

# Ansible

Дано:

- Образы VM на yandex cloud (собранные Packer-ом)
- Подготовлены и развернуты VM на yandex cloud (инстансы описаны в Terraform)

## Задание 1

С помощью ansible развернуть и настроить на сервере app rybi, puma.service, а на сервере db mongodb.

В итоге по ip app на порту 9292 должно быть приложение.

## Задание 2

Перенести созданные плейбуки в отдельные роли.

Использовать коммьюнити роль nginx

В итоге приложение должно быть на 80 порту.

Задание 1 в файле ansible-Часть 1.

Задание 2

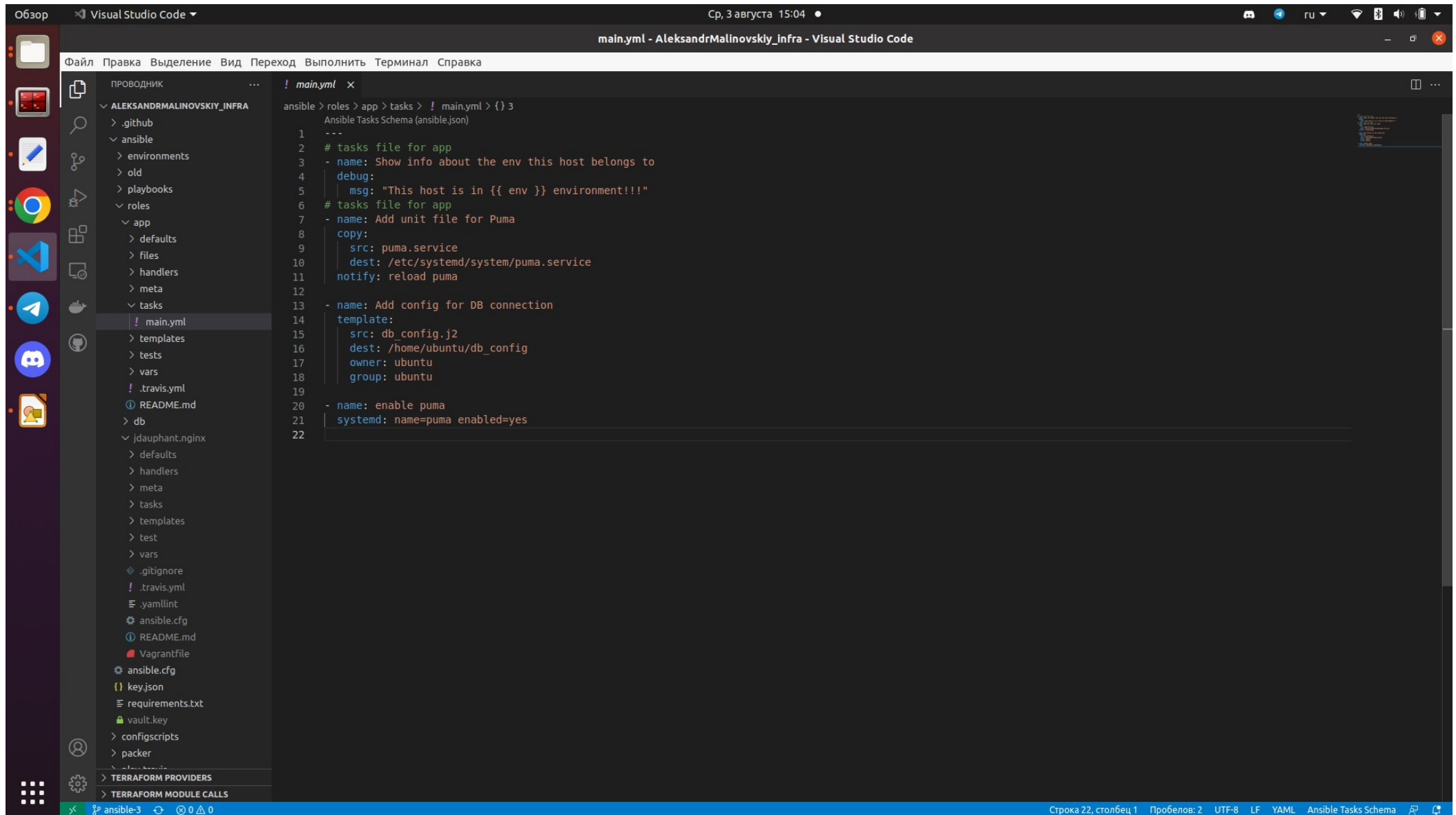
- Перенести созданные плейбуки в отдельные роли.
- Использовать коммьюнити роль `nginx` что бы приложение было на 80 порту.

