

Docker/Docker-compose/ Docker-machine

Дано:

Проект тестового приложения с кодом.

Задание:

- Написать докер файлы для билдинга каждой части контейнера (comment, post, ui).
- Запустить на yandex cloud VM
- Подключить docker-machine к VM на yandex cloud
- Собрать контейнер с nginx. Написать конфиг для проксирования запросов на порт 9292.
- Описать и запустить через docker-compose.

Проект тестового приложения с кодом

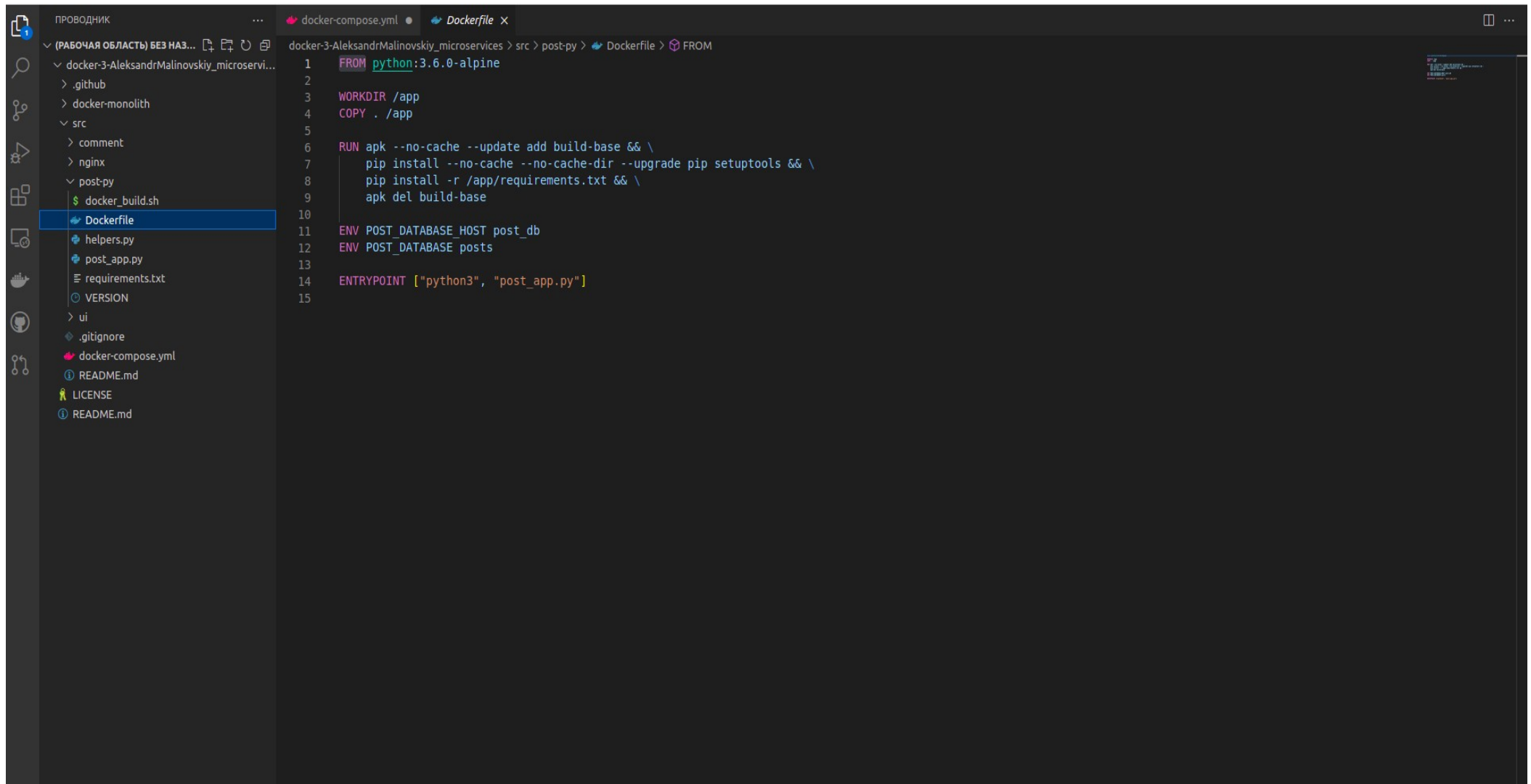
```
root@alex:/home/alex# tree reddit-microservices/
```

```
reddit-microservices/
```

```
├── comment
│   ├── comment_app.rb
│   ├── config.ru
│   ├── docker_build.sh
│   ├── Gemfile
│   ├── Gemfile.lock
│   ├── helpers.rb
│   └── VERSION
├── post-py
│   ├── docker_build.sh
│   ├── helpers.py
│   ├── post_app.py
│   ├── requirements.txt
│   └── VERSION
├── README.md
├── ui
│   ├── config.ru
│   ├── docker_build.sh
│   ├── Gemfile
│   ├── Gemfile.lock
│   ├── helpers.rb
│   ├── middleware.rb
│   ├── ui_app.rb
│   ├── VERSION
│   └── views
│       ├── create.haml
│       ├── index.haml
│       ├── layout.haml
│       └── show.haml
```

```
4 directories, 25 files
```

