



# Разработка инфраструктуры программного обеспечения

Разработка CI/CD (Continuous integration / Continuous deployment)  
Часть 2

Лаборатория RISC-V технологий,  
2025 г.

# Использование actions

- Довольно часто возникает ситуация, когда надо выполнить в своем workflow некоторые стандартные популярные действия — например, выкачать репозиторий



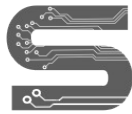
# Использование actions




- Довольно часто возникает ситуация, когда надо выполнить в своем workflow некоторые стандартные популярные действия — например, выкачать репозиторий
- В GitHub actions для этого используются, well, **actions**. По сути это некоторые библиотеки, которые предоставляют стандартный функционал (<https://github.com/actions>)



# Использование actions






## GitHub Actions


Automate your GitHub workflows

Verified




 6.8k followers   <https://github.com/features/actions>


---

Pinned




 **starter-workflows** Public


Accelerating new GitHub Actions workflows

 TypeScript    9.8k    5.7k




 **toolkit** Public


The GitHub ToolKit for developing GitHub Actions.

 TypeScript    5.1k    1.5k




 **setup-node** Public


Set up your GitHub Actions workflow with a specific version of node.js

 TypeScript    4.1k    1.3k




 **javascript-action** Public template


Create a JavaScript Action with tests, linting, workflow, publishing, and versioning

 JavaScript    1k    389




 **typescript-action** Public template

Create a TypeScript Action with tests, linting, workflow, publishing, and versioning

 TypeScript    2.1k    508

 **labeler** Public

An action for automatically labelling pull requests

 TypeScript    2.1k    431

# Использование actions



- Довольно часто возникает ситуация, когда надо выполнить в своем workflow некоторые стандартные популярные действия — например, выкачать репозиторий
- В GitHub actions для этого используются, well, **actions**. По сути это некоторые библиотеки, которые предоставляют стандартный функционал (<https://github.com/actions>)



## Популярные actions

- **checkout** (клонировать репозиторий)
- **deploy-pages** (создать страничку с контентом на GitHub Pages)
- **container-action** (экшн для создания контейнеров в пайплайне)
- **cache** (кеширование зависимостей и артефактов для ускорения пайпалайнов)
- **Actions** можно использовать следующими способами:

```
jobs:  
  main:  
    runs-on: ubuntu-24.04  
    steps:  
      - uses: actions/checkout@v4
```

```
jobs:  
  main:  
    runs-on: ubuntu-24.04  
    steps:  
      - name: Clone repository  
        uses: actions/checkout@v4  
        with:  
          ref: `d8390cffb8e2e6974cfecc84b0e9cdfbecb1482e`
```



# Где запускается мой пайплайн ??



- Как мы уже обсуждали, ваш ворклофу запускается на **раннерах**

# Где запускается мой пайплайн ??



- Как мы уже обсуждали, ваш ворклофу запускается на **раннерах**
- Раннер — это машина, которая сконфигурирована с помощью **Github Actions** инфраструктуры на запуск воркфлоу

# Где запускается мой пайплайн ??



- Как мы уже обсуждали, ваш ворклофу запускается на **раннерах**
- Раннер — это машина, которая сконфигурирована с помощью **Github Actions** инфраструктуры на запуск воркфлоу
- **Github actions** предоставляют бесплатных раннеров в ограниченном объеме запусков на трех платформах **Ubuntu Linux, Microsoft Windows и macOS**



# Где запускается мой пайплайн ??



Virtual Machine	Processor (CPU)	Memory (RAM)	Storage (SSD)	Architecture	Workflow label
Linux	4	16 GB	14 GB	x64	<a href="#">ubuntu-latest</a> , <a href="#">ubuntu-24.04</a> , <a href="#">ubuntu-22.04</a> , <a href="#">ubuntu-20.04</a>
Windows	4	16 GB	14 GB	x64	<a href="#">windows-latest</a> , <a href="#">windows-2025</a> [Public preview], <a href="#">windows-2022</a> , <a href="#">windows-2019</a>
Linux [Public preview]	4	16 GB	14 GB	arm64	<a href="#">ubuntu-24.04-arm</a> , <a href="#">ubuntu-22.04-arm</a>
macOS	4	14 GB	14 GB	Intel	<a href="#">macos-13</a>
macOS	3 (M1)	7 GB	14 GB	arm64	<a href="#">macos-latest</a> , <a href="#">macos-14</a> , <a href="#">macos-15</a> [Public preview]

# Где запускается мой пайплайн ??



- Как мы уже обсуждали, ваш ворклофу запускается на **раннерах**
- Раннер — это машина, которая сконфигурирована с помощью **Github Actions** инфраструктуры на запуск воркфлоу
- **Github actions** предоставляют бесплатных раннеров в ограниченном объеме запусков на трех платформах **Ubuntu Linux, Microsoft Windows и macOS**
- Что делать если не хватает?