Определим с помощью ansible-galaxy init роль app и db.  
Создается общепринятая структура папок для ролей.

```

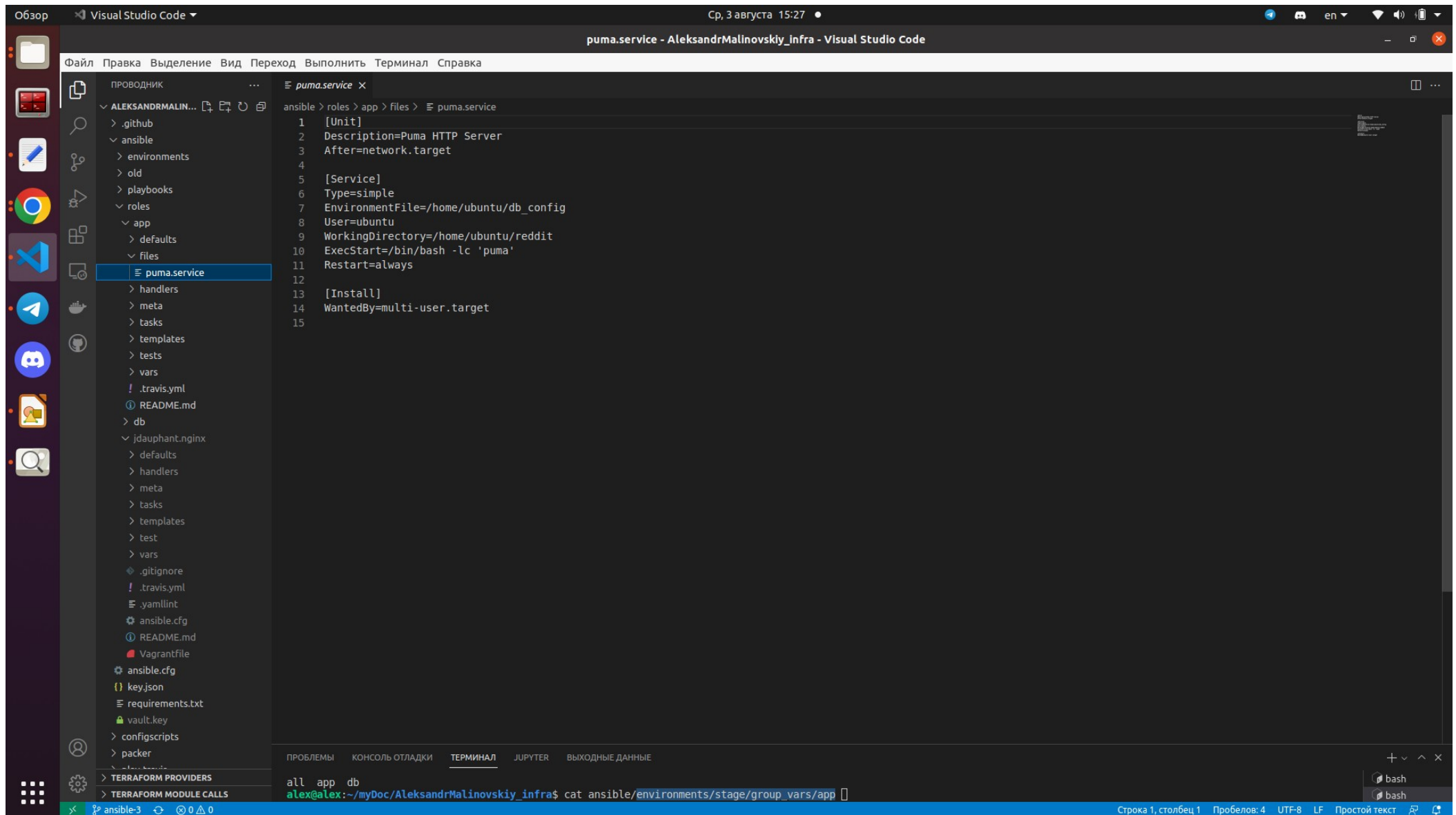
└─ roles
  └─ app
    > defaults
    > files
    > handlers
    > meta
    > tasks
    > templates
    > tests
    > vars
    ! .travis.yml
    ⓘ README.md
  └─ db
    > defaults
    > files
    > handlers
    > meta
    > tasks
    > templates
    > tests
    > vars
    ! .travis.yml
    ⓘ README.md
  └─ jdauphant.nginx
    > defaults
    > handlers
    > meta
    > tasks
    > templates
    > test
    > vars
```