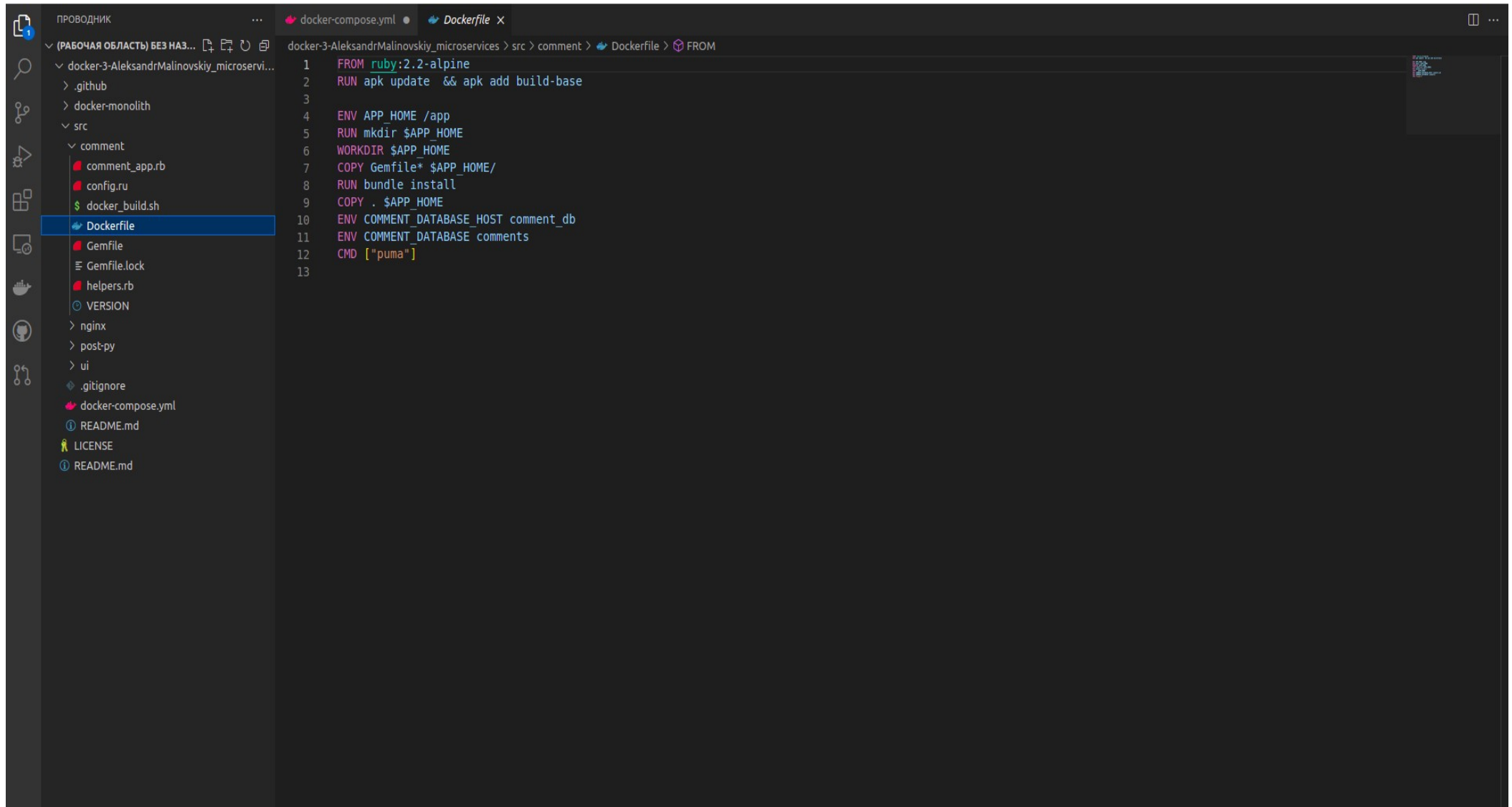
Dockerfile post-py



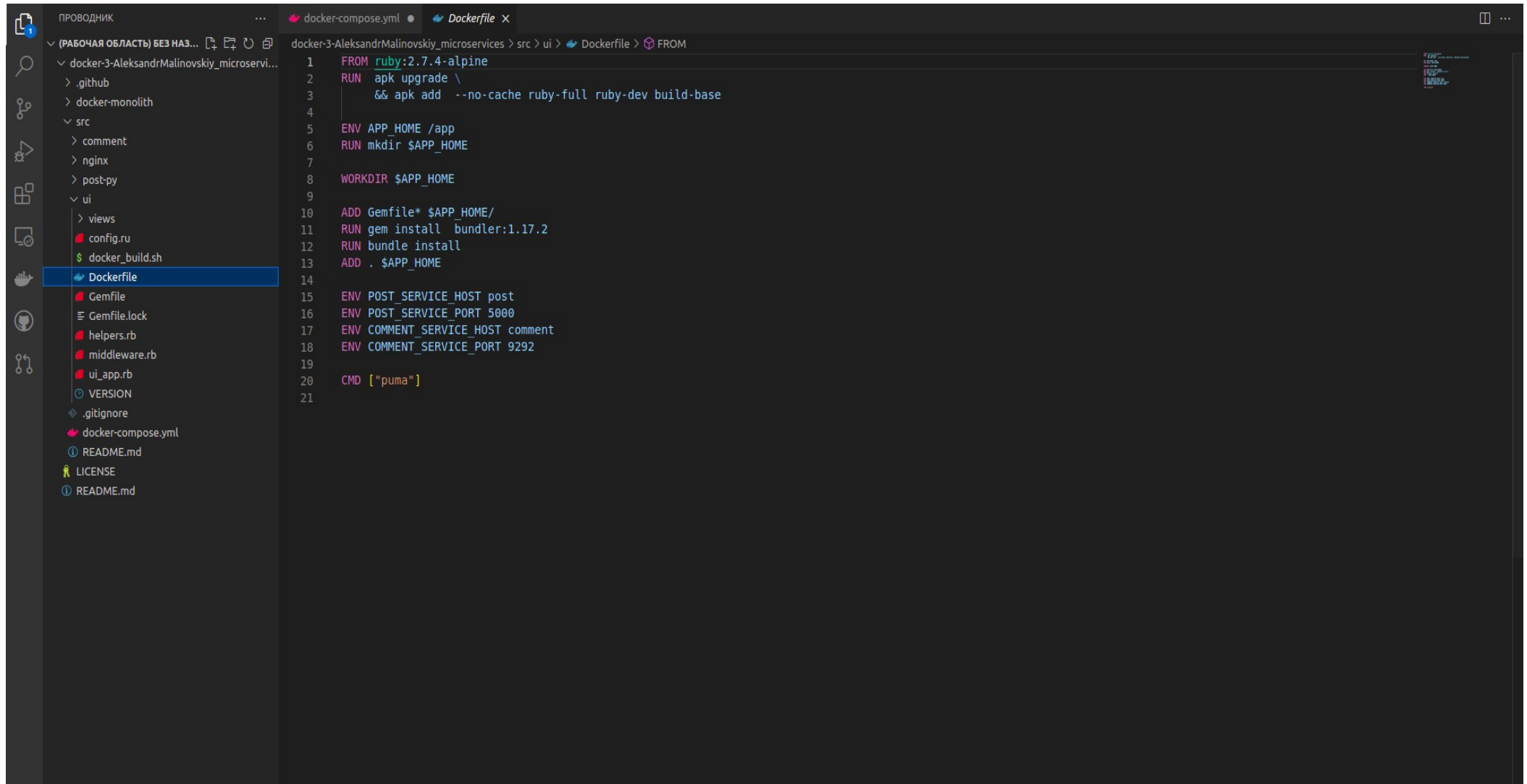
The screenshot shows a code editor with a dark theme. On the left is a file explorer pane titled 'ПРОВОДНИК' (File Explorer). It shows a directory structure for a project named 'docker-3-AleksandrMalinovskiy_microservi...'. The 'post-py' directory is expanded, showing files like 'docker_build.sh', 'Dockerfile' (which is selected), 'helpers.py', 'post_app.py', 'requirements.txt', 'VERSION', 'ui', '.gitignore', 'docker-compose.yml', 'README.md', and 'LICENSE'. The main editor area shows the content of the 'Dockerfile' with line numbers 1 through 15. The Dockerfile instructions are as follows:

```
1 FROM python:3.6.0-alpine
2
3 WORKDIR /app
4 COPY . /app
5
6 RUN apk --no-cache --update add build-base && \
7     pip install --no-cache --no-cache-dir --upgrade pip setuptools && \
8     pip install -r /app/requirements.txt && \
9     apk del build-base
10
11 ENV POST_DATABASE_HOST post_db
12 ENV POST_DATABASE posts
13
14 ENTRYPOINT ["python3", "post_app.py"]
15
```

