



Докеризация и CI/CD

Вечерняя школа Kubernetes

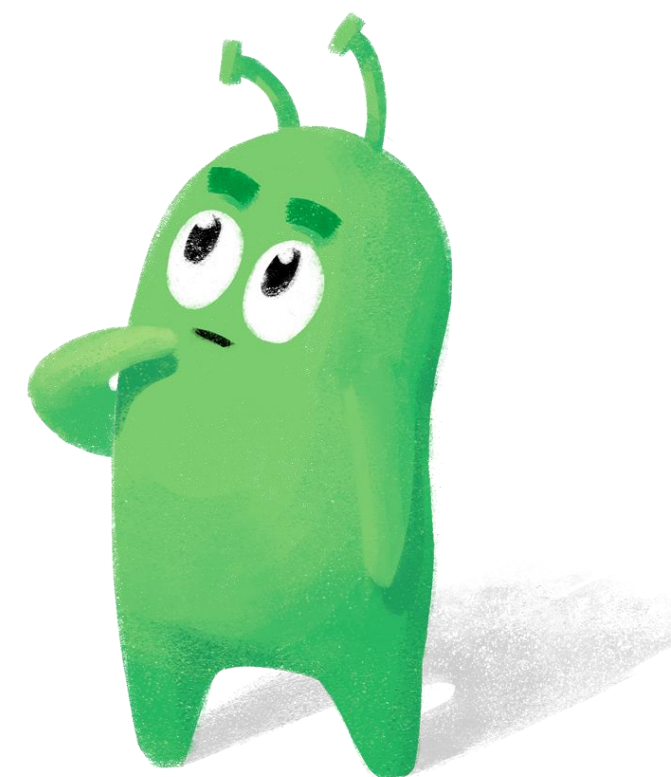
Александр Швалов

Работаю с Linux более 6 лет,
Инженер Southbridge с 2017,
Certified Kubernetes Administrator.



Краткий обзор

1. Докеризация и возможные проблемы
2. Ручное прохождение цикла разработки
3. Переход на CI/CD
4. Используемые инструменты
5. Обзор стадий пайплайна
6. Демонстрация работы



Докеризация и CI/CD

1. Докеризация решает проблемы повторяемости.

Докеризация и CI/CD

1. Докеризация решает проблемы повторяемости.
2. Позволяет значительно упростить и ускорить процесс доставки новых изменений в тестовую среду и в продакшен.

Докеризация и CI/CD

1. Докеризация решает проблемы повторяемости.
2. Позволяет значительно упростить и ускорить процесс доставки новых изменений в тестовую среду и в продакшен.
3. При использовании автоматизированного тестирования позволяет отсеять некоторый процент ошибок на пути в продакшен.

Докеризация и CI/CD

1. Докеризация решает проблемы повторяемости.
2. Позволяет значительно упростить и ускорить процесс доставки новых изменений в тестовую среду и в продакшен.
3. При использовании автоматизированного тестирования позволяет отсеять некоторый процент ошибок на пути в продакшен.
4. Позволяет хранить историю сборок за какой-то промежуток времени, при необходимости можно откатиться назад.

Возможные проблемы:

1. ваше приложение читает настройки из файлов;

Возможные проблемы:

1. ваше приложение читает настройки из файлов;
2. приложение пишет логи только в файл;

Возможные проблемы:

1. ваше приложение читает настройки из файлов;
2. приложение пишет логи только в файл;
3. приложение сохраняет важные данные в файлы во время работы;

Возможные проблемы:

1. ваше приложение читает настройки из файлов;
2. приложение пишет логи только в файл;
3. приложение сохраняет важные данные в файлы во время работы;
4. приложению нужны права root;

Возможные проблемы:

1. ваше приложение читает настройки из файлов;
2. приложение пишет логи только в файл;
3. приложение сохраняет важные данные в файлы во время работы;
4. приложению нужны права root;
5. нужно четко понимать, какие версии ПО нужны для запуска приложения, и указывать это в Dockerfile;

Подробности по работе Docker смотрите в лекциях Слёрм.

Возможные проблемы:

1. ваше приложение читает настройки из файлов;
2. приложение пишет логи только в файл;
3. приложение сохраняет важные данные в файлы во время работы;
4. приложению нужны права root;
5. нужно четко понимать, какие версии ПО нужны для запуска приложения, и указывать это в Dockerfile;
6. нужно хотя бы примерно понимать, какие аппаратные ресурсы нужны приложению для работы.

Подробности по работе Docker смотрите в лекциях Слёрм.

