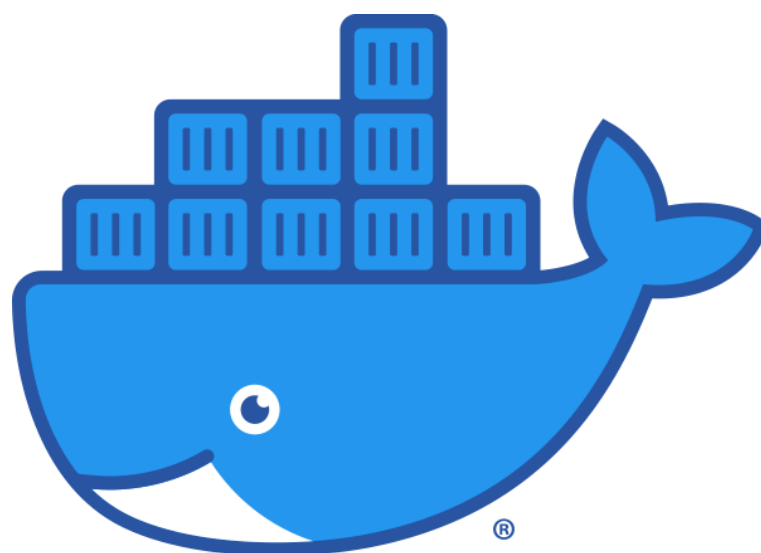


# Docker



# Docker

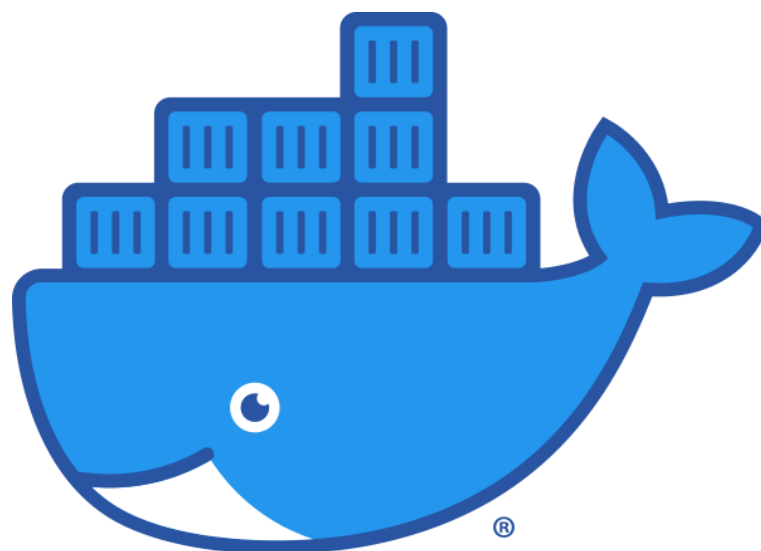
1 процесс - 1 контейнер



# Docker

1 процесс - 1 контейнер

Все зависимости в  
контейнере

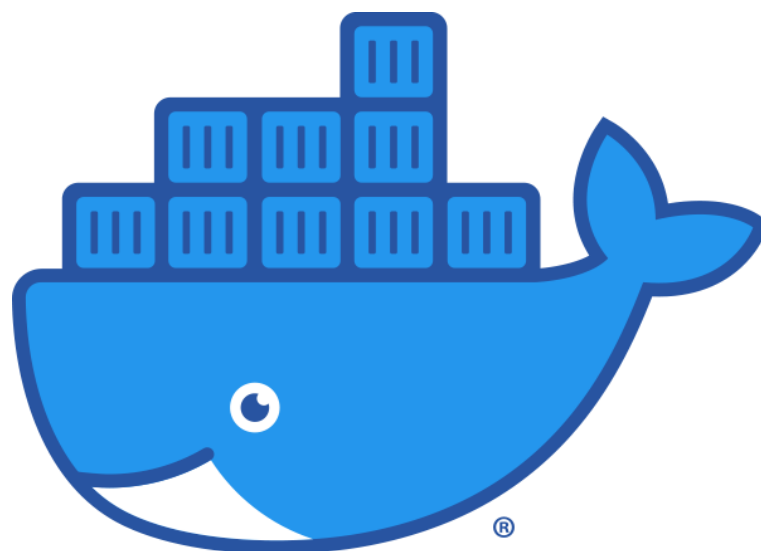


# Docker

1 процесс - 1 контейнер

Все зависимости в  
контейнере

Чем меньше образ,  
тем лучше

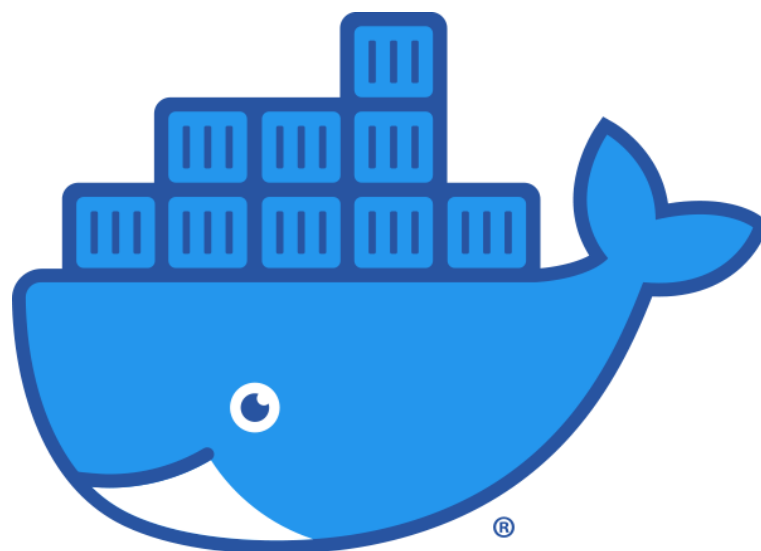


# Docker

1 процесс - 1 контейнер

Все зависимости в  