Dockerfile comment



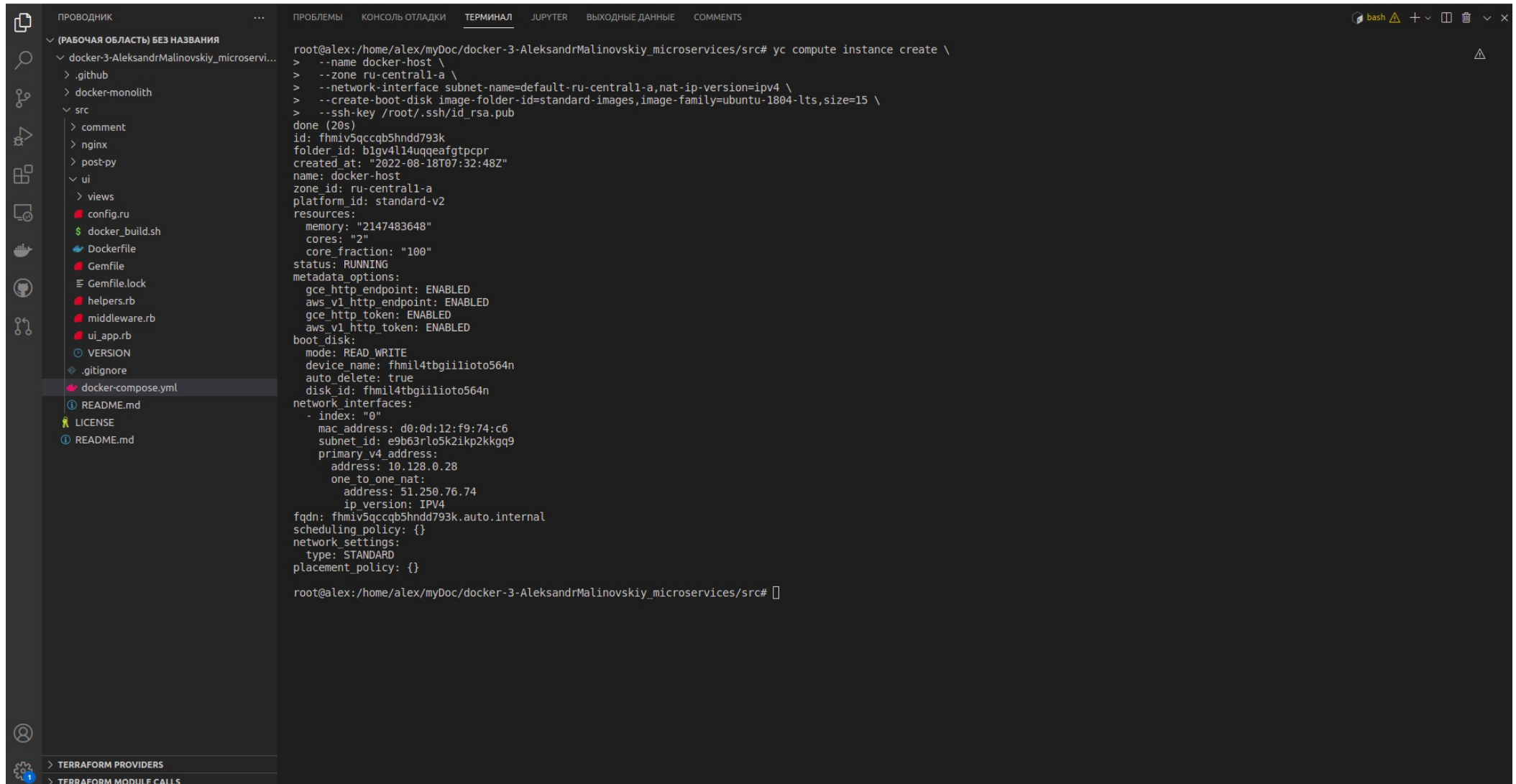
Dockerfile ui



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer displays a project structure for 'docker-3-AleksandrMalinovskiy_microservi...'. The 'ui' directory is expanded, showing files like 'views', 'config.ru', 'docker_build.sh', 'Dockerfile', 'Gemfile', 'Gemfile.lock', 'helpers.rb', 'middleware.rb', 'ui_app.rb', 'VERSION', '.gitignore', 'docker-compose.yml', 'README.md', and 'LICENSE'. The 'Dockerfile' file is selected and highlighted in blue. The code editor displays the content of the 'Dockerfile' file, which is a multi-line script for building a Docker image. The script starts with 'FROM ruby:2.7.4-alpine' and ends with 'CMD ["puma"]'. The script includes instructions for upgrading 'apk', adding packages, setting environment variables, creating directories, installing gems, and running the application.

```
1 FROM ruby:2.7.4-alpine
2 RUN apk upgrade \
3     && apk add --no-cache ruby-full ruby-dev build-base
4
5 ENV APP_HOME /app
6 RUN mkdir $APP_HOME
7
8 WORKDIR $APP_HOME
9
10 ADD Gemfile* $APP_HOME/
11 RUN gem install bundler:1.17.2
12 RUN bundle install
13 ADD . $APP_HOME
14
15 ENV POST_SERVICE_HOST post
16 ENV POST_SERVICE_PORT 5000
17 ENV COMMENT_SERVICE_HOST comment
18 ENV COMMENT_SERVICE_PORT 9292
19
20 CMD ["puma"]
21
```

