

Gitlab

Дано:

Код приложения

Три подготовленные VM на yandex cloud

(установлено git, gitlab-runner, docker, docker-compose):

- gitlab-server — на сервере развернут gitlab
- project-main — сервер для ветки main он же production
- project-stage — сервер для ветки stage

Задание:

Настроить авто деплой приложения на сервера согласно веток.




Три VM на yandex cloud

← → ↻ console.cloud.yandex.ru/folders/b1gv4l14uqqeafgtpcpr/compute/instances

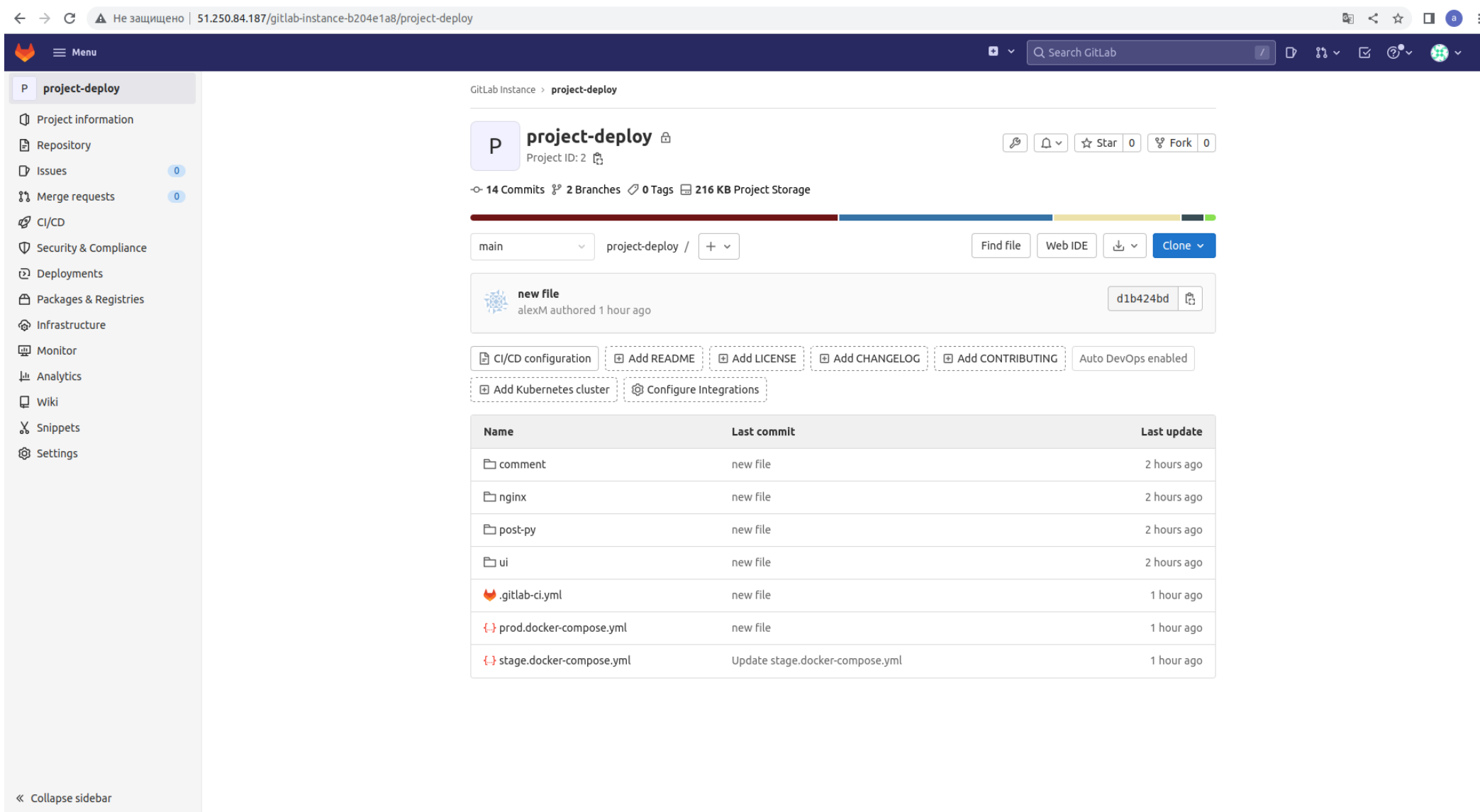
cloud-mailnovs... Infa Compute Cloud / Виртуальные машины [Создать VM](#)