Наше приложение

- По конвейеру будем запускать старую версию <https://xpaste.pro>
- Ruby-приложение с рита+nginx и базой данных.

Ручной труд

- `docker build`

Ручной труд

- `docker build`
- `docker-compose up`

Ручной труд

- `docker build`
- `docker-compose up`
- `docker-compose down`

Ручной труд

- `docker build`
- `docker-compose up`
- `docker-compose down`
- `docker login`

Ручной труд

- `docker build`
- `docker-compose up`
- `docker-compose down`
- `docker login`
- `docker push`

Ручной труд

- `docker build`
- `docker-compose up`
- `docker-compose down`
- `docker login`
- `docker push`
- `helm upgrade --install`



Автоматизация труда

Наш инструмент – Gitlab:

- Система контроля версий
- Встроенный Docker registry
- Разделение инстансов - gitlab и gitlab-runner
- Поддержка Kubernetes и Auto DevOps



Автоматизация труда

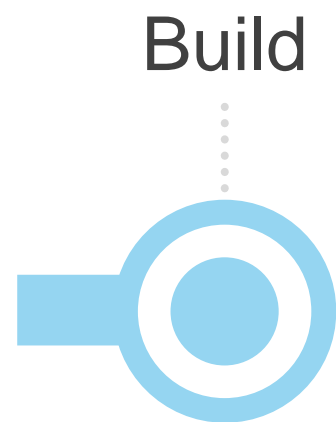
Наш инструмент – Gitlab:

- Система контроля версий
- Встроенный Docker registry
- Разделение инстансов - gitlab и gitlab-runner
- Поддержка Kubernetes и Auto DevOps

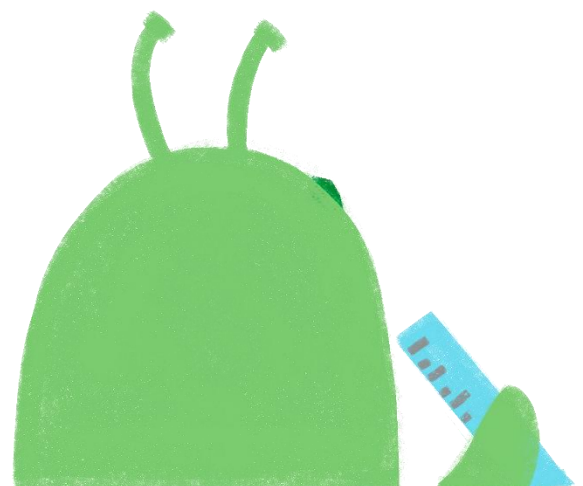
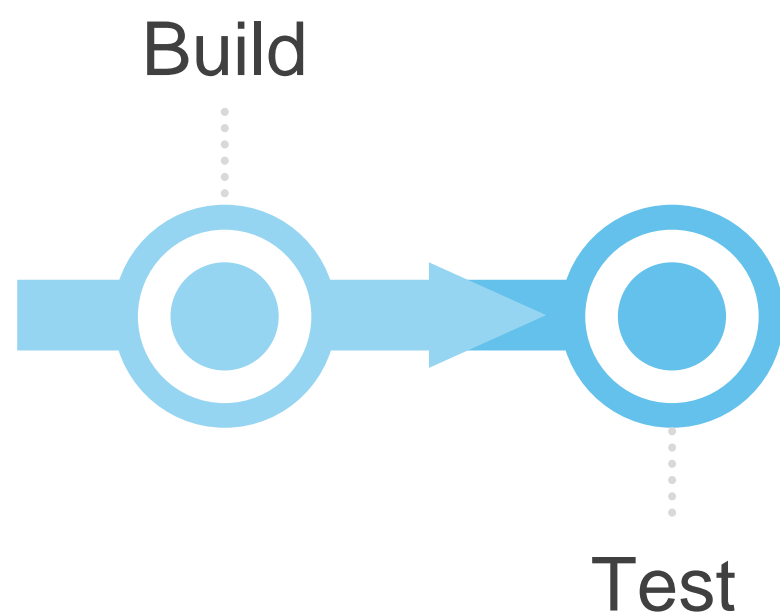
Альтернативы:

- Jenkins
- Teamcity
- Bamboo
- CircleCI
- ...