# Где запускается мой пайплайн ??



- Как мы уже обсуждали, ваш воркфлоу запускается на **раннерах**
- Раннер — это машина, которая сконфигурирована с помощью **Github Actions** инфраструктуры на запуск воркфлоу
- **Github actions** предоставляют бесплатных раннеров в ограниченном объеме запусков на трех платформах **Ubuntu Linux, Microsoft Windows и macOS**
- Что делать если не хватает?
- Что делать, если в принципе работаете над коммерческим проектом и не хотите, чтобы ваши воркфлоу гонялись на каких-то публичных машинах?

# Selfhosted runners



- **Selfhosted runners** — это машины, сконфигурированные под запуски workflow, при это находящиеся в **вашей** инфраструктуре (например, у вас в офисе, дома, в личном датацентре или серверной комнате компании)

# Selfhosted runners



- **Selfhosted runners** — это машины, сконфигурированные под запуски workflow, при это находящиеся в **вашей** инфраструктуре (например, у вас в офисе, дома, в личном датацентре или серверной комнате компании)
- На таких раннерах можно ставить любую **OS**, сами раннера могут быть произвольной мощности и только для вашего пользования

# Selfhosted runners



- **Selfhosted runners** — это машины, сконфигурированные под запуски workflow, при это находящиеся в **вашей** инфраструктуре (например, у вас в офисе, дома, в личном датацентре или серверной комнате компании)
- На таких раннерах можно ставить любую **OS**, сами раннера могут быть произвольной мощности и только для вашего пользования
- Является ли это стандартом индустрии?



# Selfhosted runners



- **Selfhosted runners** — это машины, сконфигурированные под запуски workflow, при это находящиеся в **вашей** инфраструктуре (например, у вас в офисе, дома, в личном датацентре или серверной комнате компании)
- На таких раннерах можно ставить любую **OS**, сами раннера могут быть произвольной мощности и только для вашего пользования
- Является ли это стандартом индустрии?
- Вообще-то да, но вообще-то и нет
- Как правило, selfhosted runners удовлетворяют требованиям высокой мощности и неограниченного личного использования



# Selfhosted runners



- **Selfhosted runners** — это машины, сконфигурированные под запуски workflow, при это находящиеся в **вашей** инфраструктуре (например, у вас в офисе, дома, в личном датацентре или серверной комнате компании)
- На таких раннерах можно ставить любую **OS**, сами раннера могут быть произвольной мощности и только для вашего пользования
- Является ли это стандартом индустрии?
- Вообще-то да, но вообще-то и нет
- Как правило, selfhosted runners удовлетворяют требованиям высокой мощности и неограниченного личного использования
- При этом сам проект, к которому эти раннеры подключены, все равно находится на гитхабе ( пусть даже и приватном репозитории )
- Гораздо чаще хотят иметь не только собственных раннеров, но и **собственный CI сервис** (например, GitLab CI)
- Инструкция по настройке для самостоятельной практики в качестве дополнительного (непростого) задания:  
<https://docs.github.com/en/actions/hosting-your-own-runners/managing-self-hosted-runners/adding-self-hosted-runners>



# Selfhosted runners



- **Selfhosted runners** — это машины, сконфигурированные под запуски workflow, при это находящиеся в **вашей** инфраструктуре (например, у вас в офисе, дома, в личном датацентре или серверной комнате компании)
- На таких раннерах можно ставить любую **OS**, сами раннера могут быть произвольной мощности и только для вашего пользования
- Является ли это стандартом индустрии?
- Вообще-то да, но вообще-то и нет
- Как правило, selfhosted runners удовлетворяют требованиям высокой мощности и неограниченного личного использования
- При этом сам проект, к которому эти раннеры подключены, все равно находится на гитхабе ( пусть даже и приватном репозитории )
- Гораздо чаще хотят иметь не только собственных раннеров, но и **собственный CI сервис** (например, гитлаб)



# Артефакты и релизы



- Теперь, когда Кирилл и Миша настроили базовый CI, они смогли довести проект до более-менее рабочего состояния



# Артефакты и релизы



- Теперь, когда Кирилл и Миша настроили базовый CI, они смогли довести проект до более-менее рабочего состояния
- Клиент хочет получить функционал! 