Виртуальные машины

Фильтр по имени Все статусы Все зоны доступности

<input type="checkbox"/>	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания	Идентификатор	⚙
<input type="checkbox"/>	gitlab-server	Running		Intel Ice Lake	2	100 %	4 ГБ	нет	13 ГБ	ru-central1-a	10.128.0.17	51.250.84.187	2022-08-31, в 10:20	fhn951bv9nr52n5s124l	...
<input type="checkbox"/>	project-stage	Running		Intel Ice Lake	2	50 %	2 ГБ	нет	13 ГБ	ru-central1-a	10.128.0.22	84.252.128.173	2022-08-31, в 10:22	fhnfi1laia5ngu991j4c	...
<input type="checkbox"/>	project-main	Running		Intel Ice Lake	2	50 %	2 ГБ	нет	13 ГБ	ru-central1-a	10.128.0.9	84.201.175.49	2022-08-31, в 10:10	fhnppvfrfv0rcbvt8478l	...

Код приложения размещенный на Gitlab



Создано две ветки stage и main

← → ↻ Не защищено | 51.250.84.187/gitlab-instance-b204e1a8/project-deploy/-/branches

Menu

project-deploy

- Project information
- Repository
 - Files
 - Commits
 - Branches**
 - Tags
 - Contributors
 - Graph
 - Compare
- Issues 0
- Merge requests 0
- CI/CD
- Security & Compliance
- Deployments
- Packages & Registries
- Infrastructure
- Monitor
- Analytics
- Wiki
- Snippets
- Settings

GitLab Instance > project-deploy > Repository > Branches

Overview Active Stale All

Filter by branch name

Delete merged branches New branch

Active branches

stage	ff5f198f · Update .gitlab-ci.yml · 8 minutes ago	0 5	Merge request	Compare	Download	Delete
main	d1b424bd · new file · 1 hour ago				Download	Delete

« Collapse sidebar

Зарегистрированы два runners.

Один имеет тег production и выполняет разворачивание приложения на сервере project-main другой имеет тег stage и выполняет разворачивание приложения на сервере project-stage.

The screenshot shows the GitLab web interface for the 'project-deploy' project. The left sidebar contains a menu with options like Project information, Repository, Issues, Merge requests, CI/CD, Security & Compliance, Deployments, Packages & Registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings. The 'CI/CD' option is selected. The main content area is titled 'Runners' and explains that runners are processes that pick up and execute CI/CD jobs. It provides instructions on how to register runners and lists two types: 'Specific runners' and 'Shared runners'. Under 'Specific runners', there are instructions to set up a specific runner for a project, including a URL and a registration token. Under 'Shared runners', there is a toggle to enable shared runners for this project. At the bottom, there is a section for 'Available specific runners' which lists two runners: '#3 (3kJoUVZF)' with tag 'stage' and '#1 (MD_xx_fg)' with tag 'production'. Each runner has a 'Remove runner' button.

Runners are processes that pick up and execute CI/CD jobs for GitLab. [What is GitLab Runner?](#)

Register as many runners as you want. You can register runners as separate users, on separate servers, and on your local machine. Runners are either:

- **active** - Available to run jobs.
- **paused** - Not available to run jobs.

Specific runners

These runners are specific to this project.

Set up a specific runner for a project

1. [Install GitLab Runner and ensure it's running.](#)
2. Register the runner with this URL:
`http://51.250.84.187/`

And this registration token:
`GR1348941xwhppwNL4sE_K81vAAWw`

[Reset registration token](#)

[Show runner installation instructions](#)

Shared runners

These runners are shared across this GitLab instance.

The same shared runner executes code from multiple projects, unless you configure autoscaling with [MaxBuilds](#) set to 1 (which it is on GitLab.com).

Enable shared runners for this project

☒ This GitLab instance does not provide any shared runners yet. Instance administrators can register shared runners in the admin area.

Group runners

These runners are shared across projects in this group.

Group runners can be managed with the [Runner API](#).

[Disable group runners](#) for this project

This group does not have any group runners yet. To register them, go to the [group's Runners page](#).