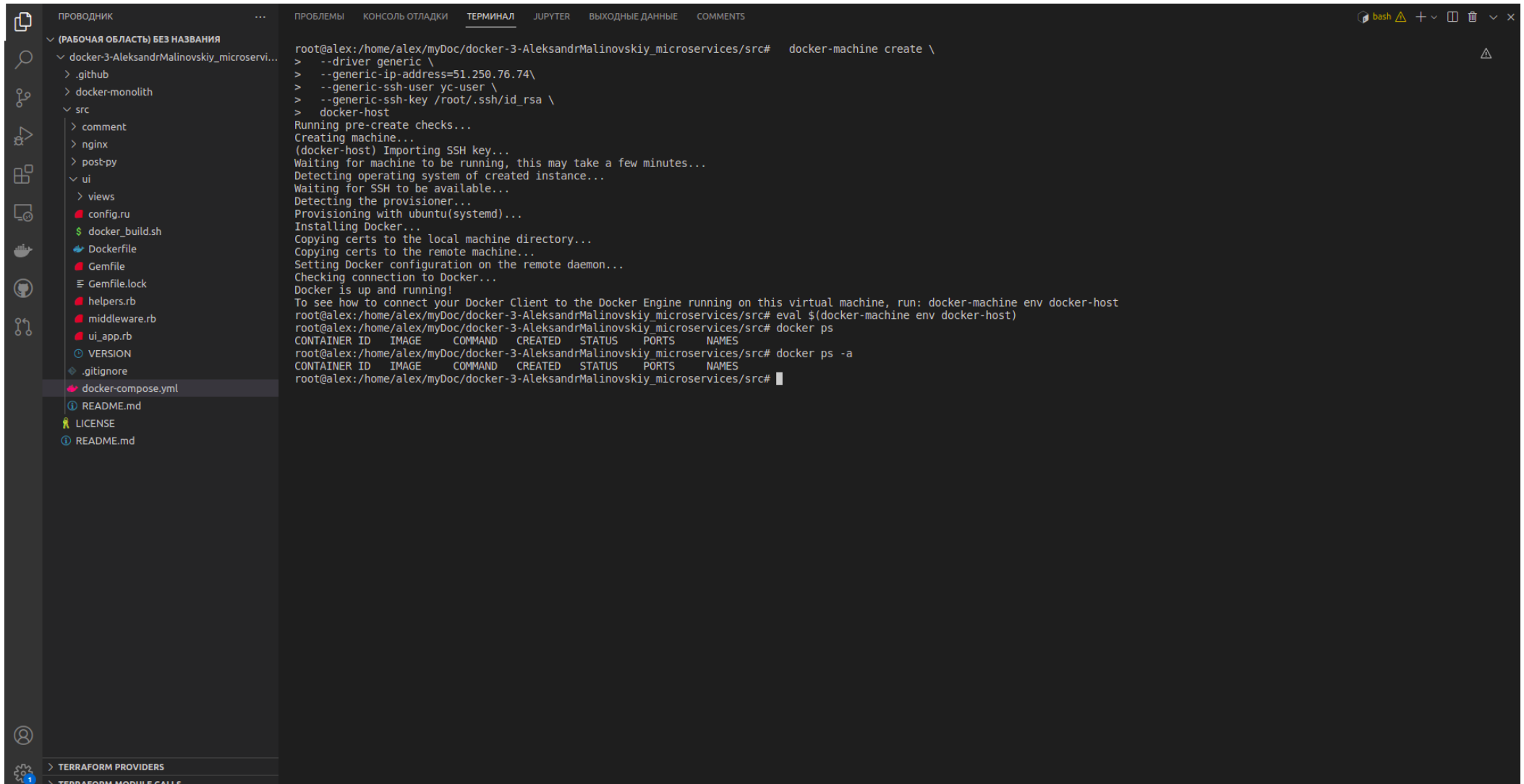
Создание ВМ на yandexcloud



```
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src# yc compute instance create \
> --name docker-host \
> --zone ru-central1-a \
> --network-interface subnet-name=default-ru-central1-a,nat-ip-version=ipv4 \
> --create-boot-disk image-folder-id=standard-images,image-family=ubuntu-1804-lts,size=15 \
> --ssh-key /root/.ssh/id_rsa.pub
done (20s)
id: fhmiV5qccqb5hndd793k
folder_id: blgv4l14uqgeafgtcpr
created_at: "2022-08-18T07:32:48Z"
name: docker-host
zone_id: ru-central1-a
platform_id: standard-v2
resources:
  memory: "2147483648"
  cores: "2"
  core_fraction: "100"
status: RUNNING
metadata_options:
  gce_http_endpoint: ENABLED
  aws_v1_http_endpoint: ENABLED
  gce_http_token: ENABLED
  aws_v1_http_token: ENABLED
boot_disk:
  mode: READ_WRITE
  device_name: fhmiL4tbgiilioto564n
  auto_delete: true
  disk_id: fhmiL4tbgiilioto564n
network_interfaces:
  - index: "0"
    mac_address: d0:0d:12:f9:74:c6
    subnet_id: e9b63rlo5k2ikp2kkgq9
    primary_v4_address:
      address: 10.128.0.28
    one_to_one_nat:
      address: 51.250.76.74
    ip_version: IPV4
fqdn: fhmiV5qccqb5hndd793k.auto.internal
scheduling_policy: {}
network_settings:
  type: STANDARD
placement_policy: {}

root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src#
```