---



# Артефакты и релизы



- Теперь, когда Кирилл и Миша настроили базовый CI, они смогли довести проект до более-менее рабочего состояния
- Клиент хочет получить функционал!
- У вас есть репозиторий с исходным кодом. Как бы передали клиенту его? Ссылка на закрытый репозиторий, архивчик с бинарями?

# Артефакты и релизы



- Теперь, когда Кирилл и Миша настроили базовый CI, они смогли довести проект до более-менее рабочего состояния
- Клиент хочет получить функционал!
- У вас есть репозиторий с исходным кодом. Как бы передали клиенту его? Ссылка на закрытый репозиторий, архивчик с бинарями?
- Вы бы сделали это по почте? Или, может быть, отправили бы zip папку в телеграмме?

# Артефакты и релизы



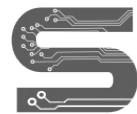
- Теперь, когда Кирилл и Миша настроили базовый CI, они смогли довести проект до более-менее рабочего состояния
  - Клиент хочет получить функционал!
  - У вас есть репозиторий с исходным кодом. Как бы передали клиенту его? Ссылка на закрытый репозиторий, архивчик с бинарями?
  - Вы бы сделали это по почте? Или, может быть, отправили бы zip папочку в телеграмме?
- 
- Представьте теперь, что в любом выбранном вами способе вам необходимо каждый раз вручную с выбранного так называемого релизного коммита, собирать, например, архивчик с проектом
  - Теперь представьте себе другую задачу — вы хотите, чтобы **каждый** коммит в ваш репозиторий тестировался в окружении заказчика / кастомера / в том репозитории, кто **зависит от вас в каком-то виде**

# Артефакты и релизы





- Теперь, когда Кирилл и Миша настроили базовый CI, они смогли довести проект до более-менее рабочего состояния
  - Клиент хочет получить функционал!
  - У вас есть репозиторий с исходным кодом. Как бы передали клиенту его? Ссылка на закрытый репозиторий, архивчик с бинарями?
  - Вы бы сделали это по почте? Или, может быть, отправили бы zip папку в телеграмме?
- 
- Представьте теперь, что в любом выбранном вами способе вам необходимо каждый раз вручную с выбранного так называемого релизного коммита, собирать, например, архивчик с проектом
  - Теперь представьте себе другую задачу — вы хотите, чтобы **каждый** коммит в ваш репозиторий тестировался в окружении заказчика / кастомера / в том репозитории, кто **зависит от вас в каком-то виде**

# Артефакты и релизы



- **Артефакт** — продукт работы вашего репозитория. Будет использоваться теми, кто от вас зависит.

Artifacts		
Produced during runtime		
Name		Size
	artifact	29 MB
		



# Артефакты и релизы



- **Артефакт** — продукт работы вашего репозитория. Будет использоваться теми, кто от вас зависит.
- Что обычно представляет собой — как правило, архив с бинарниками. В некоторых ситуациях может быть что-то еще

# Артефакты и релизы



- **Артефакт** — продукт работы вашего репозитория. Будет использоваться теми, кто от вас зависит.
- Что обычно представляет собой — как правило, архив с бинарниками. В некоторых ситуациях может быть что-то еще
- Артефакты мы хотим собирать и выгружать куда-нибудь в место хранения данных артефактов. Это называется процесс **деплойа**.