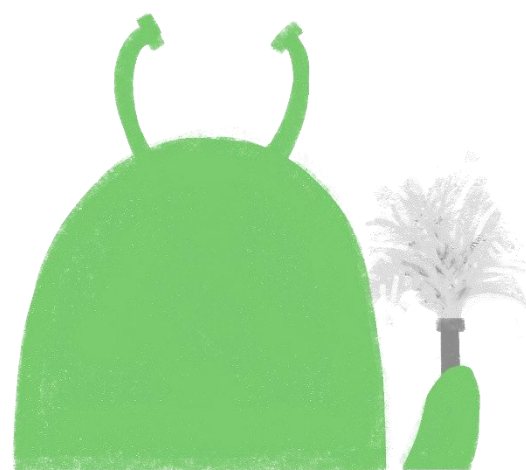
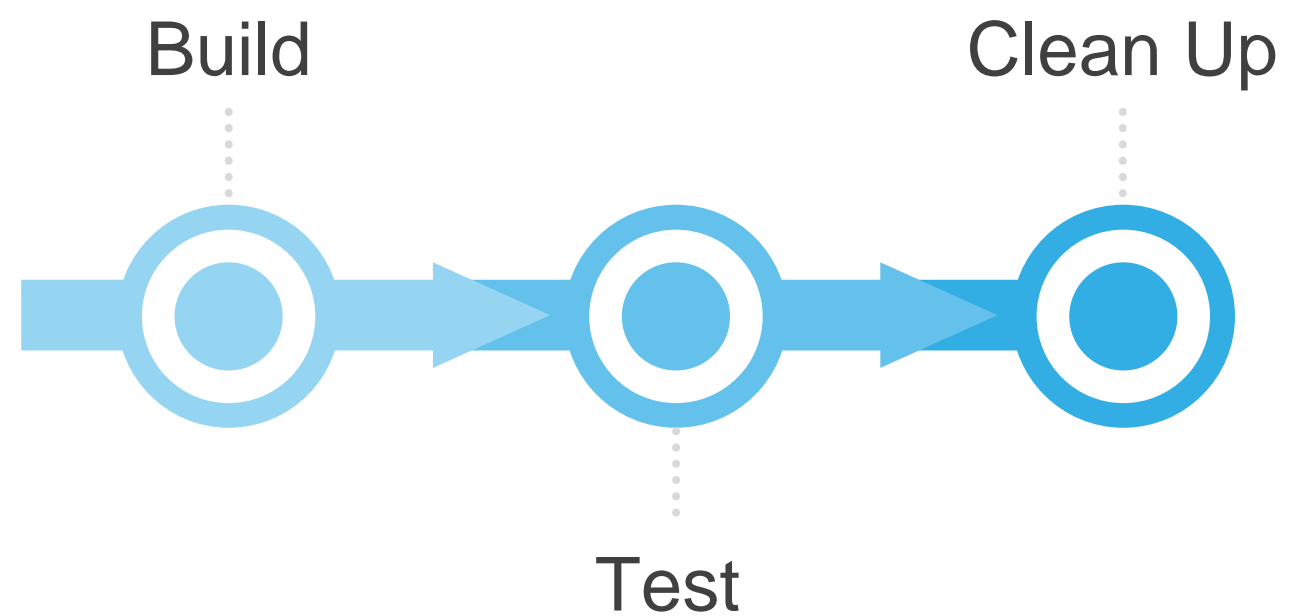
Этапы CI/CD