В файл `roles/app/tasks/main` пишем task на которая будет выводить информацию о том, в каком окружении находится конфигурируемый хост, перенесет файл `puma.service` и `db.config`. Информацию о том в каком мы окружении ansible возьмет из `env`.

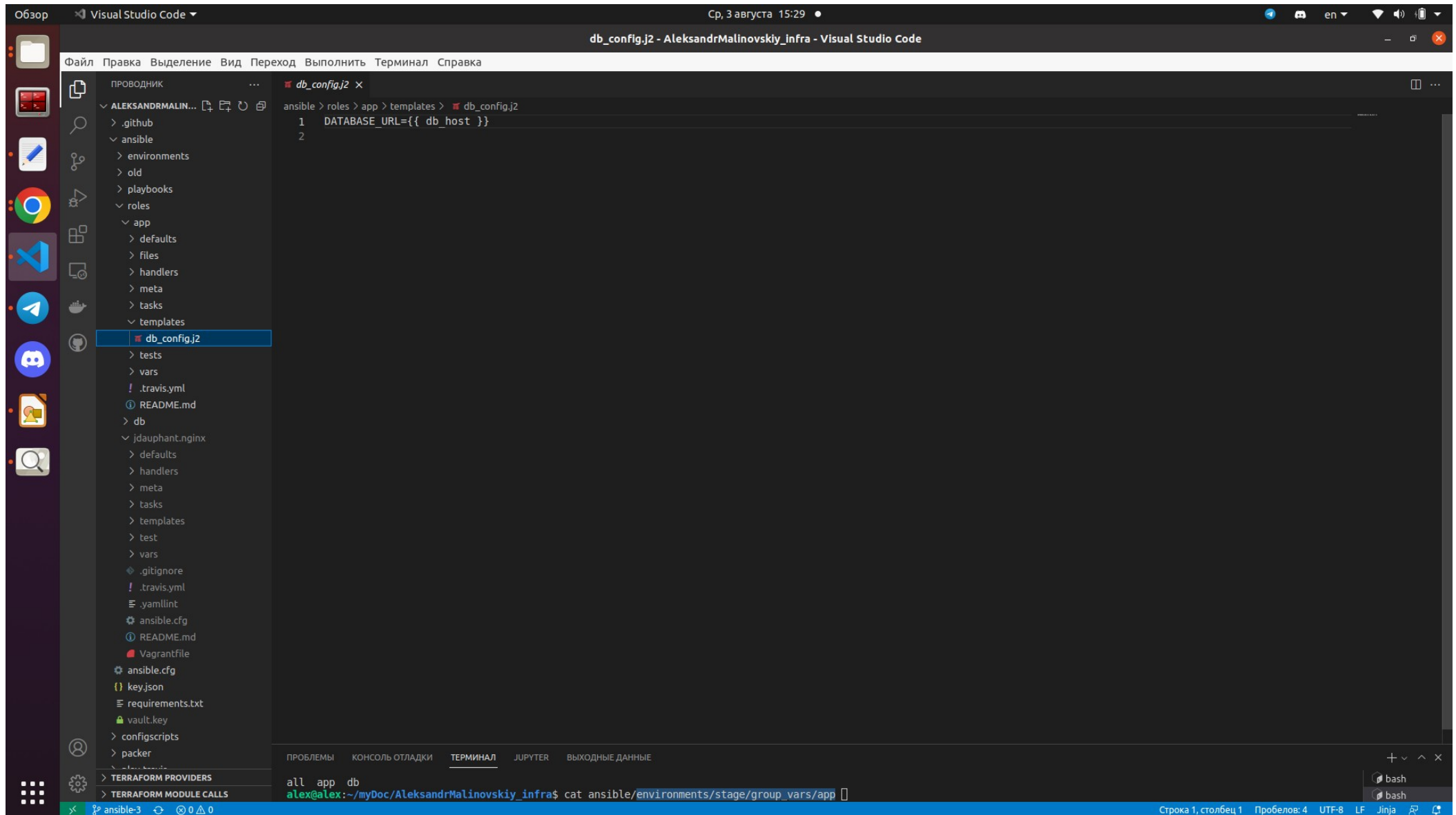


```
ansible > roles > app > tasks > ! main.yml > {} 3
Ansible Tasks Schema (ansible.json)
1  ---
2  # tasks file for app
3  - name: Show info about the env this host belongs to
4    debug:
5      msg: "This host is in {{ env }} environment!!!"
6  # tasks file for app
7  - name: Add unit file for Puma
8    copy:
9      src: puma.service
10     dest: /etc/systemd/system/puma.service
11     notify: reload puma
12
13  - name: Add config for DB connection
14    template:
15      src: db_config.j2
16      dest: /home/ubuntu/db_config
17      owner: ubuntu
18      group: ubuntu
19
20  - name: enable puma
21    systemd: name=puma enabled=yes
22
```

# Переносим файл puma.service в директорию roles/app/file/



Переносим файл шаблон db\_config в директорию roles/app/templates/. В данном файле мы указываем переменную db\_host. Для хранения переменных в одном месте создана директория environments/stage/group\_vars/ и переменная описана в файле app.