# Артефакты и релизы

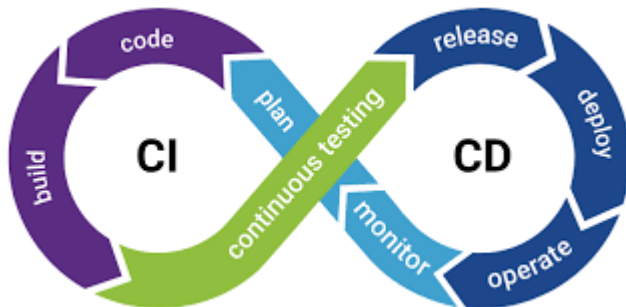


- **Артефакт** — продукт работы вашего репозитория. Будет использоваться теми, кто от вас зависит.
- Что обычно представляет собой — как правило, архив с бинарниками. В некоторых ситуациях может быть что-то еще
- Артефакты мы хотим собирать и выгружать куда-нибудь в место хранения данных артефактов. Это называется процесс **деплойа**.
- Процесс запуска workflow в вашем репозитории, проверка целостности и валидности его состояния на **постоянной автоматической основе** — **Continious Integration**
- Процесс создания с вашего репозитория **артефакта** и загрузка его в некоторое хранилище на **постоянной автоматической основе** — **Continious Deployment**

# Артефакты и релизы



- **Артефакт** — продукт работы вашего репозитория. Будет использоваться теми, кто от вас зависит.
- Что обычно представляет собой — как правило, архив с бинарниками. В некоторых ситуациях может быть что-то еще
- Артефакты мы хотим собирать и выгружать куда-нибудь в место хранения данных артефактов. Это называется процесс **деплойа**.
- Процесс запуска workflow в вашем репозитории, проверка целостности и валидности его состояния на **постоянной автоматической основе** — **Continuous Integration**
- Процесс создания с вашего репозитория **артефакта** и загрузка его в некоторое хранилище на **постоянной автоматической основе** — **Continuous Deployment**
- Вместе - **CI/CD**



# Артефакты и релизы



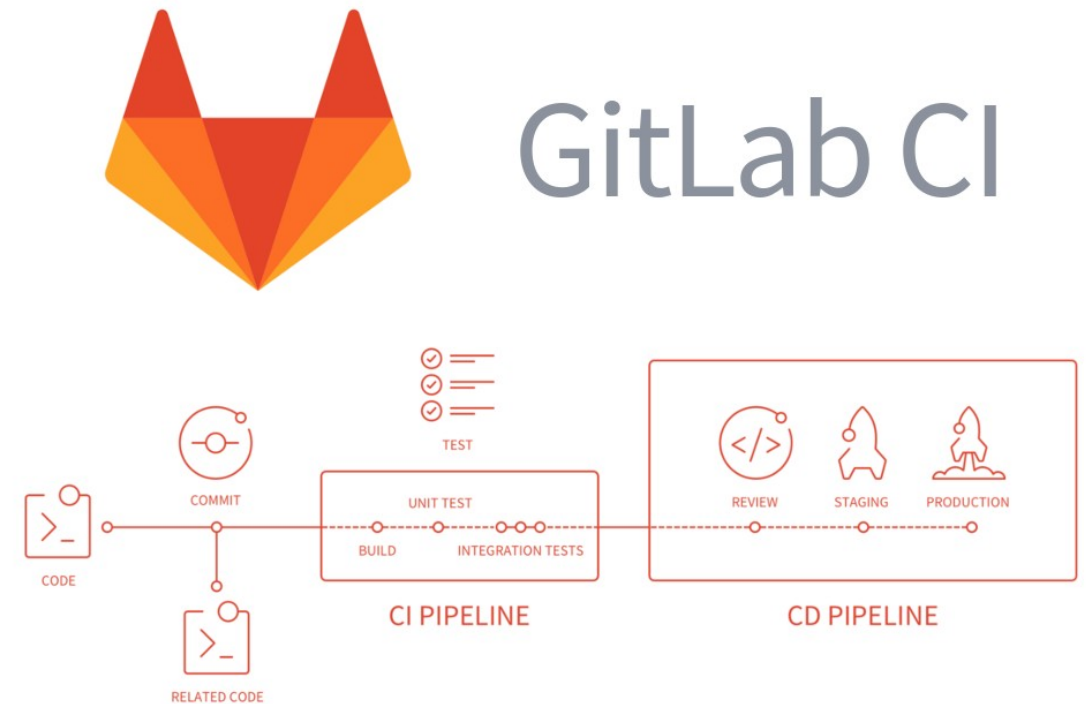
- В GitHub actions есть механизм для создания артефактов ваших проектов и выгрузки их в некоторые реестр
- Для простых релизов можно просто выгружать артефакты на тот же GitHub — в папку releases
- Для этого используется **actions/upload-artifact@v4**