контейнере

Чем меньше образ,  
тем лучше



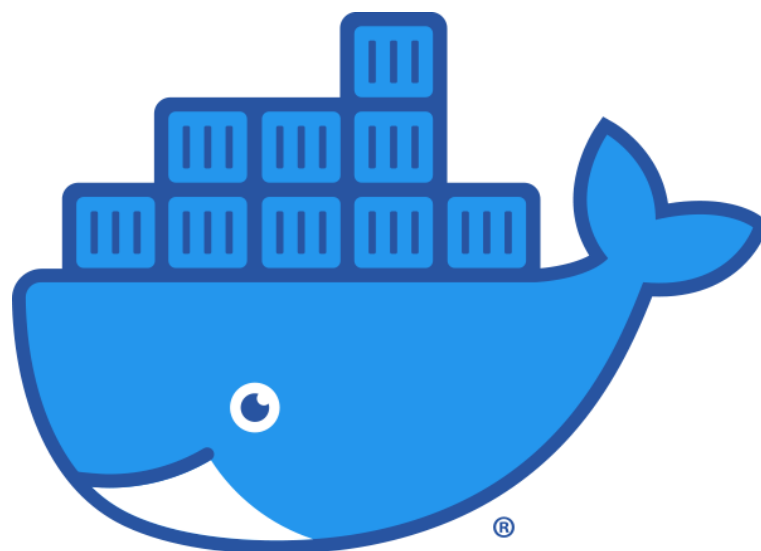
Стандартизация  
упаковки приложения

# Docker

1 процесс - 1 контейнер

Все зависимости в  
контейнере

Чем меньше образ,  
тем лучше



Стандартизация  
упаковки приложения

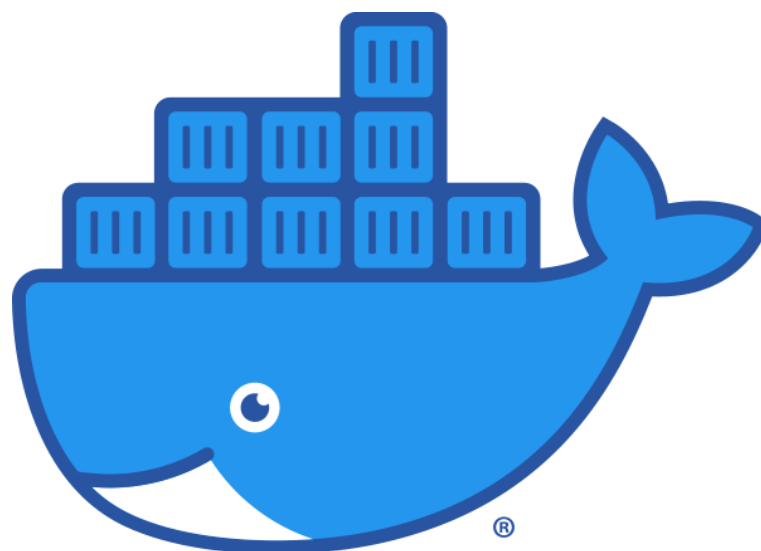
Гарантирует  
воспроизводимость

# Docker

1 процесс - 1 контейнер

Все зависимости в  
контейнере

Чем меньше образ,  
тем лучше



