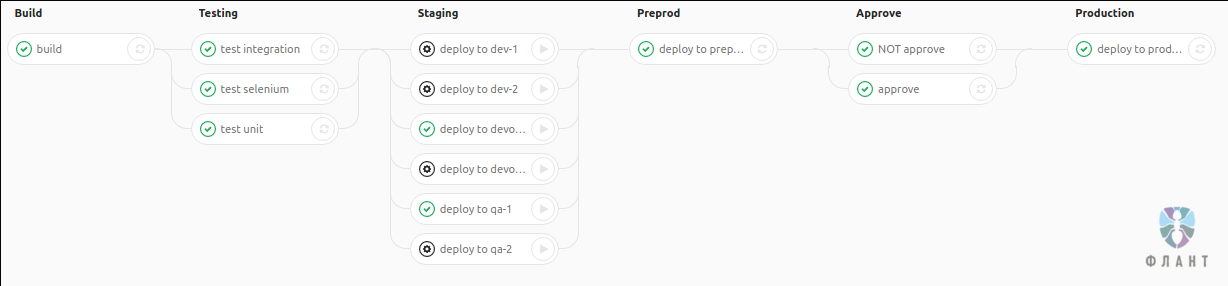
# Возможности построение CI/CD на GitLab

# “Если вы сталкивались с CI-системами ранее, то понятие пайплайна вам знакомо — это последовательность выполнения стадий *(здесь и далее в статье для термина stage использую перевод «стадия»)*, каждая из которых включает несколько задач *(здесь и далее в статье job — «задача»)*. От момента внесения изменений в код до выката в production приложение по очереди оказывается в руках разных команд — подобному тому, как это происходит на конвейере. Отсюда и возник термин pipeline *(«конвейер» — один из вариантов дословного перевода)*. В нашем случае это выглядит так:

Краткие пояснения по используемым стадиям:



* **build** — сборка приложения;
* **testing** — автотесты;
* **staging** — развёртывание приложения для разработчиков, DevOps, тестировщиков;
* **pre-production** — развёртывание в «предварительный production» для команды тестировщиков;
* **approve** — «предохранитель», благодаря которому релиз-инженер заказчика может отказать в развёртывании на production определённого тега;
* **production** — развёртывание на production.

# *Примечание: Нет ничего совершенно универсального, поэтому для вашего конкретного случая этот пайплайн скорее всего не подходит: он либо избыточен, либо прост. Однако цель статьи не описывать единственно верный вариант, подходящий каждому. Цель — рассказать на примере о том, как можно работать в GitLab CI нескольким командам и какие возможности есть, чтобы реализовать такой пайплайн. На основе этой информации можно разработать свою собственную конфигурацию GitLab CI.”*

# <https://habr.com/ru/company/flant/blog/332712/> - GitLab CI для непрерывной интеграции и доставки в production.

# <https://medium.com/@cakeinpanic/основы-ci-на-gitlab-19489cc79fe8> - Основы CI/CD на gitlab

**Различие построение CI/CD на GitLab и на GitHub**

# Одно из главных отличий между ними заключается в том, что GitLab предлагает собственный встроенный встроенный модуль Continuous Integration / Delivery (CI / CD), который вам не нужно устанавливать отдельно. Это поможет командам сократить ошибки в коде и обеспечить более быстрые результаты, придерживаясь стандартов качества вашей команды. Напротив, он не интегрируется с GitHub; на самом деле, для этого есть несколько инструментов.

# https://ru.esdifferent.com/difference-between-github-and-gitlab