# GitLab CI



- Рассмотрим вкратце другой CI сервис — **GitLab CI** и его отличия от **GitHub Actions**



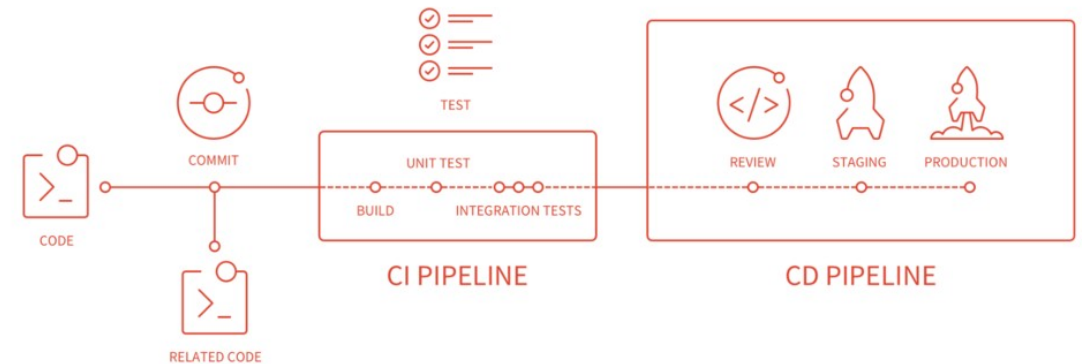
# GitLab CI



- Рассмотрим вкратце другой CI сервис — **GitLab CI** и его отличия от **GitHub Actions**
- Сам по себе **GitLab** такое же облачное хранилище репозитория, как и **GitHub**



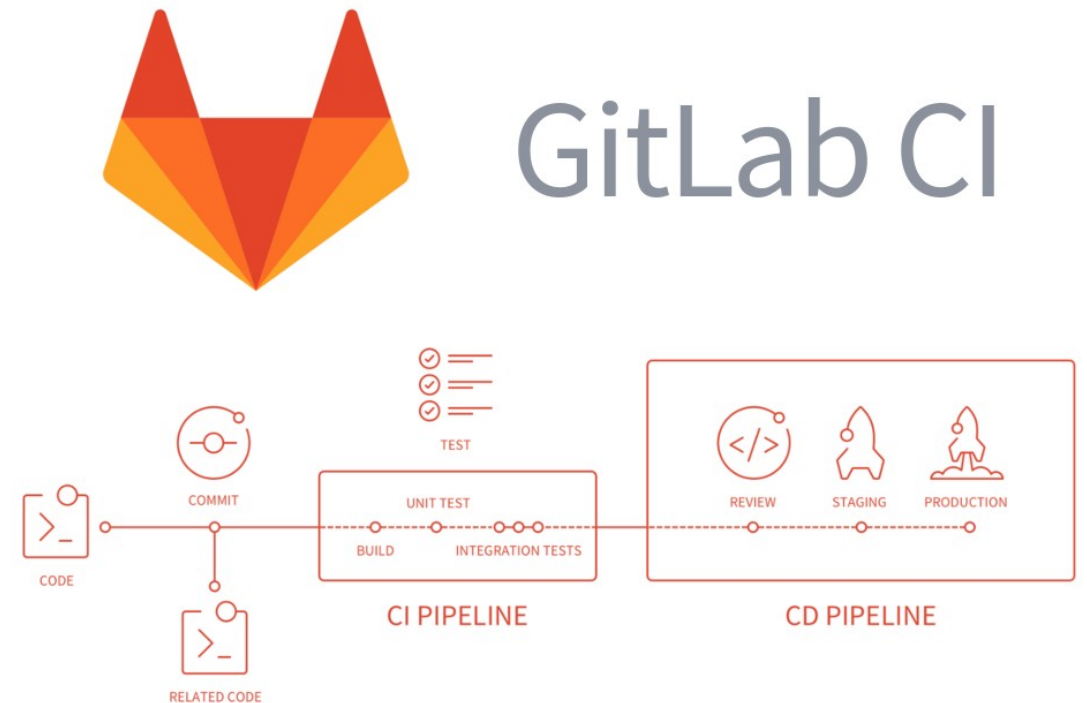
## GitLab CI



# GitLab CI



- Рассмотрим вкратце другой CI сервис — **GitLab CI** и его отличия от **GitHub Actions**
- Сам по себе **GitLab** такое же облачное хранилище репозиториев, как и **GitHub**
- Он менее популярен как платформа, однако намного более популярен внутри компаний, ведь каждый развернуть копию гитлаба у себя во внутренней сети компании **локально**

