

## **КАК ЗАДЕПЛОИТЬ САЙТ VUE НА VK CLOUD**

КЕРБЕР ЕГОР

17 ФЕВРАЛЯ 2024 Г.

# СОДЕРЖАНИЕ

<b>1 Регистрация на vk cloud и добавление правила</b>	<b>3</b>
<b>2 Настройка nginx</b>	<b>7</b>
<b>3 Покупка домена</b>	<b>11</b>
<b>4 Конфигурация vite и docker-compose</b>	<b>14</b>
4.1 Объяснение конфигурации от ChatGPT . . . . .	15

# РЕГИСТРАЦИЯ НА VK CLOUD И ДОБАВЛЕНИЕ ПРАВИЛА

## 1. Регистрируемся на vk cloud

## 2. Создаем инстанс

The screenshot shows the VK Cloud control panel. On the left is a sidebar with various service links: Главная, Облачные вычисления, Виртуальные сети, Data Platform, Мониторинг, CDN, DNS, Объектное хранилище, Контейнеры, Базы данных, Аналитические БД, Магазин приложений, Большие данные, Графические адаптеры, ML Platform, AI API, and Настройки меню. The main area is titled 'Панель управления' (Control Panel) and displays resource usage: Истории (1 из 4 шт), CPU (1 из 8 шт), RAM (1 из 16 ГБ), Объем (10 из 100 ГБ), and Диски (1 из 10 шт). A callout box highlights the 'Создать инстанс' (Create instance) button under the 'Виртуальные машины' (Virtual machines) section. Other sections include S3 объектное хранилище, Контейнеры Kubernetes, and Базы данных.

## 3. Желательно выбираем Ubuntu

The screenshot shows the 'Создание нового инстанса' (Create new instance) page. It's step 1 of 3: Конфигурация (Configuration). The configuration fields include: Имя виртуальной машины (Virtual machine name: Ubuntu\_STD2-1-1\_10GB), Категория виртуальной машины (Category: Все актуальные типы виртуальных машин), Тип виртуальной машины (Type: STD2-1-1), Зона доступности (Availability zone: Москва (MS1)), Количество машин в конфигурации (Number of machines: 1), and Размер диска (Disk size: 10 ГБ). To the right, a summary table shows the configuration details and a note about the cost calculation. The total cost is listed as 983 ₽ за 1 месяц (983 ₽ for 1 month).

## 4. Делаем следующие шаги и все, инстанс создан

## 5. В разделе 'Виртуальные сети'-'Настройки firewall' добавляем новую настройку firewall

Группа	Инстансы	Кол-во правил	Дата изменения
default fe14a888-d1f7-48c9-851b-536ddb3cc981	1	2	17 февр. 2024, 09:06
ssh 107ab5ac-bb4d-48ae-9277-86fe74189497	1	1	17 февр. 2024, 09:07
web 33afb8f7-8419-4758-8b76-b15a09ab2b92	1	2	17 февр. 2024, 09:40

## 6. Обзываем web

**Создание группы правил**

Имя группы правил: **web**

Описание:

**Создать группу**      Отменить

## 7. Делаем настройку

**Настройки**

## 8. Добавляем 2 новых правила

VK Cloud

mcs4064050056

VK Cloud > Cloud Networks > Настройки firewall > web

**web**

**Входящий трафик**

Тип	Протокол	Порты	Источник	Группа	Описание
IPv4	TCP	443	0.0.0.0/0	—	...
IPv4	TCP	8000	0.0.0.0/0	—	...

+ Добавить правило

**Исходящий трафик**

Отсутствуют правила

+ Добавить правило

Виртуальные машины с группой правил web

### Новое входящее правило

Тип

HTTP

Протокол

TCP

Порты

8000

например, 8080 или 10 - 65535

Удаленный адрес

Все IP-адреса    Диапазон IP-адресов    Группа безопасности

Добавить описание

**Сохранить правило**    Отменить

### Новое входящее правило

Тип

HTTPS

Протокол

TCP

Порты

443

например, 8080 или 10 - 65535

Удаленный адрес

Все IP-адреса    Диапазон IP-адресов    Группа безопасности

Добавить описание

**Сохранить правило**    Отменить

## 9. Добавляем правило к сети

**Ubuntu\_STD2-1-1\_10GB**

Сети

Имя сети	Имя подсети	IP-адрес	Доменное имя	Настройки Firewall
internet	ext-sub2 Шлюз: 212.111.87.254 CIDR: 212.111.84.0/22	212.111.84.6	- MAC-адрес: fa:16:3e:5d:8d:27	web, default, ssh ... Редактировать подключение Удалить подключение

**Настройка подключения**

Для правильной настройки подключения прочтите [инструкцию](#).