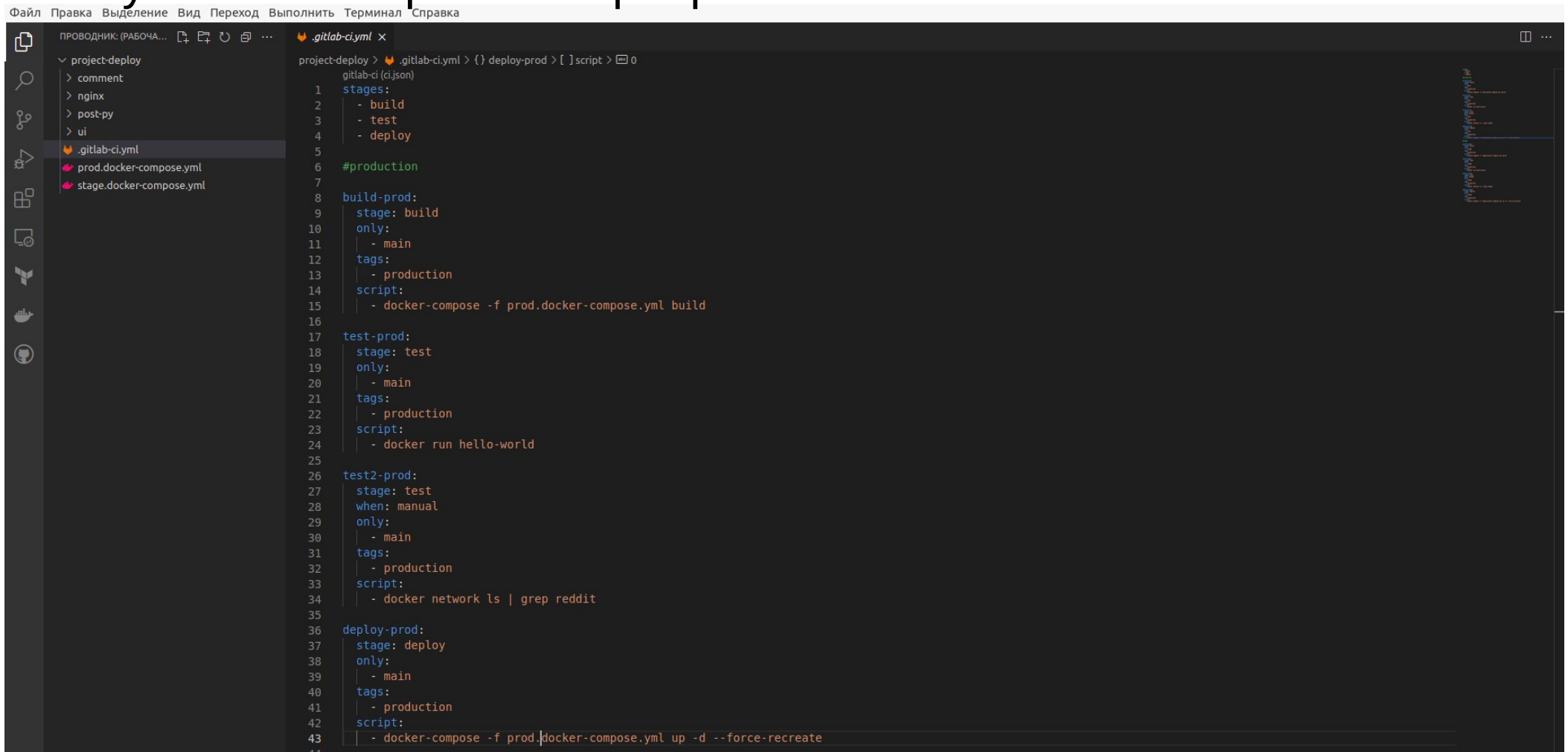
Available specific runners

#3 (3kJoUVZF)	Edit Pause Remove runner
stage	
stage	
#1 (MD_xx_fg)	Edit Pause Remove runner
production	
production	

Файл .gitlab-ci.yml часть про production (ветка main)

В нем описано:

- сборка контейнеров
- тест №1 с авто запуском (для примера)
- тест №2 с ручным запуском (для примера)
- запуск контейнеров на сервере