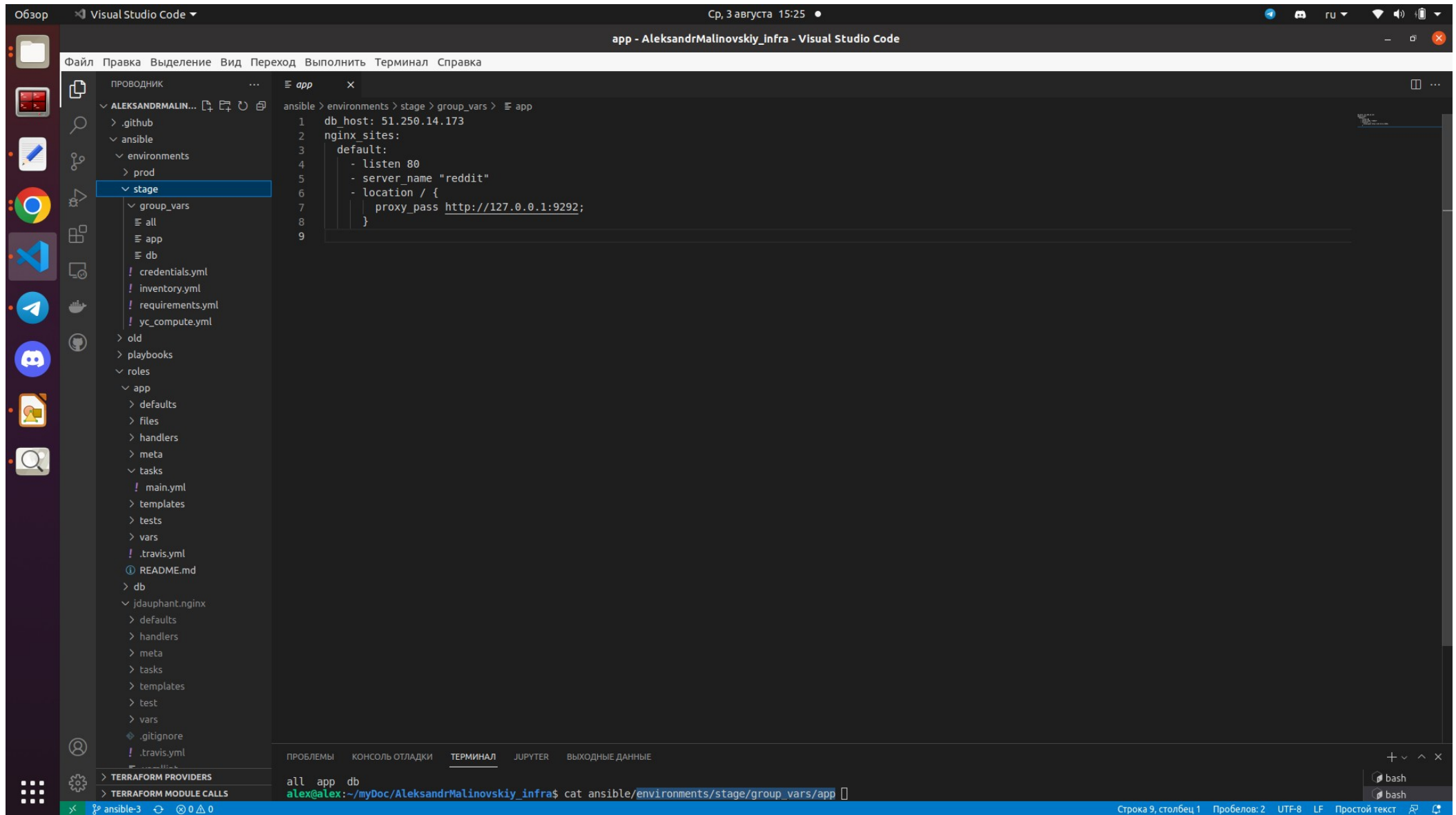


The screenshot shows the Visual Studio Code interface. The file explorer on the left displays the project structure, with the file `db_config.j2` selected under the path `roles/app/templates/`. The editor window shows the content of `db_config.j2`, which is an Ansible Jinja2 template. The terminal at the bottom shows the command `cat ansible/environments/stage/group_vars/app` being executed, with the output `all app db`.

```
ansible > roles > app > templates > db_config.j2
1 DATABASE_URL={{ db_host }}
2
```

```
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
all app db
```

Файл app расположенный в environments/stage/group\_vars/. В файле указаны переменные используемые для хостов app. На этом этапе мы начинаем разделять окружения stage и prod. Они будут одинаковые.