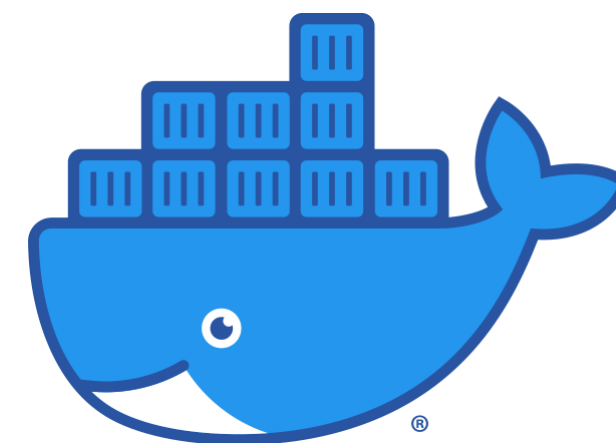
Стандартизация  
упаковки приложения

Гарантирует  
воспроизводимость

Минимум оверхеда

# 12 факторов разработки

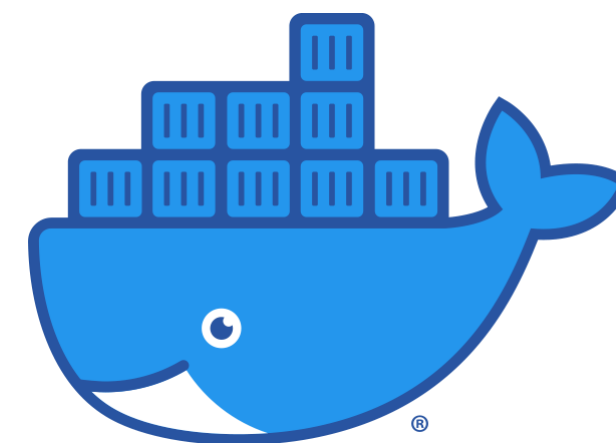
1. Единый репозиторий исходного кода





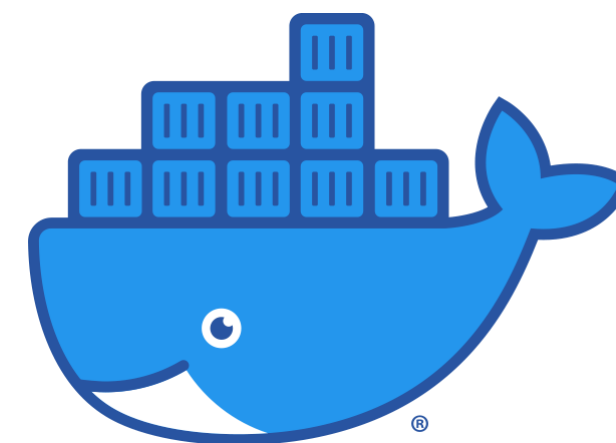
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости



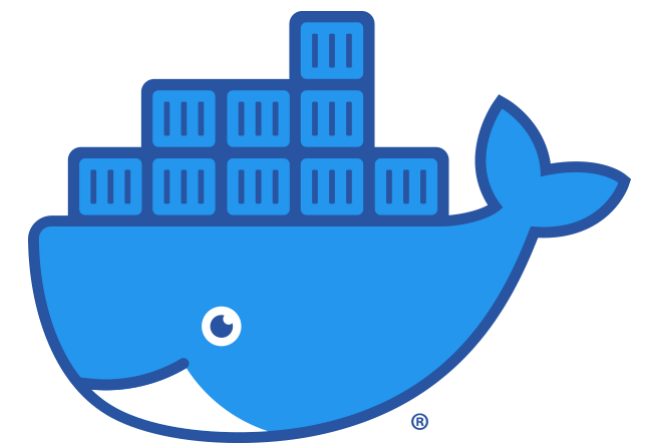
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация