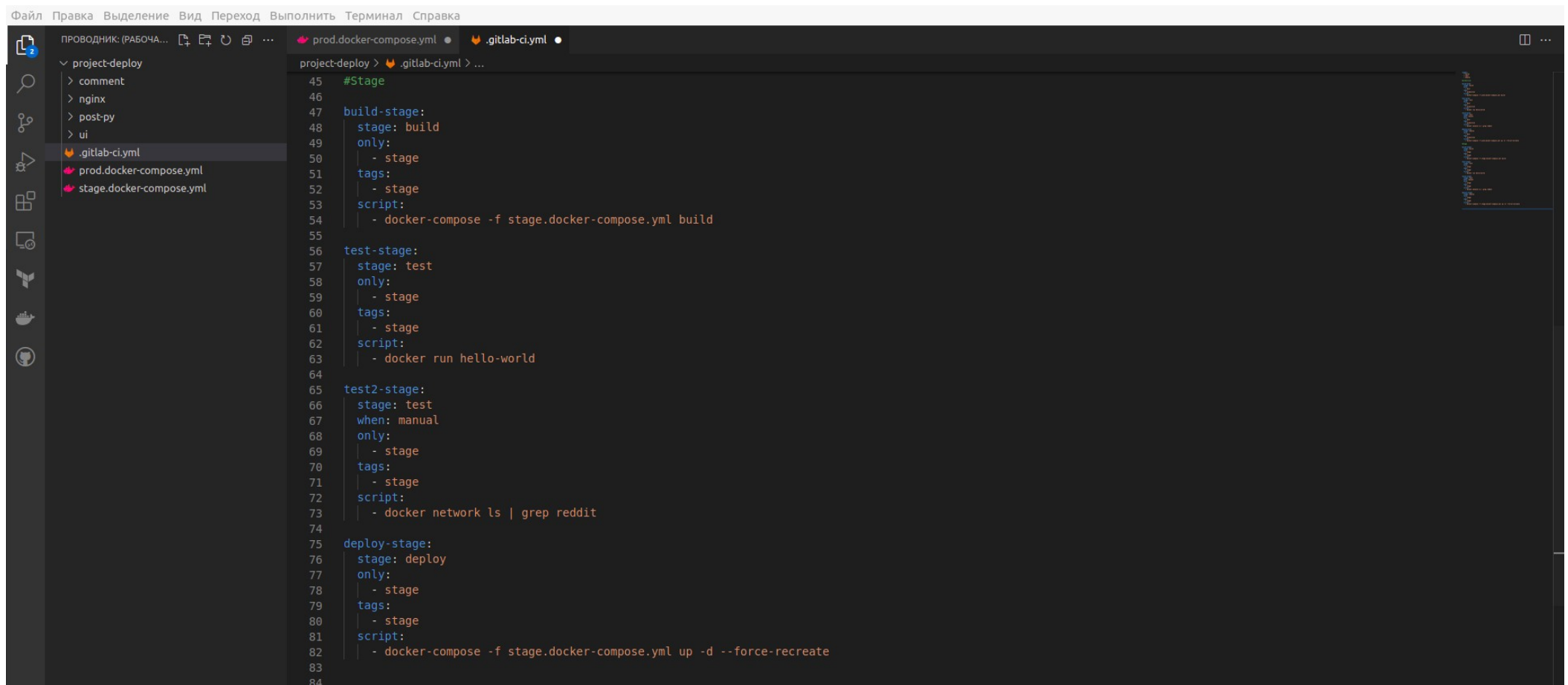
The screenshot shows a code editor with a dark theme. On the left, a file explorer shows a directory structure with files like `comment`, `nginx`, `post.py`, `ui`, `.gitlab-ci.yml`, `prod.docker-compose.yml`, and `stage.docker-compose.yml`. The main editor area displays the content of `.gitlab-ci.yml`. The top of the editor has a menu bar with options like 'Файл', 'Правка', 'Выделение', 'Вид', 'Переход', 'Выполнить', 'Терминал', and 'Справка'. The code in the editor is as follows:

```
project-deploy > .gitlab-ci.yml > {} deploy-prod > [ ] script > 0
gitlab-ci (ci.json)
1  stages:
2    - build
3    - test
4    - deploy
5
6  #production
7
8  build-prod:
9    stage: build
10   only:
11     - main
12   tags:
13     - production
14   script:
15     - docker-compose -f prod.docker-compose.yml build
16
17  test-prod:
18    stage: test
19    only:
20     - main
21   tags:
22     - production
23   script:
24     - docker run hello-world
25
26  test2-prod:
27    stage: test
28    when: manual
29    only:
30     - main
31   tags:
32     - production
33   script:
34     - docker network ls | grep reddit
35
36  deploy-prod:
37    stage: deploy
38    only:
39     - main
40   tags:
41     - production
42   script:
43     - docker-compose -f prod.docker-compose.yml up -d --force-recreate
44
```

