

СЛЁРМ

+



Southbridge

Docker, его устройство, Dockerfile, docker- compose

План

- Технология Docker
- Docker engine
- Dockerfile
- Docker-compose
- Docker для Kubernetes

Docker

Не про контейнеры!

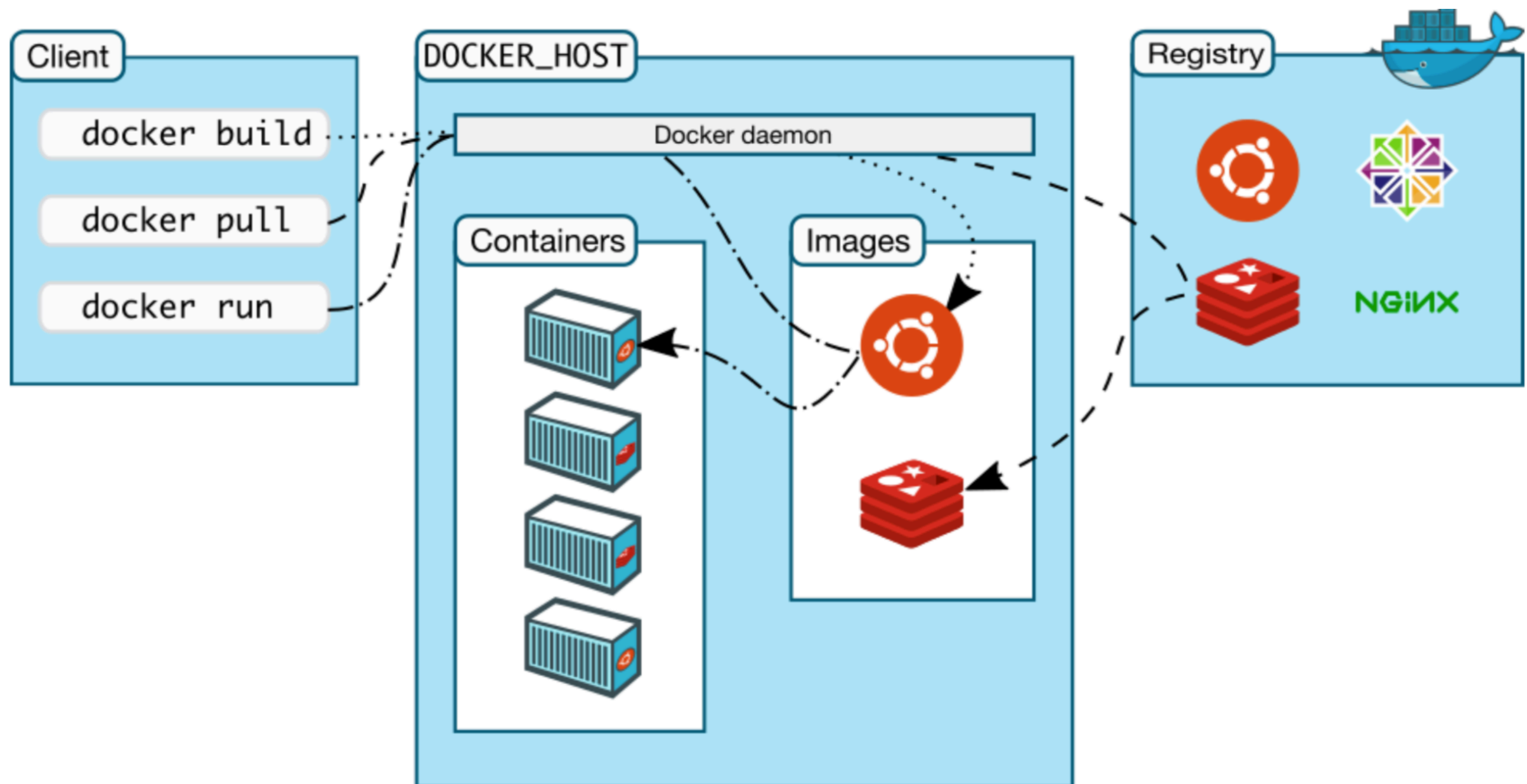
Docker

- Стандарт упаковки приложений
 - Сборка
 - Поставка
- Воспроизводимость
- Консистентность

Особенности

- Нужно адаптировать приложения
- Для эффективного использования нужно много дополнительных сущностей