**Все документация**  
Практические руководства и пошаговые инструкции по работе с VK Cloud

**Диагностика VM**  
Управление виртуальной машиной с помощью VNC-консоли. Просмотр логов сообщений VM

**Диагностика и устранение проблем**  
Способы диагностики и устранения проблем с виртуальными машинами

**Шифрование диска**  
Как настроить шифрование машины с помощью

**Ubuntu\_STD2-1-1\_10GB**

Сети

**Редактирование подключения**

Имя:

Сеть для подключения:  Внешняя сеть (internet)

Назначить внешний IP:

Настройки Firewall:  default x ssh x web

**Сохранить**   **Отмена**

Это правило нужно для того, чтобы сайт вскоре был доступен

по <http://212.111.84.6:8000/>

## 1. При помощи данной команды подключаемся по ssh к удаленной виртуальной машине

The screenshot shows the VK Cloud interface under the 'Virtual machines' section. It displays the external IP address (212.111.84.6) and a note that an internal IP does not exist. It provides instructions for connecting via SSH:

1. На прошлом шаге вы выбрали уже созданный ключ от виртуальной машины. Вам нужно найти файл от него на вашем компьютере. Этот файл загрузился, когда вы создавали ключ в одном из предыдущих запусков инстанса.
2. Скопируйте путь к этому файлу.
3. Делаем ключ доступным только для текущего пользователя. Если на вашем компьютере установлена операционная система Linux, Unix или MacOS, то выполните команду:

```
chmod 400 <путь к ключу>
```

4. Первое подключение к инстансу возможно только по протоколу SSH. Для Linux, Unix или MacOS используйте команду:

```
ssh -i <путь к ключу> ubuntu@212.111.84.6
```

5. Чтобы получить права root-пользователя, пропишите команду:

```
sudo bash
```

More detailed instructions can be found in the documentation.

По следующему адресу кстати будет доступен сайт

The terminal window shows the user connecting via SSH to the IP 212.111.84.6. The user has logged in as the root user (1 user). The system information is displayed, including the date (Sat Feb 17 16:53:46 UTC 2024), load average (0.0), memory usage (27.6% of 9.31GB), and swap usage (0%). The user then runs the command `sudo bash`.

Там как и в обычной линуке делаем sudo bash, чтобы удобнее было.

Потом клонируем репозиторий, запускаем контейнер при помощи docker-compose up -d

(-d чтобы работа терминала не зависала только на этом процессе)

далее по необходимости делаем docker exec -it vite\_docker sh

чтобы войти в контейнер и выполнить команды npm i и потом npm run dev

Все, сервер поднялся, теперь конфигурируем сам nginx.

2. В папке conf.d делаем конфигурацию при помощи команды nano

/etc/nginx/conf.d/resume.conf



```
ubuntu@ubuntu-std2-1-1-10g ~ % nano /etc/nginx/conf.d/resume.conf
GNU nano 2.9.3
server {
    listen 80;
    server_name egorkerber.ru; # или ваш домен, если он есть

    location / {
        proxy_pass http://localhost:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_redirect off;
    }
}

[ File '/etc/nginx/conf.d/resume.conf' is unwritable ]
^G Get Help      ^O Write Out      ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos      M-U Undo
^X Exit          ^R Read File      ^N Replace      ^U Uncut Text     ^T To Spell      ^L Go To Line   M-E Redo
                                         M-A Mark Text      M-G Copy Text
```

3. Делаем более общую конфигурацию nano /etc/nginx/nginx.conf

```
1 user www-data;
2
3 worker_processes auto;
4
5 pid /run/nginx.pid;
6
7 include /etc/nginx/modules-enabled/*.conf;
8
9
10
11 events {
12     worker_connections 768;
13     # multi_accept on;
14 }
15
16
17 http {
18     ##
19     # Basic Settings
20     ##
21
22     sendfile on;
23     tcp_nopush on;
```

```
18     tcp_nodelay on;
19
20     keepalive_timeout 65;
21
22     types_hash_max_size 2048;
23
24
25     ##
26
27     # SSL Settings
28
29     ##
30
31
32     # Logging Settings
33
34     ##
35
36     access_log /var/log/nginx/access.log;
37     error_log /var/log/nginx/error.log;
38
39     ##
40
41     # Gzip Settings
42
43     ##
44
45     gzip on;
```