Файл .gitlab-ci.yml часть про stage (ветка stage)

В нем описано:

- сборка контейнеров
- тест №1 с авто запуском (для примера)
- тест №2 с ручным запуском (для примера)
- запуск контейнеров на сервере



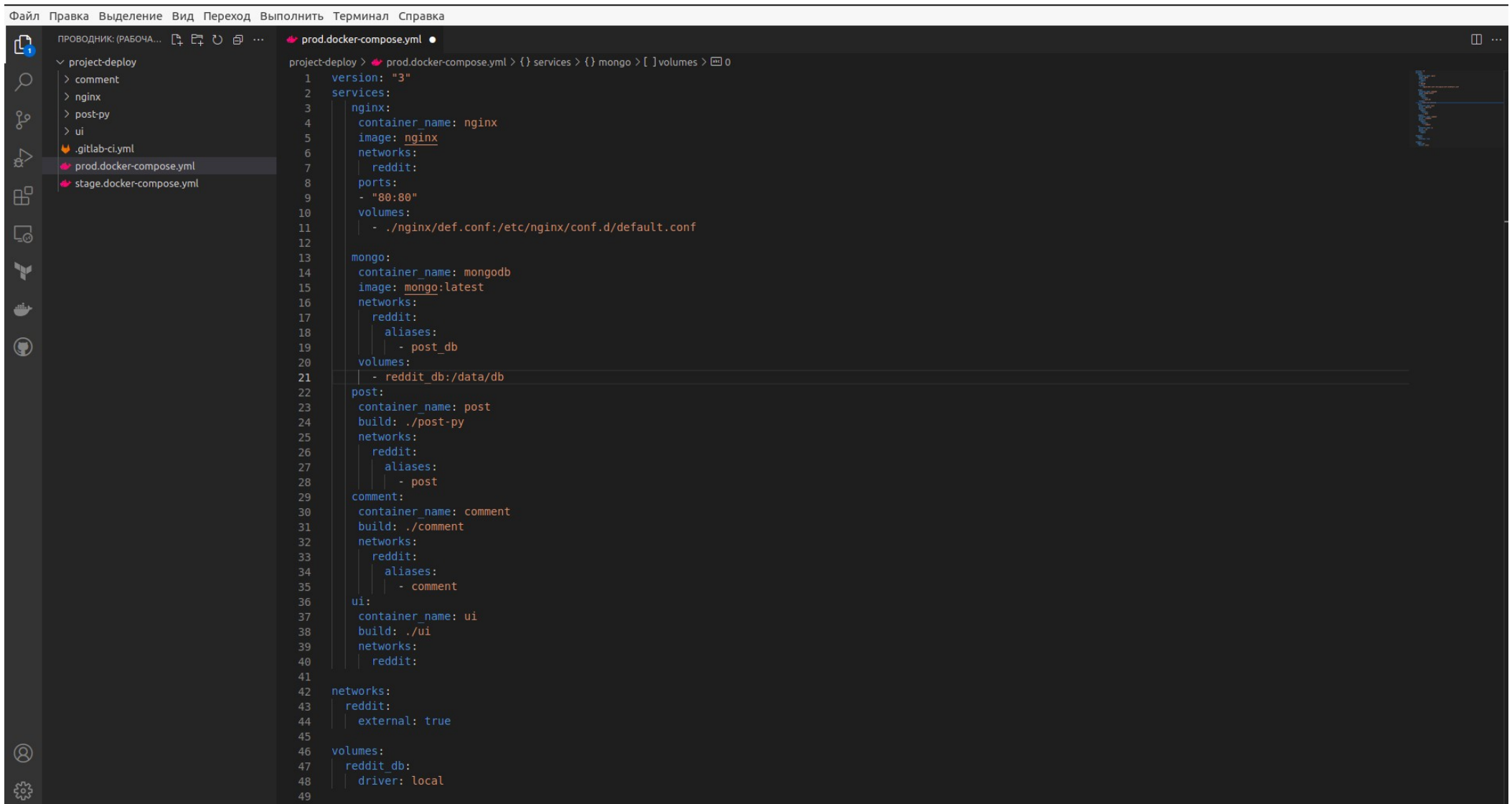
```
45 #Stage
46
47 build-stage:
48   stage: build
49   only:
50     - stage
51   tags:
52     - stage
53   script:
54     - docker-compose -f stage.docker-compose.yml build
55
56 test-stage:
57   stage: test
58   only:
59     - stage
60   tags:
61     - stage
62   script:
63     - docker run hello-world
64
65 test2-stage:
66   stage: test
67   when: manual
68   only:
69     - stage
70   tags:
71     - stage
72   script:
73     - docker network ls | grep reddit
74
75 deploy-stage:
76   stage: deploy
77   only:
78     - stage
79   tags:
80     - stage
81   script:
82     - docker-compose -f stage.docker-compose.yml up -d --force-recreate
83
84
```

Содержимое docker-compose файла.