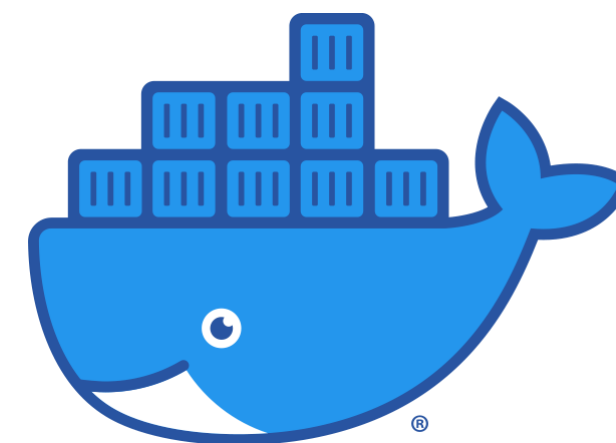
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)



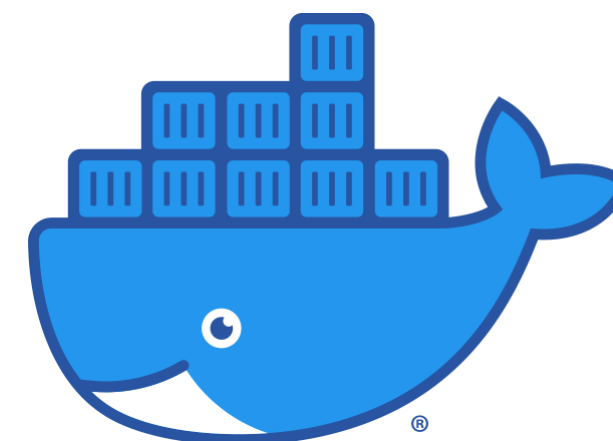
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение



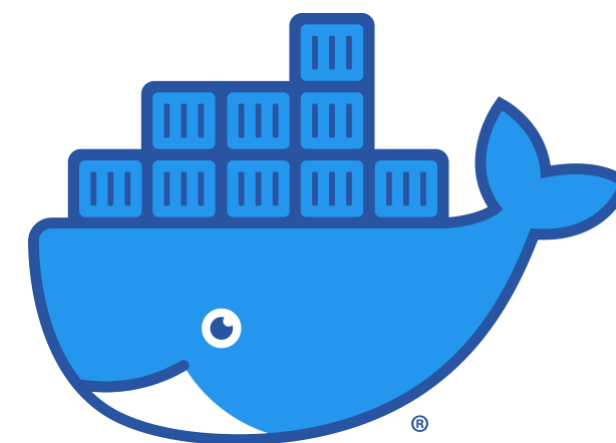
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение
6. Изолированные процессы



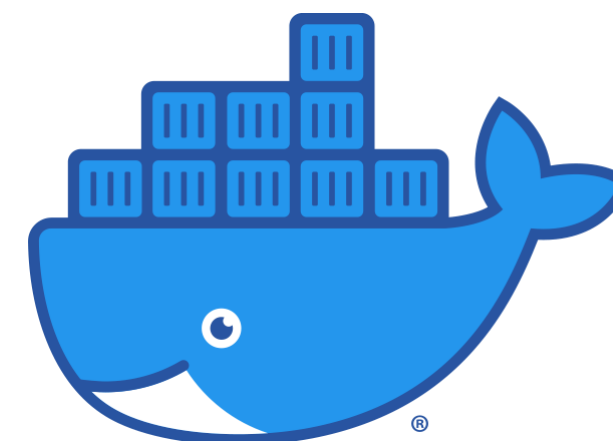
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение
6. Изолированные процессы
7. Привязка портов (Port binding)



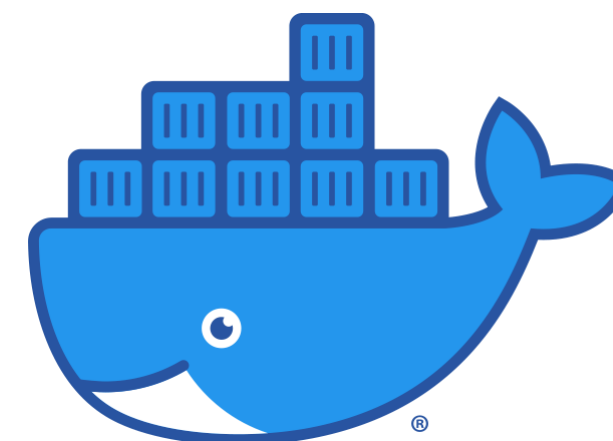
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение
6. Изолированные процессы
7. Привязка портов (Port binding)
8. Параллелизм