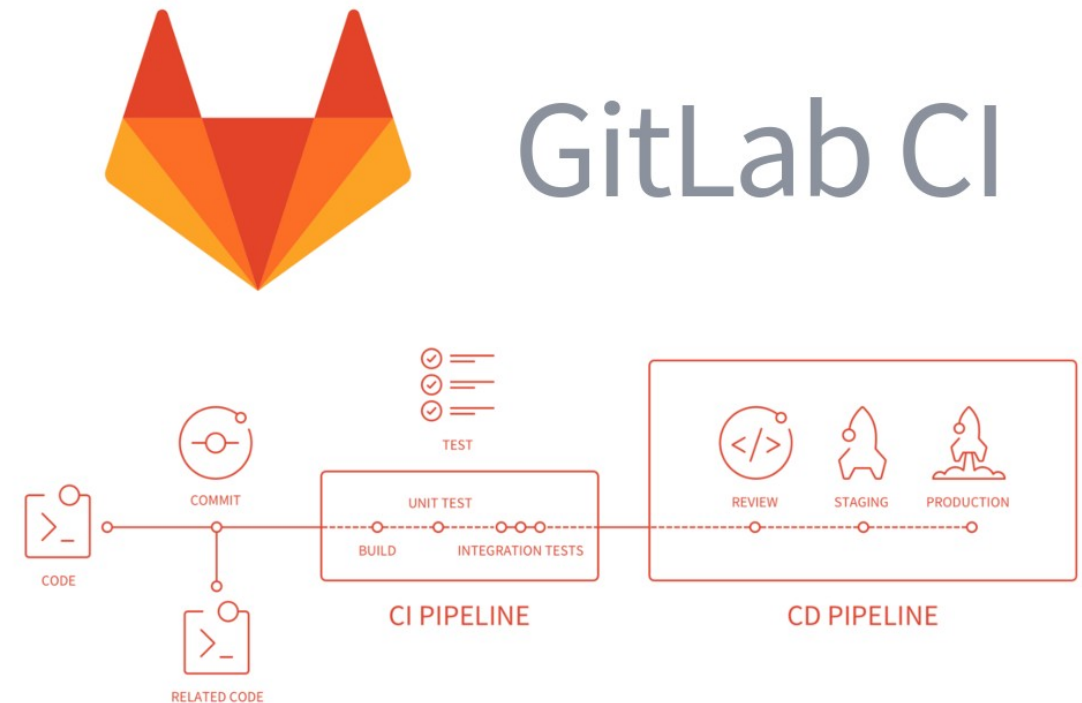




# GitLab CI



- Рассмотрим вкратце другой CI сервис — **GitLab CI** и его отличия от **GitHub Actions**
- Сам по себе **GitLab** такое же облачное хранилище репозиториев, как и **GitHub**
- Он менее популярен как платформа, однако намного более популярен внутри компаний, ведь каждый развернуть копию гитлаба у себя во внутренней сети компании **локальн**
- Это позволяет намного более гибко настраивать свой гитлаб, безопасно вести работу во внутренней сети и не зависеть от внешней сети и интернета



# GitLab CI



- Пример пайплайна с использованием **GitLab CI**

## **workflow:**

### **rules:**

- **if:** \$CI\_COMMIT\_TITLE =~ /-draft\$/  
**when:** never
- **if:** \$CI\_PIPELINE\_SOURCE == "merge\_request\_event"  
- **if:** \$CI\_COMMIT\_BRANCH == \$CI\_DEFAULT\_BRANCH

## **main-job:**

### **tags:**

- saas-linux-small-amd64

### **script:**

- apt update && apt install -y libgtest-dev
- cd examples/arithmetic\_progression
- mkdir build
- cmake -S . -B build
- cmake --build build
- ctest --test-dir build