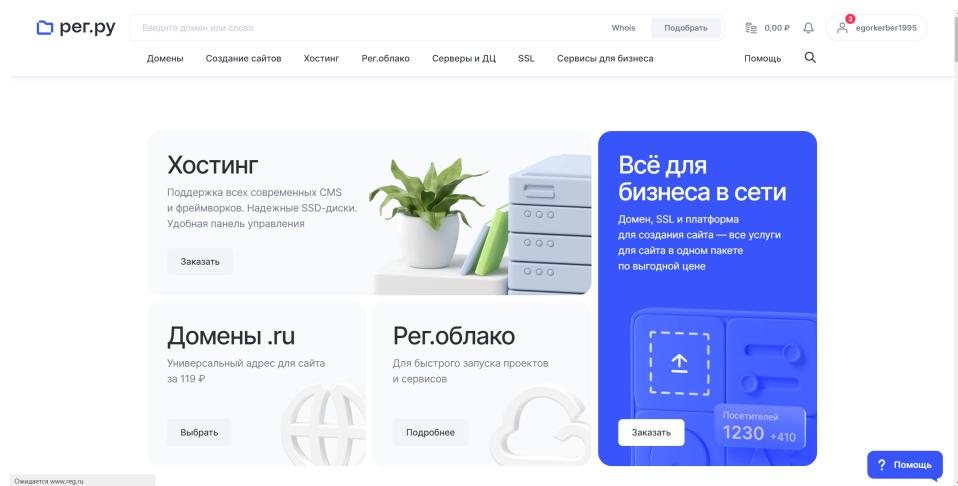
```
45     ##
46     # Virtual Host Configs
47     ##
48
49     include /etc/nginx/conf.d/*.conf;
50     include /etc/nginx/sites-enabled/*;
51 }
52
```