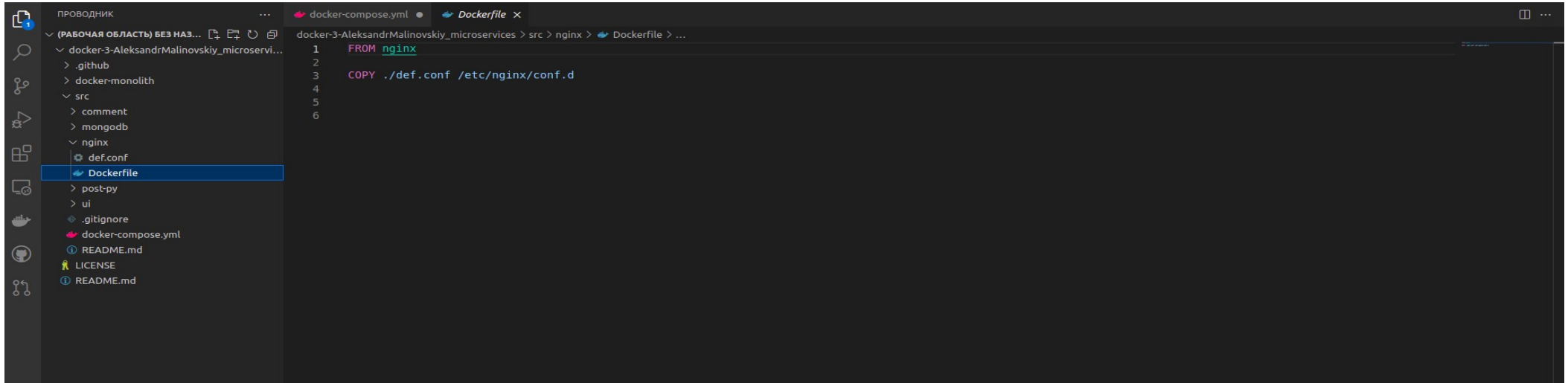
Создание docker-machine и подключение её к ВМ на yandex cloud



```
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src# docker-machine create \
> --driver generic \
> --generic-ip-address=51.250.76.74\
> --generic-ssh-user yc-user \
> --generic-ssh-key /root/.ssh/id_rsa \
> docker-host
Running pre-create checks...
Creating machine...
(docker-host) Importing SSH key...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with ubuntu(systemd)...
Installing Docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docker-machine env docker-host
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src# eval $(docker-machine env docker-host)
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src# docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src#
```