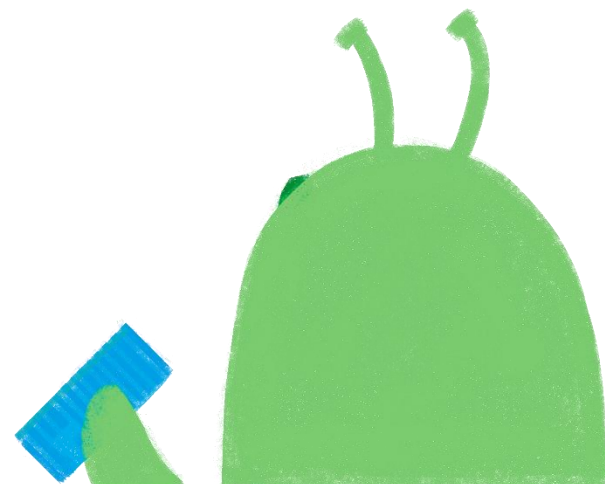
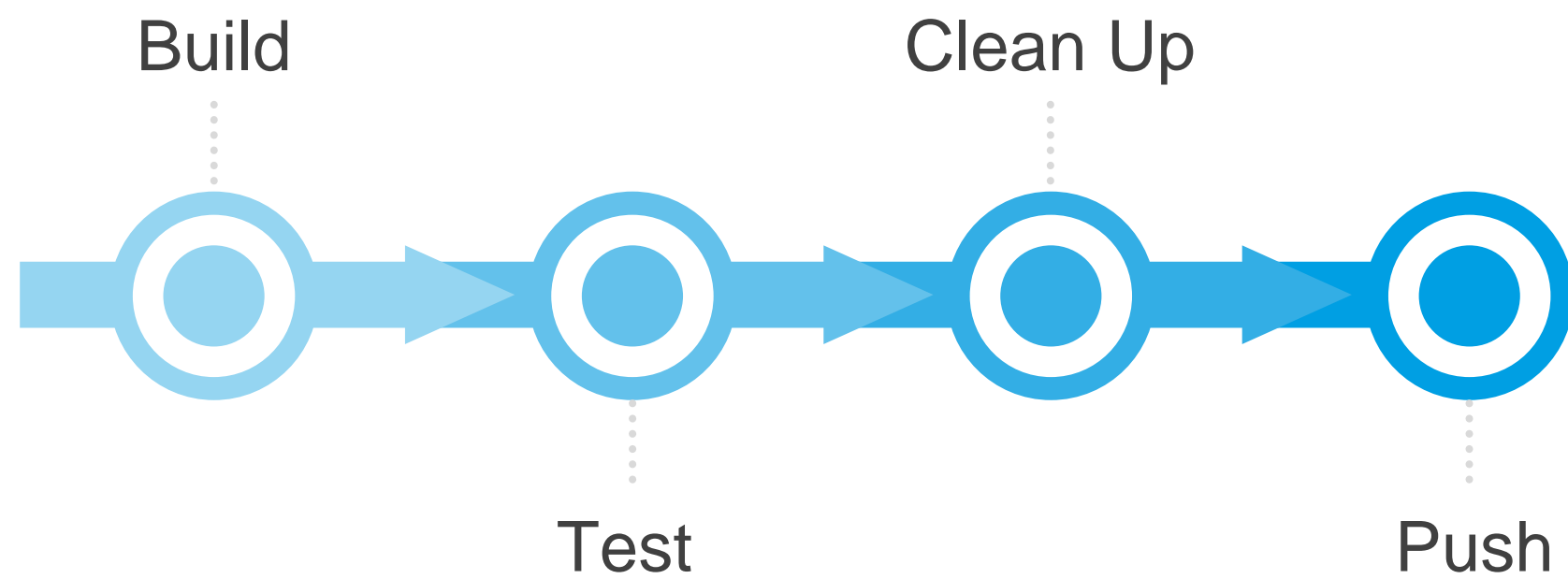
Этапы CI/CD



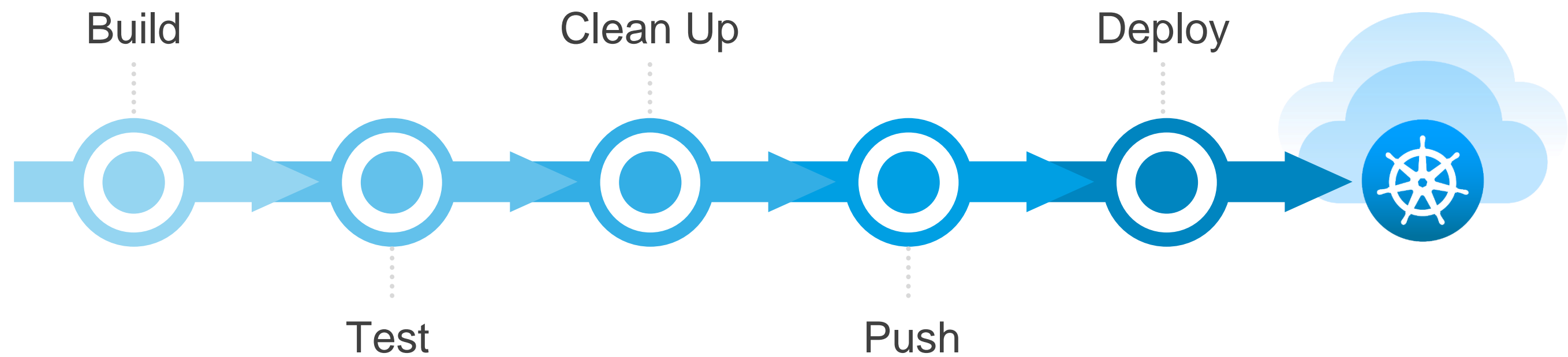
Этапы CI/CD



Этапы CI/CD



Этапы CI/CD



Хватит теории,
давайте перейдём к практике!

СЛЁРМ



slurm.io



Southbridge



southbridge.io