4. Выполняем команду systemctl restart nginx чтобы запустить nginx

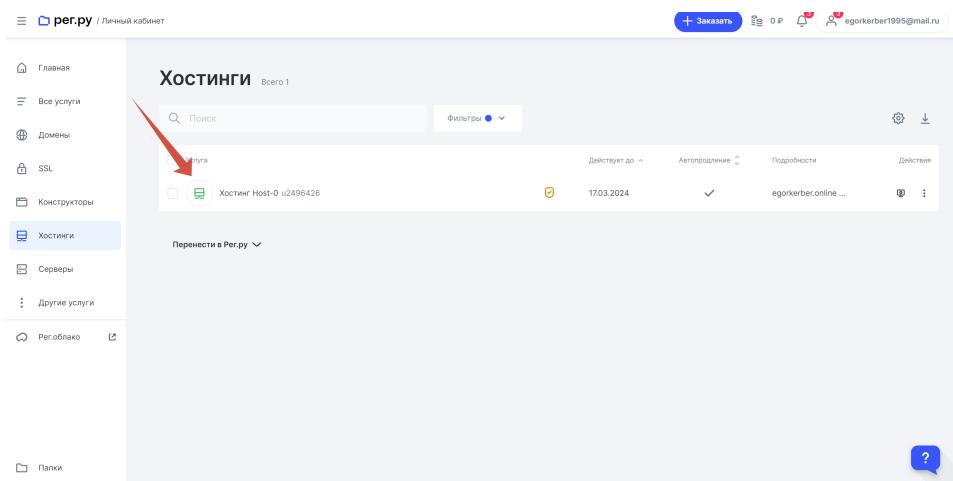
3

## ПОКУПКА ДОМЕНА

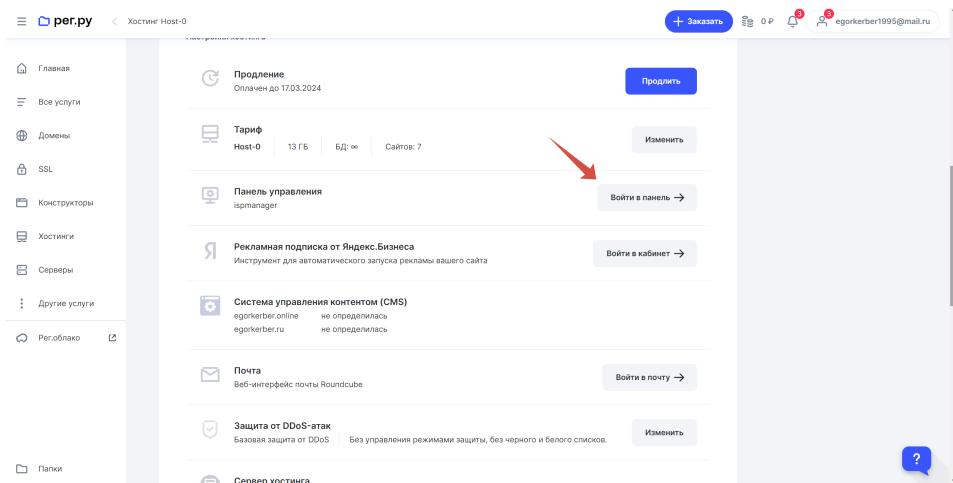
### 1. Заходим на рег.ру, регистрируемся и т.п. и покупаем домен



### 2. Потом заходим в личный кабинет, хостинги



### 3. Заходим в нужный хостинг, мотаем вниз, заходим в панель управления



## 4. Заходим в управление DNS

The screenshot shows the ispmanager dashboard with the 'Управление DNS' (DNS Management) section selected. On the left sidebar, there is a red arrow pointing to the 'Управление DNS' link under the 'DNS Management' heading. The main area displays a 'Дашборд' (Dashboard) with sections for 'Использование диска' (Disk Usage) and 'Журнал посещений' (Access Log). To the right, there is a 'Ограничения' (Limits) table showing resource usage limits.

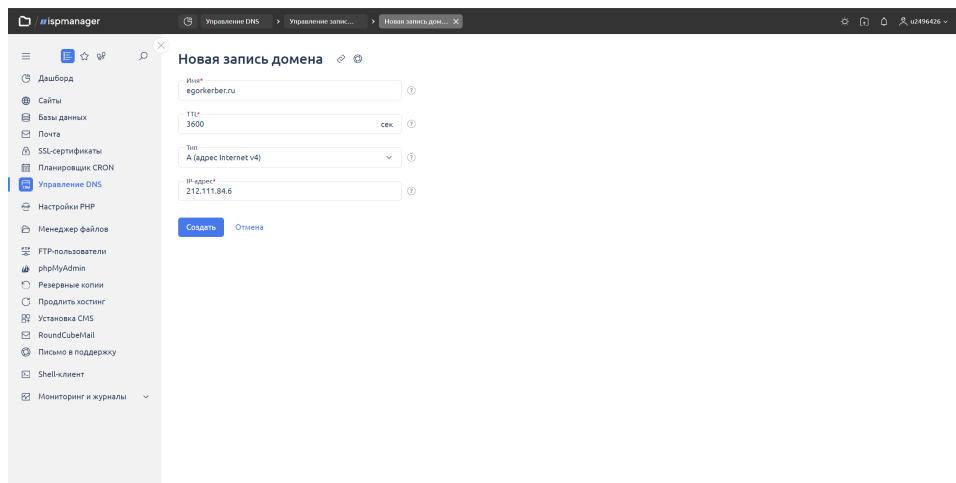
## 5. Управлять DNS записями

The screenshot shows the 'Управление DNS' (DNS Management) page. A red arrow points to the 'Управление DNS записями' (Manage DNS records) button in the top right corner of the main content area. The page lists domain records for 'egorkerber.online' and 'egorkerber.ru'. For 'egorkerber.ru', the 'Изменить' (Edit) and 'Управление DNS записями' (Manage DNS records) buttons are highlighted.

