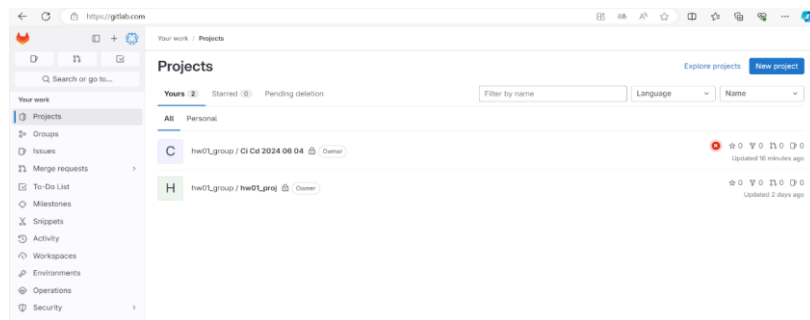


1 Зарегистрироваться на gitlab.com

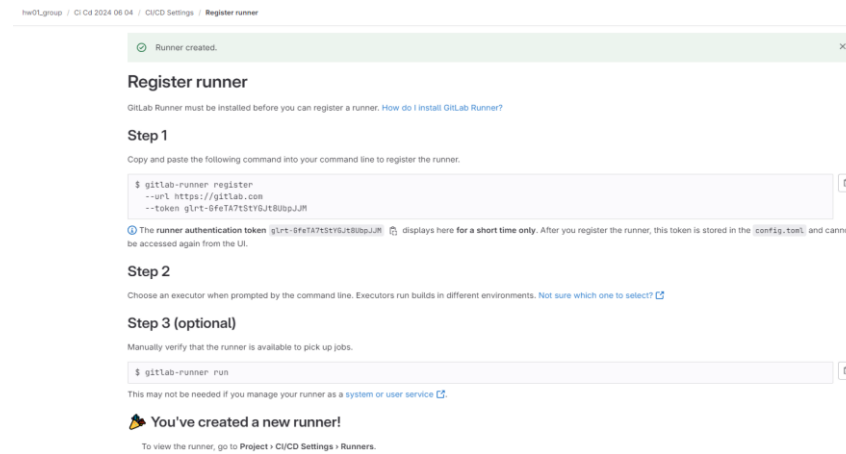


2 Создать pipeline и runner

2.1 создать runner

установил runner на virtualbox

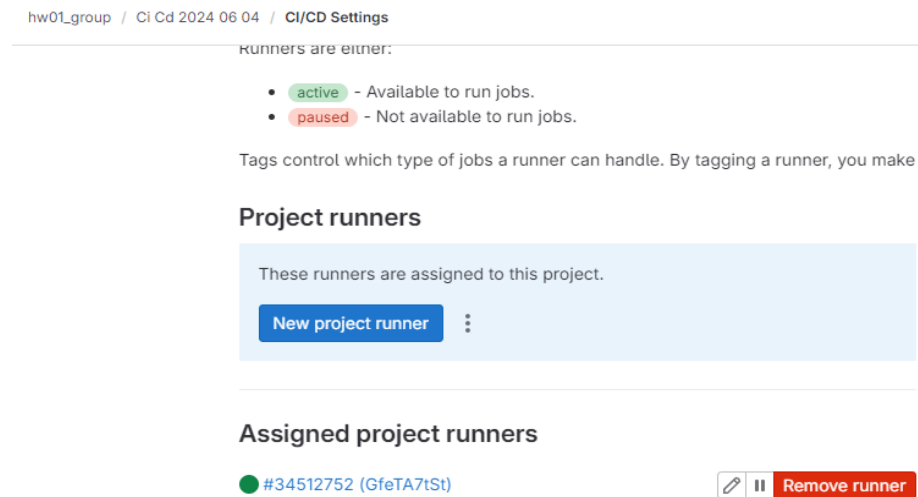
зарегистрировал runner - получил токен на GitLab



настроил runner (указал token)