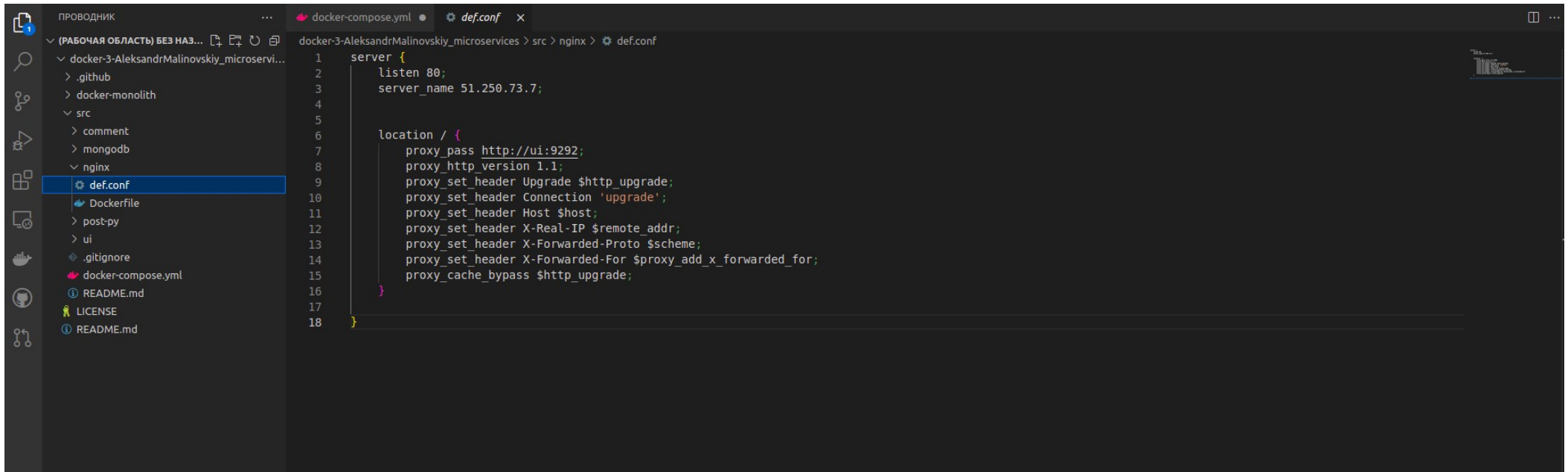
Dockerfile nginx

Конфиг с проксированием на порт 9292



This screenshot shows the Dockerfile for the nginx service in a project. The file is located at `docker-3-AleksandrMalinovskiy_microservices > src > nginx > Dockerfile`. The content of the Dockerfile is as follows:

```
1 FROM nginx
2
3 COPY ./def.conf /etc/nginx/conf.d
4
5
6
```

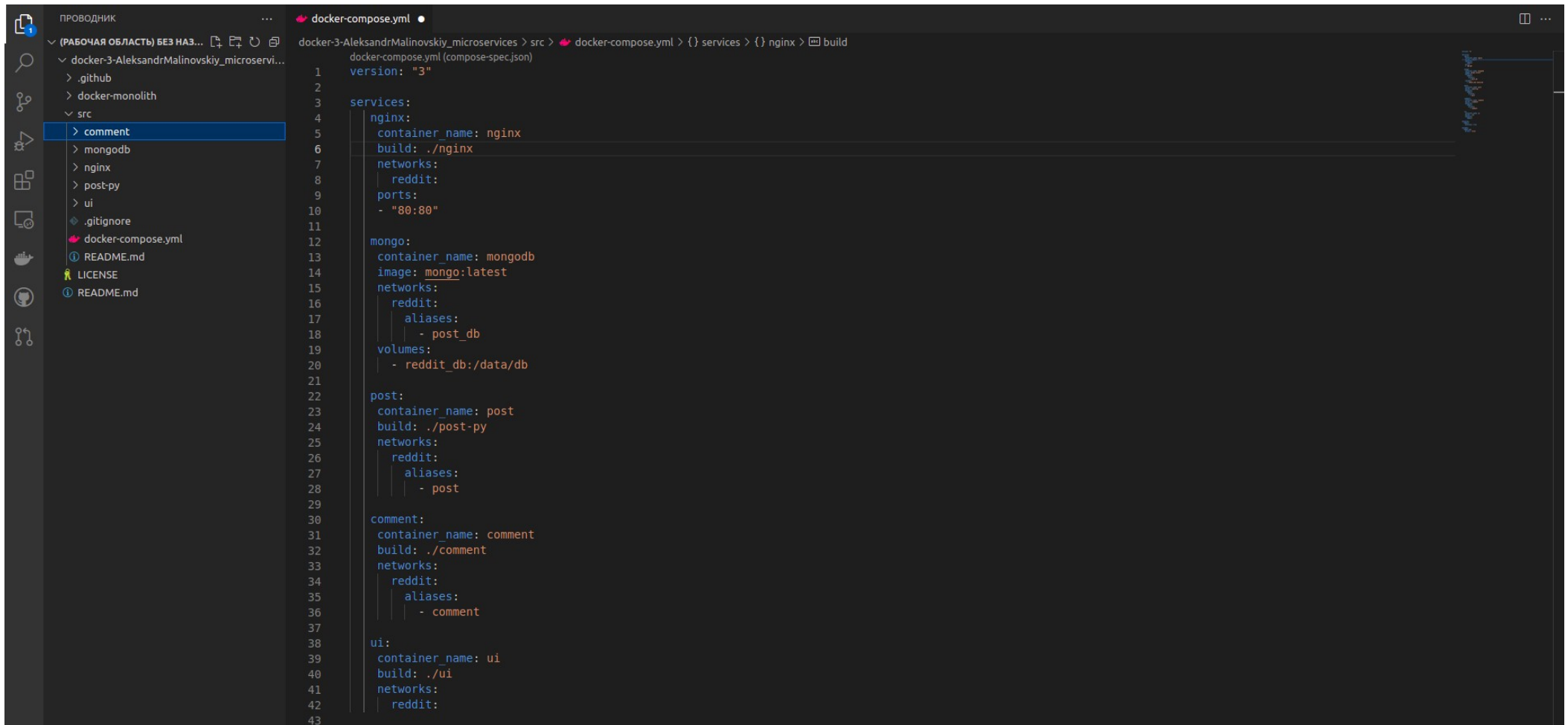


This screenshot shows the configuration file `def.conf` for the nginx service, located at `docker-3-AleksandrMalinovskiy_microservices > src > nginx > def.conf`. The content of the file is as follows:

```
1 server {
2     listen 80;
3     server_name 51.250.73.7;
4
5
6     location / {
7         proxy_pass http://ui:9292;
8         proxy_http_version 1.1;
9         proxy_set_header Upgrade $http_upgrade;
10        proxy_set_header Connection 'upgrade';
11        proxy_set_header Host $host;
12        proxy_set_header X-Real-IP $remote_addr;
13        proxy_set_header X-Forwarded-Proto $scheme;
14        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
15        proxy_cache_bypass $http_upgrade;
16    }
17
18 }
```