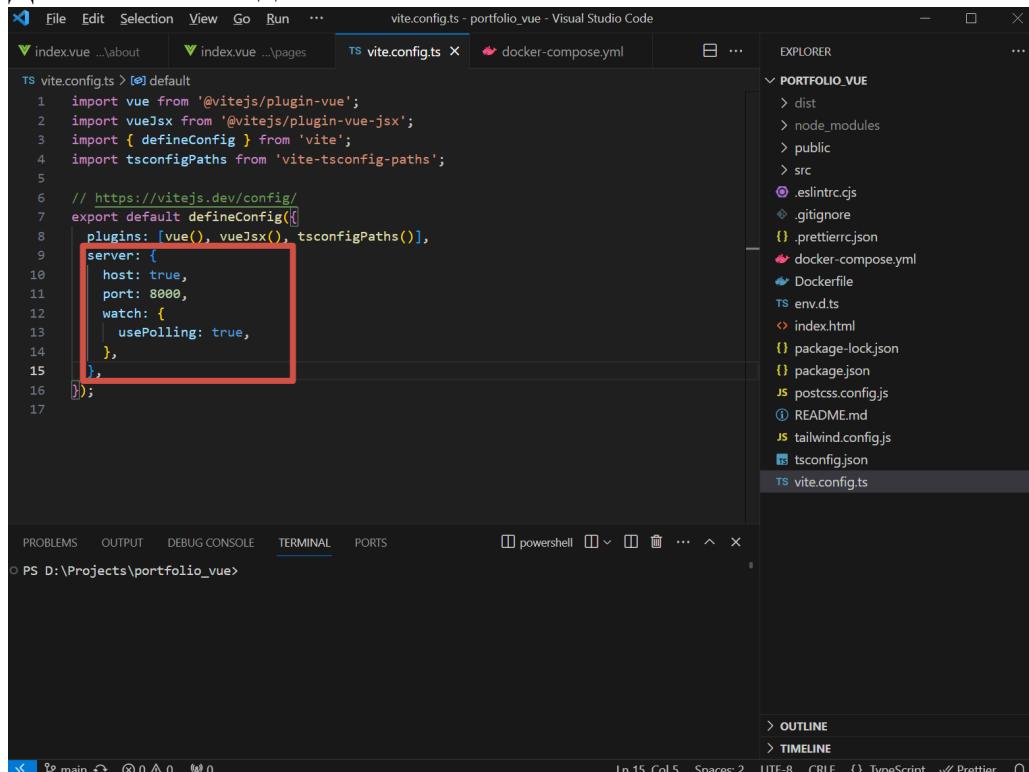
## 6. Создаем запись

The screenshot shows the 'Управление записями домена - egorkerber.ru' (Manage domain records - egorkerber.ru) page. A red arrow points to the 'Создать запись' (Create record) button at the top left. The main area displays a table of existing DNS records for the 'egorkerber.ru' domain, including entries for A, AAAA, MX, and NS records.

## 7. Делаем так



Для vite важно добавить это поле



```

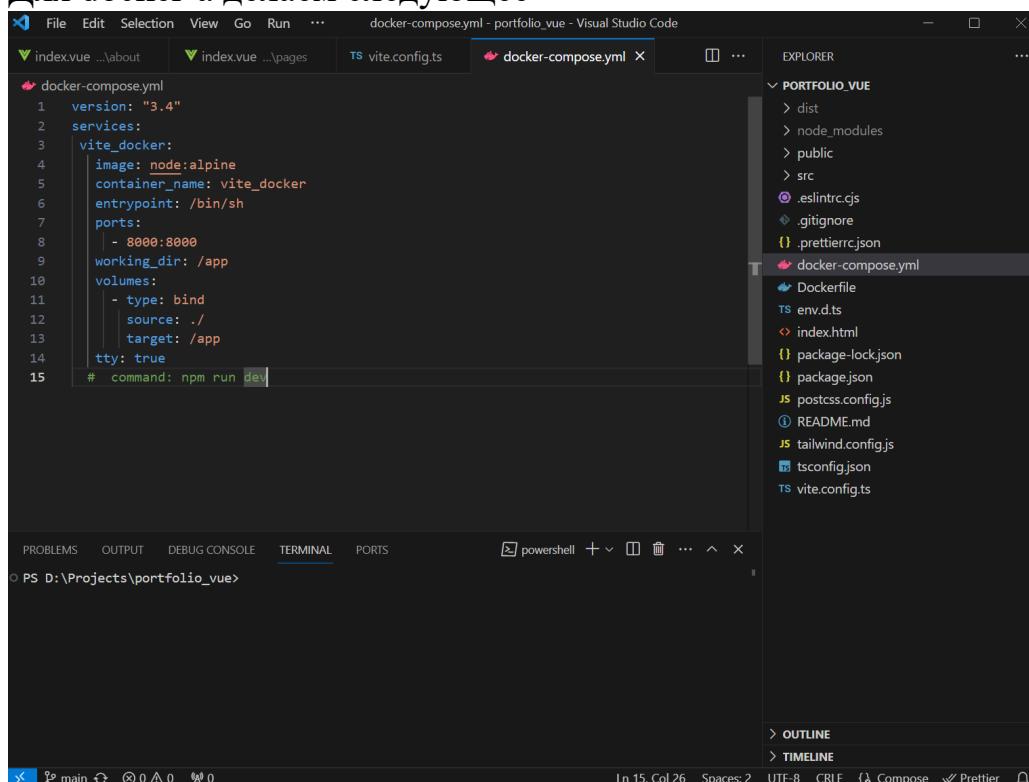
1 import vue from '@vitejs/plugin-vue';
2 import vueJsx from '@vitejs/plugin-vue-jsx';
3 import { defineConfig } from 'vite';
4 import tsconfigPaths from 'vite-tsconfig-paths';

5 // https://vitejs.dev/config/
6 export default defineConfig({
7   plugins: [vue(), vueJsx(), tsconfigPaths()],
8   server: {
9     host: true,
10    port: 8000,
11    watch: {
12      usePolling: true,
13    },
14  },
15);
16
17