Docker engine



Контейнер

- Namespaces
 - PID
 - Networking
 - Mount
 - User
- Control Groups
 - Memory
 - CPU
 - Block I/O
 - Network

PID NS

- Процессы в PID ns видят только процессы в том же ns
- У каждого PID ns своя нумерация начинающаяся с 1
- Когда завершается PID 1 весь ns перестает существовать

NET NS

- Процессы в рамках своего net ns получают свое сетевое окружение
 - Сетевые интерфейсы (в том числе lo)
 - Таблицу маршрутизации
 - Правила iptables
 - Сокеты
- Veth
- Общий lo для нескольких контейнеров

MNT NS

- Процесс может иметь свою root fs
- Процесс может иметь свои приватные маунты
 - /tmp
 - /sys и /proc

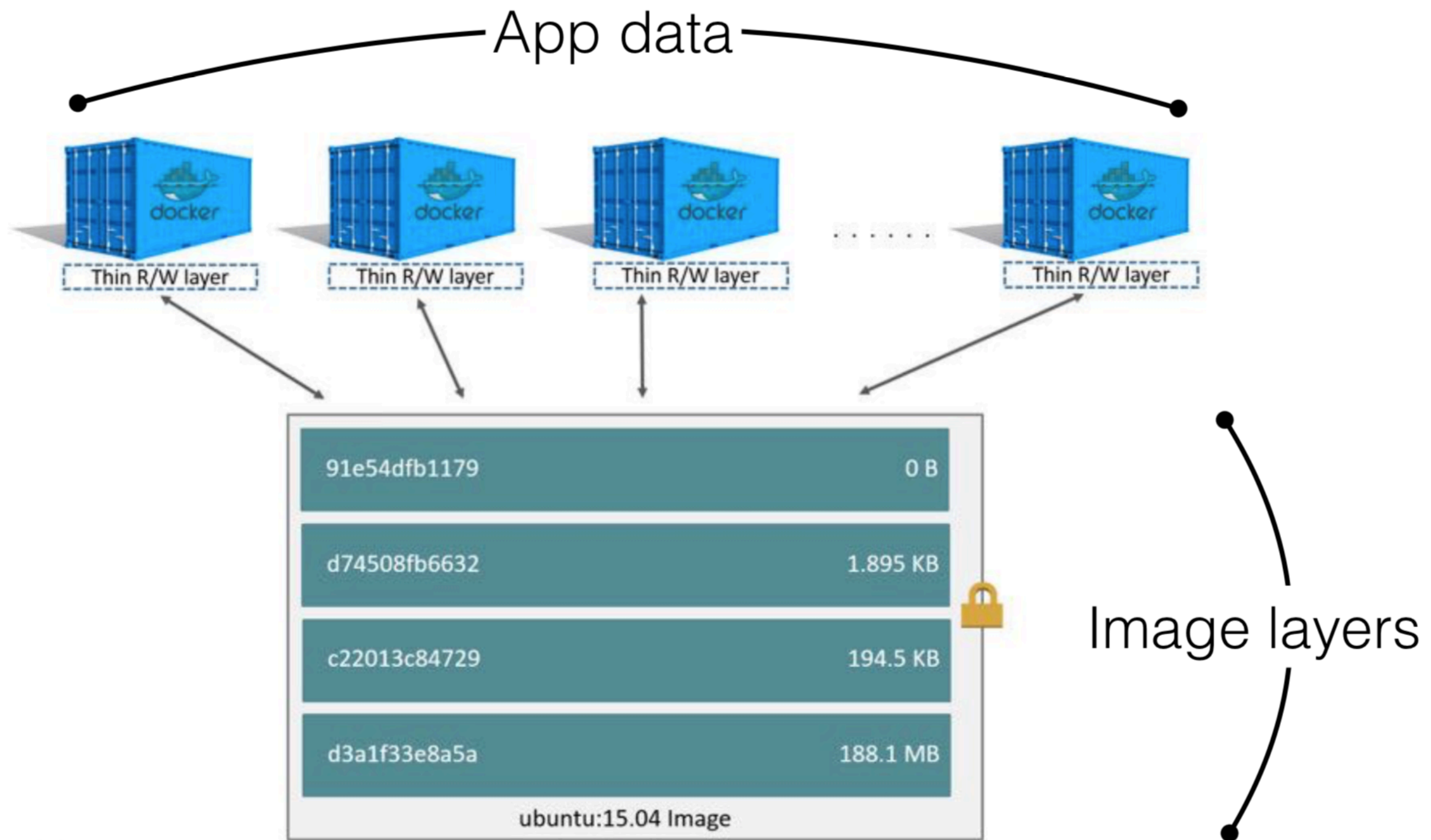
USER NS

- Позволяет мапить GID/UID
- Можно смэпить UID 0 (root) в нерутовый UID в хостовом NS