Docker-compose.yml

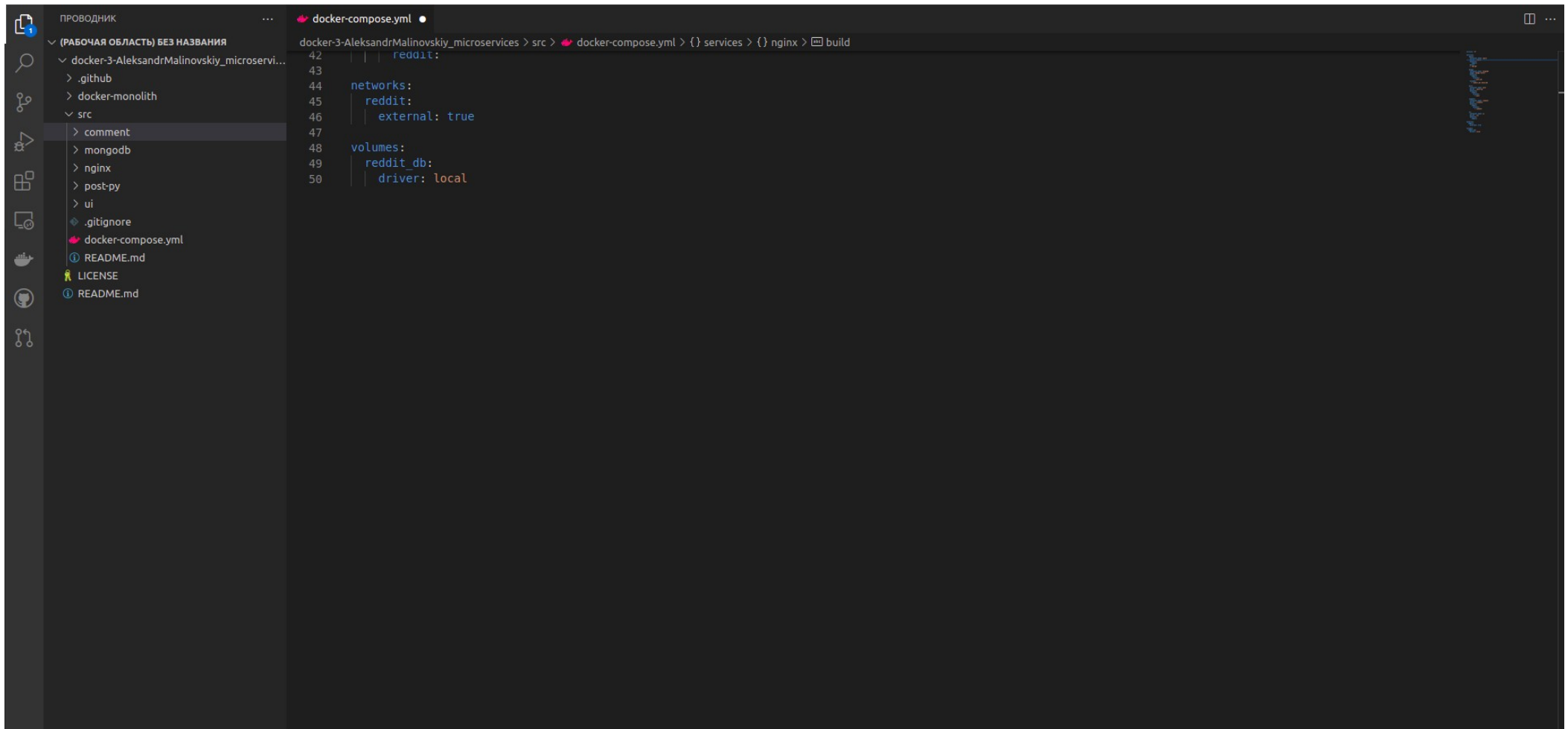
Описание билдинга nginx, comment, post, ui