```

В порте указываем внешний порт (watch usePolling добавил на всякий случай, нужен вроде чтобы hot reload работал или что-то такое)

Для docker-а делаем следующее



```

version: "3.4"
services:
  vite_docker:
    image: node:alpine
    container_name: vite_docker
    entrypoint: /bin/sh
    ports:
      - 8000:8000
    working_dir: /app
    volumes:
      - type: bind
        source: ../
        target: /app
    tty: true
#   command: npm run dev

```

## 4.1 | Объяснение конфигурации от ChatGPT

Этот файл является файлом конфигурации Docker Compose, который используется для определения и настройки многоконтейнерных приложений. Давайте разберем каждую часть этого файла:

`version: "3.4"`: Это указывает на версию формата файла Docker Compose, которую следует использовать при интерпретации этого файла. В данном случае используется версия 3.4.

`services::` Этот ключевой раздел определяет список всех сервисов, которые будут запущены при помощи Docker Compose.

`vite_docker::` Это имя сервиса, которое мы определяем. Здесь `vite_docker` – это произвольное имя, которое мы дали нашему сервису.

`image: node:alpine`: Это образ Docker, который будет использоваться для запуска нашего сервиса. Здесь мы используем образ `node:alpine`, основанный на Alpine Linux, который содержит Node.js.

`container_name: vite_docker`: Это имя контейнера, которое будет присвоено контейнеру, когда он будет запущен.

`entrypoint: /bin/sh`: Это команда, которая будет выполнена при запуске контейнера. Здесь мы указываем запуск интерактивной оболочки `sh` в контейнере.

`ports::` Этот параметр определяет порты, которые будут проброшены из контейнера на хостовую машину. В данном случае, порт 8000 контейнера будет проброшен на порт 8000 хостовой машины.

`working_dir: /app`: Это рабочий каталог внутри контейнера, к которому будет сделано изменение после запуска контейнера.

`volumes::` Этот параметр позволяет примонтировать локальные файловые системы или тома внутрь контейнера. Здесь используется тип `bind`, что означает привязку (`mount`) каталога или файла с хостовой машины к каталогу в контейнере.

`type: bind`: Это тип тома, который мы используем для монтирования файло-

вой системы хоста.

source: ./: Это путь к каталогу или файлу на хостовой машине, который будет примонтирован в контейнер.

target: /app: Это место в контейнере, куда будет примонтирован каталог или файл с хостовой машины.

tty: true: Этот параметр устанавливает взаимодействие с терминалом (TTY) внутри контейнера, что обеспечивает возможность использования интерактивной оболочки внутри контейнера.

Этот файл Docker Compose определяет контейнер, который будет запущен с образом Node.js, пробрасывает порт 8000, монтирует текущий рабочий каталог хостовой машины внутрь контейнера и запускает интерактивную оболочку sh внутри контейнера.

## Ссылки

<https://dev.to/ysmnikhil/how-to-build-with-react-or-vue-with-vite-and-docker-1a3l>