запустил раннер

проверил, что runner отображается в проект Gitlab



2.2 создать pipeline

https://gitlab.com/hw01_group/ci-cd-2024-06-04/-/pipelines

hw01_group / Ci Cd 2024 06 04 / Pipelines

All 5 Finished Branches Tags

Clear runner caches CI lint Run pipeline

Filter pipelines

Show Pipeline ID

Status	Pipeline	Created by	Stages
Failed 00:00:21 3 minutes ago	Update .gitlab-ci.yml file #1243230906 main b3c01cbc latest		✓ ✗ » ↺ ↓
Failed 00:00:18 8 minutes ago	Update .gitlab-ci.yml file #1243228779 main 3dd13b57		✓ ✗ » ↺ ↓
Failed 00:00:19 10 minutes ago	Update .gitlab-ci.yml file #1243227713 main 9600f81d		✓ ✗ » ↺ ↓
Passed 00:00:28 17 minutes ago	Update .gitlab-ci.yml file #1243226597 main 5026c1b0		✓ ✓ ✓ ↓
Passed 00:00:30 14 hours ago	Update .gitlab-ci.yml file #1242769889 main 25327c57		✓ ✓ ✓ ↓

3 Сохранить артефакт одной из стадий + исключить из папки с артефактами любой файл

3.1 Сохранил артефакт

image: **busybox:latest**

stages:

- **build**
- **test**
- **deploy**

build1:

stage: **build**

script:

- **echo "Do your build here"**
- **echo one >> file1.txt**

artifacts:

paths:

- **file1.txt**

expire_in: **30 days**

test1:

stage: **test**

script:

- **echo "Do a test here"**
- **echo "For example run a test suite"**
- **grep one file1.txt**

test2:

stage: **test**

script:

- **echo "Do another parallel test here"**
- **echo "For example run a lint test"**
- **grep two file1.txt**

deploy1:

stage: **deploy**

script:

- **echo "Do your deploy here"**

environment: **production**

```
5 Preparing the "docker+machine" executor 00:05
6 Using Docker executor with image busybox:latest ...
7 Pulling docker image busybox:latest ...
8 Using docker image sha256:ba5dc23f65d4cc4a4535bce55cf9e63b068eb02946e3422d3587e8ce803b6aab for busybox:la
test with digest busybox@sha256:c3839dd80b9eb7603340509769c43e146a74c63dca3045a8e7dc8ee07e53966 ...
9 Preparing environment 00:01
10 Running on runner-j1aldqxs-project-56580813-concurrent-0 via runner-j1aldqxs-s-l-s-amd64-1712478677-69633
5f8...
11 Getting source from Git repository 00:01
18 Downloading artifacts 00:01
19 Downloading artifacts for build1 (6566034020)...
20 Downloading artifacts from coordinator... ok host=storage.googleapis.com id=6566034020 responseSta
tus=200 OK token=glcvt-65
21 Executing "step_script" stage of the job script 00:00
22 Using docker image sha256:ba5dc23f65d4cc4a4535bce55cf9e63b068eb02946e3422d3587e8ce803b6aab for busybox:la
test with digest busybox@sha256:c3839dd80b9eb7603340509769c43e146a74c63dca3045a8e7dc8ee07e53966 ...
23 $ echo "Do a test here"
24 Do a test here
25 $ echo "For example run a test suite"
26 For example run a test suite
27 $ grep one file1.txt
28 one
29 Cleaning up project directory and file based variables 00:00
30 Job succeeded
```

3.2 Исключил артефакт

```
test1:
  stage: test
  script:
    - echo "Do a test here"
    - echo "For example run a test suite"
    - grep one file1.txt
  artifacts:
    paths:
      - ./
    exclude:
      - file1.txt

test2:
  stage: test
  script:
    - echo "Do another parallel test here"
    - echo "For example run a lint test"
    - grep two file2.txt
  artifacts:
    paths:
      - ./
    exclude:
      - file2.txt
```

The screenshot displays the GitLab CI/CD interface for a job named `test1`. The left sidebar shows the project structure with `Jobs` selected. The main panel shows the job log, which includes the following steps:

- 1. Downloading artifacts (00:00)
- 2. Downloading artifacts from repository (00:00)
- 3. Downloading artifacts from repository (00:00)
- 4. Downloading artifacts from repository (00:00)
- 5. Downloading artifacts from repository (00:00)
- 6. Downloading artifacts from repository (00:00)
- 7. Downloading artifacts from repository (00:00)
- 8. Downloading artifacts from repository (00:00)
- 9. Downloading artifacts from repository (00:00)
- 10. Downloading artifacts from repository (00:00)
- 11. Downloading artifacts from repository (00:00)
- 12. Downloading artifacts from repository (00:00)
- 13. Downloading artifacts from repository (00:00)
- 14. Downloading artifacts from repository (00:00)
- 15. Downloading artifacts from repository (00:00)
- 16. Downloading artifacts from repository (00:00)
- 17. Downloading artifacts from repository (00:00)
- 18. Downloading artifacts from repository (00:00)
- 19. Downloading artifacts from repository (00:00)
- 20. Downloading artifacts from repository (00:00)
- 21. Downloading artifacts from repository (00:00)
- 22. Downloading artifacts from repository (00:00)
- 23. Downloading artifacts from repository (00:00)
- 24. Downloading artifacts from repository (00:00)
- 25. Downloading artifacts from repository (00:00)
- 26. Downloading artifacts from repository (00:00)
- 27. Downloading artifacts from repository (00:00)
- 28. Downloading artifacts from repository (00:00)
- 29. Downloading artifacts from repository (00:00)
- 30. Downloading artifacts from repository (00:00)
- 31. Downloading artifacts from repository (00:00)
- 32. Downloading artifacts from repository (00:00)
- 33. Downloading artifacts from repository (00:00)
- 34. Downloading artifacts from repository (00:00)
- 35. Downloading artifacts from repository (00:00)
- 36. Downloading artifacts from repository (00:00)
- 37. Downloading artifacts from repository (00:00)
- 38. Downloading artifacts from repository (00:00)
- 39. Downloading artifacts from repository (00:00)
- 40. Downloading artifacts from repository (00:00)
- 41. Downloading artifacts from repository (00:00)
- 42. Downloading artifacts from repository (00:00)
- 43. Downloading artifacts from repository (00:00)
- 44. Downloading artifacts from repository (00:00)
- 45. Downloading artifacts from repository (00:00)
- 46. Downloading artifacts from repository (00:00)
- 47. Downloading artifacts from repository (00:00)
- 48. Downloading artifacts from repository (00:00)
- 49. Downloading artifacts from repository (00:00)
- 50. Downloading artifacts from repository (00:00)
- 51. Downloading artifacts from repository (00:00)
- 52. Downloading artifacts from repository (00:00)
- 53. Downloading artifacts from repository (00:00)
- 54. Downloading artifacts from repository (00:00)
- 55. Downloading artifacts from repository (00:00)
- 56. Downloading artifacts from repository (00:00)
- 57. Downloading artifacts from repository (00:00)
- 58. Downloading artifacts from repository (00:00)
- 59. Downloading artifacts from repository (00:00)
- 60. Downloading artifacts from repository (00:00)
- 61. Downloading artifacts from repository (00:00)
- 62. Downloading artifacts from repository (00:00)
- 63. Downloading artifacts from repository (00:00)
- 64. Downloading artifacts from repository (00:00)
- 65. Downloading artifacts from repository (00:00)
- 66. Downloading artifacts from repository (00:00)
- 67. Downloading artifacts from repository (00:00)
- 68. Downloading artifacts from repository (00:00)
- 69. Downloading artifacts from repository (00:00)
- 70. Downloading artifacts from repository (00:00)
- 71. Downloading artifacts from repository (00:00)
- 72. Downloading artifacts from repository (00:00)
- 73. Downloading artifacts from repository (00:00)
- 74. Downloading artifacts from repository (00:00)
- 75. Downloading artifacts from repository (00:00)
- 76. Downloading artifacts from repository (00:00)
- 77. Downloading artifacts from repository (00:00)
- 78. Downloading artifacts from repository (00:00)
- 79. Downloading artifacts from repository (00:00)
- 80. Downloading artifacts from repository (00:00)
- 81. Downloading artifacts from repository (00:00)
- 82. Downloading artifacts from repository (00:00)
- 83. Downloading artifacts from repository (00:00)
- 84. Downloading artifacts from repository (00:00)
- 85. Downloading artifacts from repository (00:00)
- 86. Downloading artifacts from repository (00:00)
- 87. Downloading artifacts from repository (00:00)
- 88. Downloading artifacts from repository (00:00)
- 89. Downloading artifacts from repository (00:00)
- 90. Downloading artifacts from repository (00:00)
- 91. Downloading artifacts from repository (00:00)
- 92. Downloading artifacts from repository (00:00)
- 93. Downloading artifacts from repository (00:00)
- 94. Downloading artifacts from repository (00:00)
- 95. Downloading artifacts from repository (00:00)
- 96. Downloading artifacts from repository (00:00)
- 97. Downloading artifacts from repository (00:00)
- 98. Downloading artifacts from repository (00:00)
- 99. Downloading artifacts from repository (00:00)
- 100. Downloading artifacts from repository (00:00)

The right sidebar shows the job details, including the duration (12 seconds), finished status, and a list of artifacts. The artifacts are listed as `file1.txt` and `file2.txt`, both of which are excluded from the job.

4 Сделать gitlab pages