Для production он называется prod.docker-compose.yml.

Для stage он называется stage.docker-compose.yml.

Содержимое файла идентично.



```
Файл  Правка  Выделение  Вид  Переход  Выполнить  Терминал  Справка
project-deploy > prod.docker-compose.yml > {} services > {} mongo > [ ] volumes > 0

1  version: "3"
2  services:
3    nginx:
4      container_name: nginx
5      image: nginx
6      networks:
7        reddit:
8      ports:
9        - "80:80"
10     volumes:
11       - ./nginx/def.conf:/etc/nginx/conf.d/default.conf
12
13    mongo:
14      container_name: mongodb
15      image: mongo:latest
16      networks:
17        reddit:
18          aliases:
19            - post_db
20      volumes:
21        - reddit_db:/data/db
22
23    post:
24      container_name: post
25      build: ./post-py
26      networks:
27        reddit:
28          aliases:
29            - post
30
31    comment:
32      container_name: comment
33      build: ./comment
34      networks:
35        reddit:
36          aliases:
37            - comment
38
39    ui:
40      container_name: ui
41      build: ./ui
42      networks:
43        reddit:
44
45  networks:
46    reddit:
47      external: true
48
49  volumes:
50    reddit_db:
51      driver: local
```


Производим комит в ветку main и переходим на Gitlab в раздел CI/CD pipelines.
Видим что все stages прошли успешно и что test2-prod ждет ручного запуска.

The screenshot displays the GitLab web interface for a project named 'project-deploy'. The left sidebar contains navigation links: Project information, Repository, Issues (0), Merge requests (0), CI/CD, Pipelines (selected), Editor, Jobs, Schedules, Security & Compliance, Deployments, Packages & Registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings.

The main content area shows the details of Pipeline #15, which was triggered 1 minute ago by the Administrator. The pipeline status is 'passed'. Below this, it indicates the pipeline is for branch 'main' and shows a commit hash 'bfb069e4'. A message states 'No related merge requests found.'

The pipeline progress bar shows three stages: Build, Test, and Deploy. The Build stage has one job, 'build-prod', which is completed. The Test stage has two jobs, 'test2-prod' and 'test-prod'; 'test-prod' is completed, while 'test2-prod' is waiting for manual intervention, indicated by a gear icon. The Deploy stage has one job, 'deploy-prod', which is completed.

Производим комит в ветку stage и переходим на Gitlab в раздел CI/CD pipelines.
Видим что все stages прошли успешно и что test2-prod ждет ручного запуска.

The screenshot shows the GitLab web interface for a project named 'project-deploy'. The left sidebar contains navigation links: Project information, Repository, Issues (0), Merge requests (0), CI/CD, Pipelines (selected), Editor, Jobs, Schedules, Security & Compliance, Deployments, Packages & Registries, Infrastructure, Monitor, Analytics, Wiki, Snippets, and Settings.

The main content area displays the details for Pipeline #16, which was triggered 4 minutes ago by the Administrator. The pipeline status is 'passed'. Below this, the 'pipeline branch stage' section shows that 4 jobs for the 'stage' branch completed in 16 seconds. A 'latest' tag is visible, and a commit hash 'ed278d4e' is shown. A message states 'No related merge requests found.'

The bottom section shows the pipeline stages: Build, Test, and Deploy. The Build stage has a job 'build-stage' that is completed. The Test stage has two jobs: 'test2-stage' (which is in progress, indicated by a play button icon) and 'test-stage' (which is completed). The Deploy stage has a job 'deploy-stage' that is completed.

В итоге:

Приложение развернуто на двух серверах.

При пуше кода в ветку main будет развернут новый код на сервере project-main.

При пуше кода в ветку stage будет развернут новый код на сервере project-stage.

Сервер project-main



Сервер project-stage