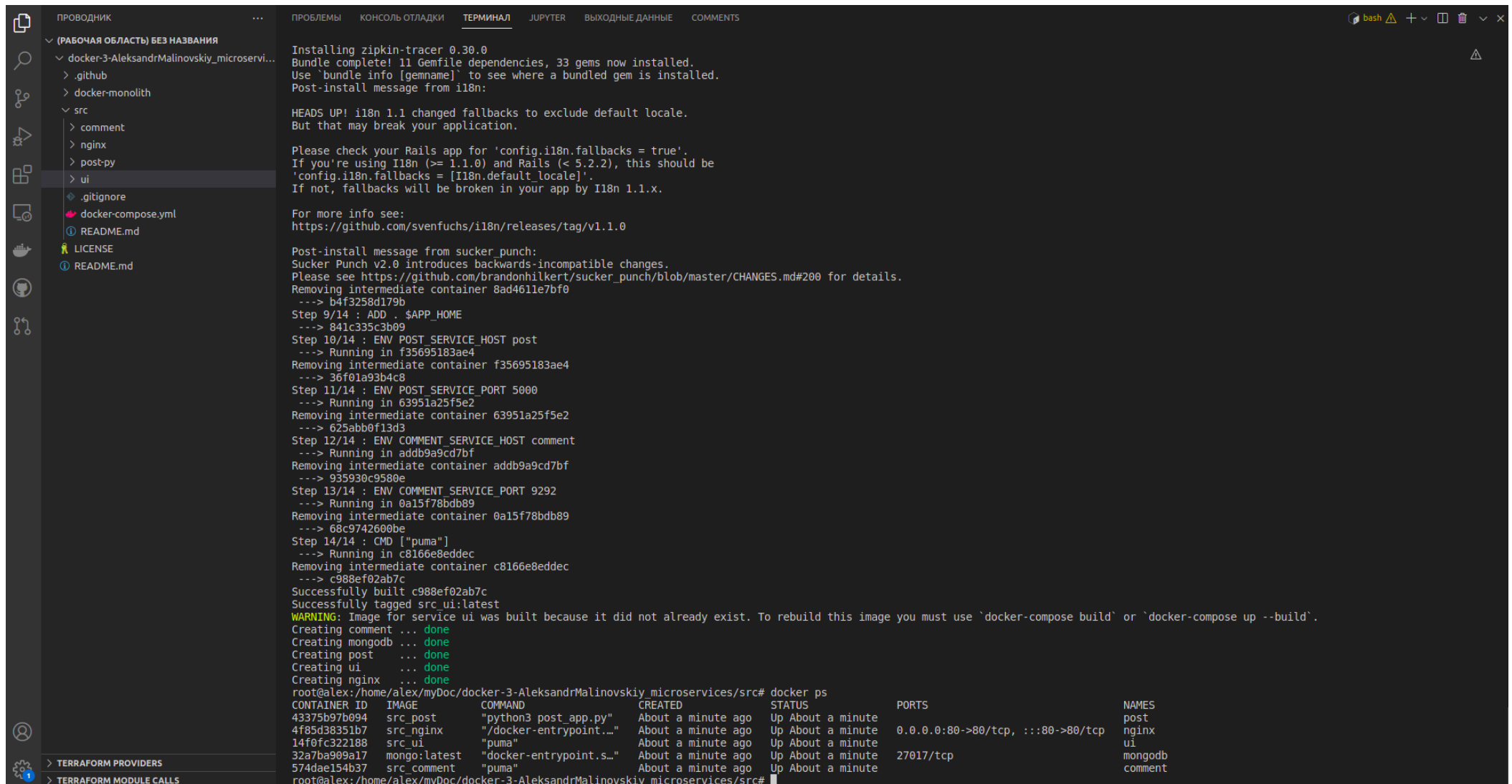
```
1 version: "3"
2
3 services:
4   nginx:
5     container_name: nginx
6     build: ./nginx
7     networks:
8       reddit:
9     ports:
10      - "80:80"
11
12   mongo:
13     container_name: mongodb
14     image: mongo:latest
15     networks:
16       reddit:
17         aliases:
18           - post_db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

Docker-compose.yml

Описание сети и volumes для monoddb



Выполнение команды docker-compose up -d. Произошел билд и запуск все контейнеров описанных в docker-compose.



```
Installing zipkin-tracer 0.30.0
Bundle complete! 11 Gemfile dependencies, 33 gems now installed.
Use 'bundle info [gemname]' to see where a bundled gem is installed.
Post-install message from i18n:

HEADS UP! i18n 1.1 changed fallbacks to exclude default locale.
But that may break your application.

Please check your Rails app for 'config.i18n.fallbacks = true'.
If you're using I18n (>= 1.1.0) and Rails (< 5.2.2), this should be
'config.i18n.fallbacks = [I18n.default_locale]'.
If not, fallbacks will be broken in your app by I18n 1.1.x.

For more info see:
https://github.com/svenfuchs/i18n/releases/tag/v1.1.0

Post-install message from sucker_punch:
Sucker Punch v2.0 introduces backwards-incompatible changes.
Please see https://github.com/brandonhilkert/sucker_punch/blob/master/CHANGES.md#200 for details.
Removing intermediate container 8ad4611e7bf0
--> b4f3258d179b
Step 9/14 : ADD . $APP_HOME
--> 841c335c3b09
Step 10/14 : ENV POST_SERVICE_HOST post
--> Running in f35695183ae4
Removing intermediate container f35695183ae4
--> 36f01a93b4c8
Step 11/14 : ENV POST_SERVICE_PORT 5000
--> Running in 63951a25f5e2
Removing intermediate container 63951a25f5e2
--> 625abb0f13d3
Step 12/14 : ENV COMMENT_SERVICE_HOST comment
--> Running in addb9a9cd7bf
Removing intermediate container addb9a9cd7bf
--> 935930c9580e
Step 13/14 : ENV COMMENT_SERVICE_PORT 9292
--> Running in 0a15f78bdb89
Removing intermediate container 0a15f78bdb89
--> 68c9742600be
Step 14/14 : CMD ["puma"]
--> Running in c8166e8eddec
Removing intermediate container c8166e8eddec
--> c988ef02ab7c
Successfully built c988ef02ab7c
Successfully tagged src_ui:latest
WARNING: Image for service ui was built because it did not already exist. To rebuild this image you must use `docker-compose build` or `docker-compose up --build`.
Creating comment ... done
Creating mongodb ... done
Creating post ... done
Creating ui ... done
Creating nginx ... done
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
43375b97b094   src_post      "python3 post_app.py"    About a minute ago Up About a minute                post
4f85d38351b7   src_nginx     "/docker-entrypoint..." About a minute ago Up About a minute                nginx
14f0fc322188   src_ui        "puma"                   About a minute ago Up About a minute                ui
32a7ba909a17   mongo:latest  "docker-entrypoint.s..." About a minute ago Up About a minute                mongodb
574dae154b37   src_comment   "puma"                   About a minute ago Up About a minute                comment
root@alex:/home/alex/myDoc/docker-3-AleksandrMalinovskiy_microservices/src#
```

Итог:

- На ip VM расположено приложение.
- Все порты кроме порта nginx из сети не доступны.

