

Docker:

непрерывная интеграция
и развертывание приложений (CI/CD)

Спикер: Александр Швалов

Что такое CI/CD?

- `Continuous integration / continuous deployment (delivery)` – непрерывная интеграция и развертывание.

Что такое CI/CD?

- **Continuous integration / continuous deployment (delivery)** – непрерывная интеграция и развертывание.
- Концепция доставки кода как конвейер (pipeline – букв. «трубопровод»).

Что такое CI/CD?

- **Continuous integration / continuous deployment (delivery)** – непрерывная интеграция и развертывание.
- Концепция доставки кода как конвейер (pipeline – букв. «трубопровод»).
- ПО, которое управляет конвейером.

Что такое CI/CD?

- **Continuous integration / continuous deployment (delivery)** – непрерывная интеграция и развертывание.
- Концепция доставки кода как конвейер (pipeline – букв. «трубопровод»).
- ПО, которое управляет конвейером.
- Инструкция для работы этого конвейера.

Зачем нам CI/CD?

1. Позволяет значительно упростить и ускорить процесс доставки новых изменений в тестовую среду и в продакшен, автоматизирует ручной труд.

Зачем нам CI/CD?

1. Позволяет значительно упростить и ускорить процесс доставки новых изменений в тестовую среду и в продакшен, автоматизирует ручной труд.
2. При использовании тестирования на этапе CI/CD мы можем отсеять некоторый процент ошибок на пути в продакшен.

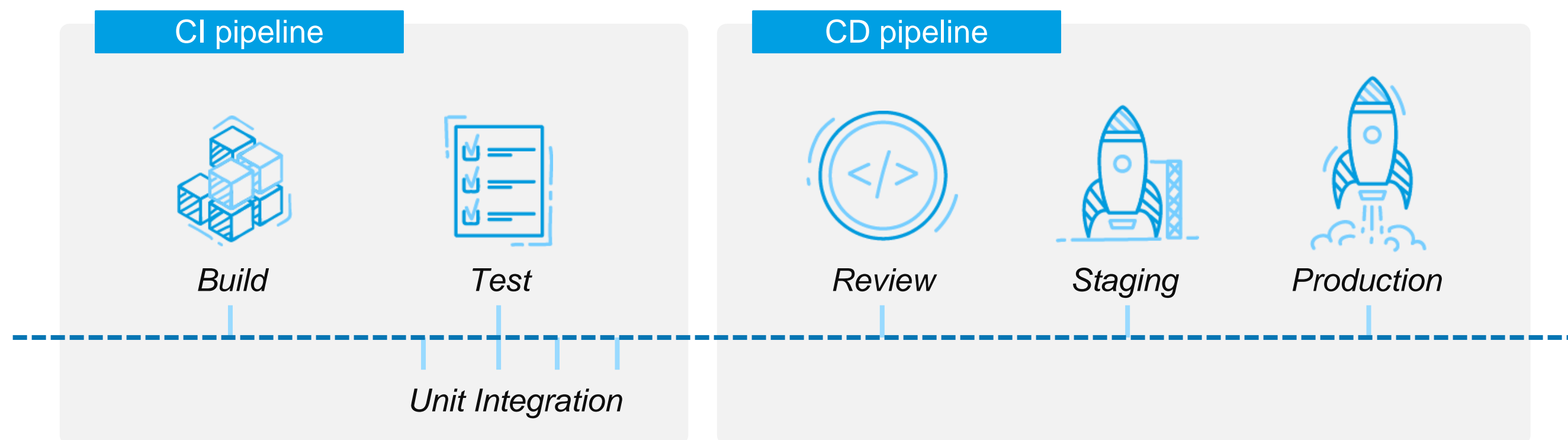
Зачем нам CI/CD?

1. Позволяет значительно упростить и ускорить процесс доставки новых изменений в тестовую среду и в продакшен, автоматизирует ручной труд.
2. При использовании тестирования на этапе CI/CD мы можем отсеять некоторый процент ошибок на пути в продакшен.
3. Свой registry может хранить историю сборок за какой-то промежуток времени, при необходимости можно откатиться назад.

Зачем нам CI/CD?

1. Позволяет значительно упростить и ускорить процесс доставки новых изменений в тестовую среду и в продакшен, автоматизирует ручной труд.
2. При использовании тестирования на этапе CI/CD мы можем отсеять некоторый процент ошибок на пути в продакшен.
3. Свой registry может хранить историю сборок за какой-то промежуток времени, при необходимости можно откатиться назад.
4. Неотъемлемая часть популярной концепции IaC – Infrastructure-as-a-code, «инфраструктура как код».