# 12 факторов разработки

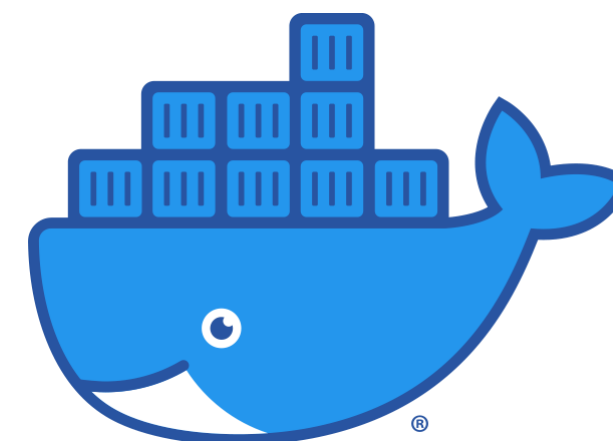
1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение
6. Изолированные процессы
7. Привязка портов (Port binding)
8. Параллелизм
9. Утилизируемость (Disposability)





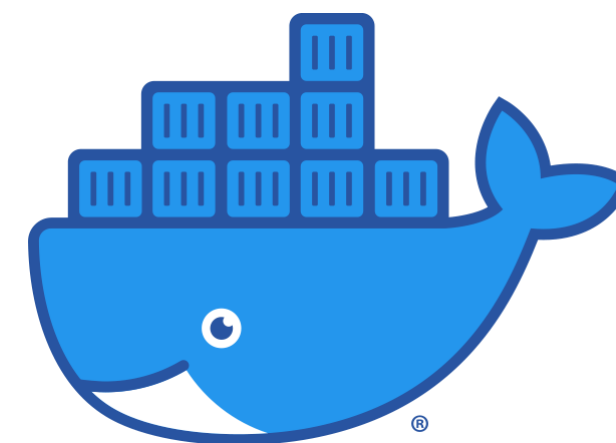
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение
6. Изолированные процессы
7. Привязка портов (Port binding)
8. Параллелизм
9. Утилизируемость (Disposability)
10. Соответствие среды разработки/работы приложения



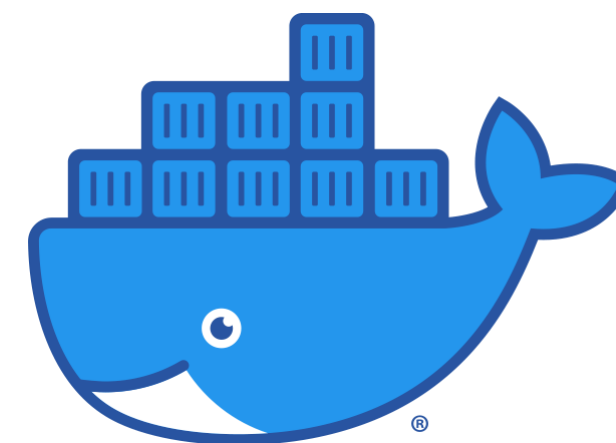
# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение
6. Изолированные процессы
7. Привязка портов (Port binding)
8. Параллелизм
9. Утилизируемость (Disposability)
10. Соответствие среды разработки/работы приложения
11. Журналирование (Logs)



# 12 факторов разработки

1. Единый репозиторий исходного кода
2. Зависимости
3. Конфигурация
4. Сторонние службы (Backing Services)
5. Сборка, релиз, выполнение
6. Изолированные процессы
7. Привязка портов (Port binding)
8. Параллелизм
9. Утилизируемость (Disposability)
10. Соответствие среды разработки/работы приложения
11. Журналирование (Logs)
12. Задачи администрирования



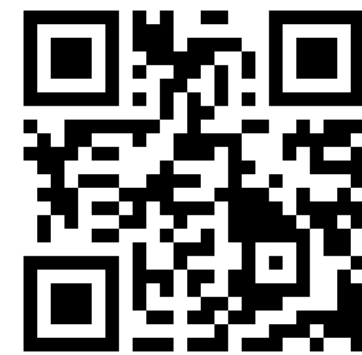
СЛЁРМ



slurm.io



Southbridge



southbridge.io