CGroups

- Управляет ресурсами для процесса
- Добавляет overhead
 - Или нет
- Определение CPU ограничений должно поддерживаться средствами приложения

Image (CoW)



Dockerfile

- Отключение кэшей
- Multi-stage сборка
- Установка и удаление в одном шаге
- Порядок выполнения шагов
- COPY vs ADD / ENTRYPOINT vs CMD

Dockerfile

```
FROM ruby:2.5.1-alpine3.7

WORKDIR /app
ENV HOME /app

COPY Gemfile Gemfile.lock ./
COPY . .

RUN addgroup -S www-data && \
    adduser -S www-data -G www-data && \
    apk update && \
    apk upgrade && \
    apk add nginx nodejs dcron tzdata postgresql-dev libxslt-dev build-base libxml2-dev && \
    bundle config build.nokogiri --use-system-libraries && \
    bundle install --without development && \
    RAILS_ENV=production bundle exec rake assets:precompile && \
    ln -sf /dev/stdout /var/log/nginx/access.log && \
    ln -sf /dev/stderr /var/log/nginx/error.log && \

COPY config/nginx.conf /etc/nginx/nginx.conf

RUN chown -R www-data:www-data /app

EXPOSE 80

CMD ["/bin/docker-entrypoint.sh"]
```


Dockerfile

```
FROM ruby:2.5.1-alpine3.7

ENV BUILD_PACKAGES nginx nodejs dcron tzdata postgresql-dev libxslt-dev
ENV BUILD_TMP_PACKAGES build-base libxml2-dev

WORKDIR /app
ENV HOME /app

RUN set -ex && \
    addgroup -S www-data && \
    adduser -S www-data -G www-data && \
    apk --update add --no-cache $BUILD_PACKAGES && \
    ln -sf /dev/stdout /var/log/nginx/access.log && \
    ln -sf /dev/stderr /var/log/nginx/error.log && \
    rm -rf /var/cache/apk/*

COPY config/nginx.conf /etc/nginx/nginx.conf

COPY Gemfile Gemfile.lock ./
RUN set -ex && \
    apk add --no-cache --virtual build_deps $BUILD_TMP_PACKAGES && \
    bundle config build.nokogiri --use-system-libraries && \
    bundle install --clean --no-cache --without development && \
    rm -rf /var/cache/apk/* && \
    apk del build_deps

COPY . .
RUN RAILS_ENV=production bundle exec rake assets:precompile

RUN chown -R www-data:www-data /app/public

EXPOSE 80

CMD ["./bin/docker-entrypoint.sh"]
```

Docker-compose

```
docker-compose -p "$CI_PROJECT_NAME"_"$CI_PIPELINE_ID" up --abort-on-container-exit --exit-code-from app --quiet-pull
```

```
version: '2.1'
```

```
services:
```

```
  app:
```

```
    image: ${CI_REGISTRY}/${CI_PROJECT_NAMESPACE}/${CI_PROJECT_NAME}:${CI_COMMIT_REF_SLUG}.${CI_PIPELINE_ID}
```

```
    environment:
```

```
    .....
```

```
      command: /bin/bash -c 'rake parallel:setup && rake parallel:spec'
```

```
      depends_on:
```

```
        db:
```

```
          condition: service_healthy
```

```
        redis:
```

```
          condition: service_healthy
```

```
  db:
```

```
    image: postgres:10.3
```

```
    environment:
```

```
    .....
```

```
      command: -c fsync=off -c full_page_writes=off
```

```
      healthcheck:
```

```
        test: ["CMD", "pg_isready", "-U", "postgres"]
```

```
      logging:
```

```
        driver: none
```

```
  redis:
```

```
    image: redis
```

```
    healthcheck:
```

```
      test: ["CMD", "redis-cli", "ping"]
```

```
    logging:
```

```
      driver: none
```

Docker для Kubernetes

- Отключение iptables
- Log driver
- DNS
- Сетевые опции

СЛЁРМ

+



Southbridge

Перерыв