CI/CD – делаем работу удобной



Используемое ПО

Наш инструмент – Gitlab:

- Система контроля версий



Используемое ПО

Наш инструмент – Gitlab:

- Система контроля версий
- Встроенный Docker registry



Используемое ПО

Наш инструмент – Gitlab:

- Система контроля версий
- Встроенный Docker registry
- Разделение инстансов - gitlab и gitlab-runner



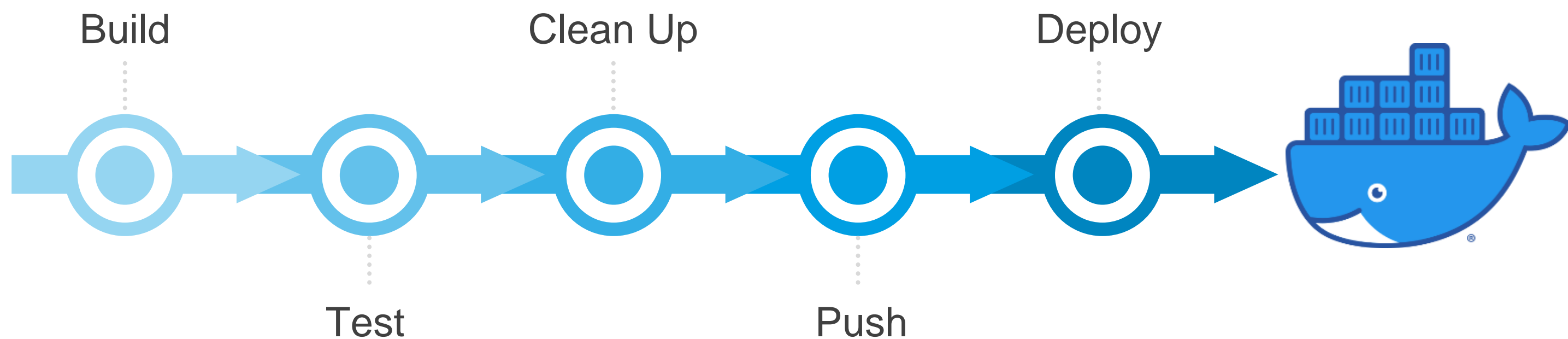
Используемое ПО

Наш инструмент – Gitlab:

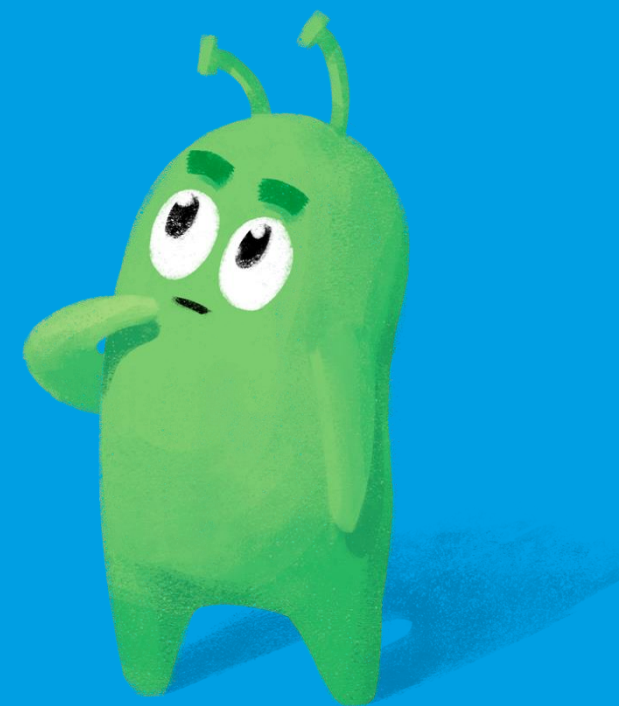
- Система контроля версий
- Встроенный Docker registry
- Разделение инстансов - gitlab и gitlab-runner
- Плюс Docker и Docker Compose



Этапы CI/CD на нашем примере



Хватит теории,
давайте перейдём к **практике!**



Проблемы нашего примера CI/CD

- Как бесшовно обновить?

Проблемы нашего примера CI/CD

- Как бесшовно обновить?
- Как откатить неудачный релиз?

Проблемы нашего примера CI/CD

- Как бесшовно обновить?
- Как откатить неудачный релиз?

ВЫХОД: использовать оркестратор

СЛЁРМ



slurm.io



Southbridge



southbridge.io