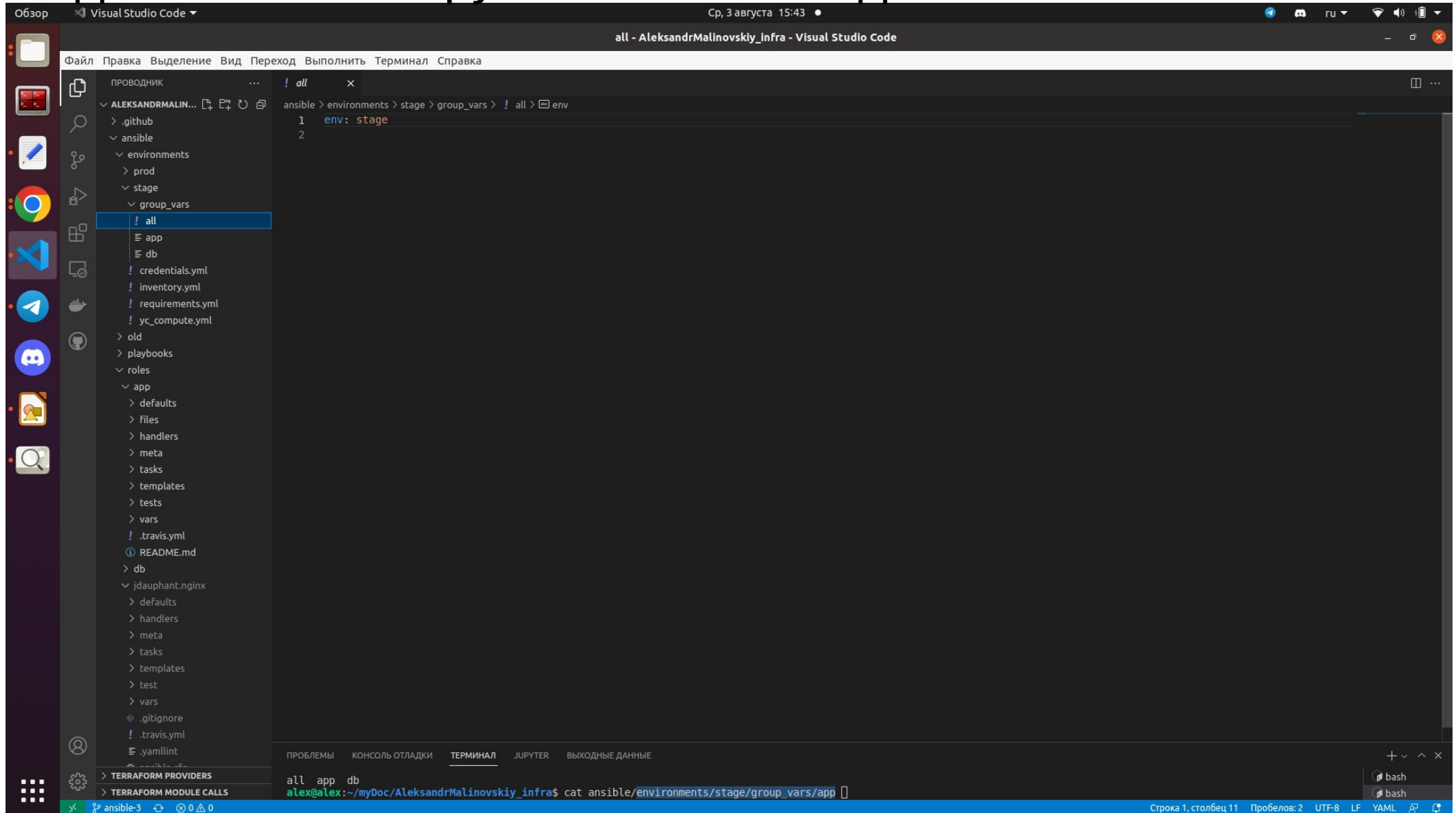


The screenshot shows the Visual Studio Code interface. The file explorer on the left shows the project structure, with the 'stage' directory under 'environments' selected. The editor displays the file 'app' located at 'environments/stage/group\_vars/app', containing Ansible YAML code for nginx configuration. The terminal at the bottom shows the command to view the file's contents.

```
ansible > environments > stage > group_vars > app
1 db_host: 51.250.14.173
2 nginx_sites:
3   default:
4     - listen 80
5     - server_name "reddit"
6     - location / {
7       proxy_pass http://127.0.0.1:9292;
8     }
9
```

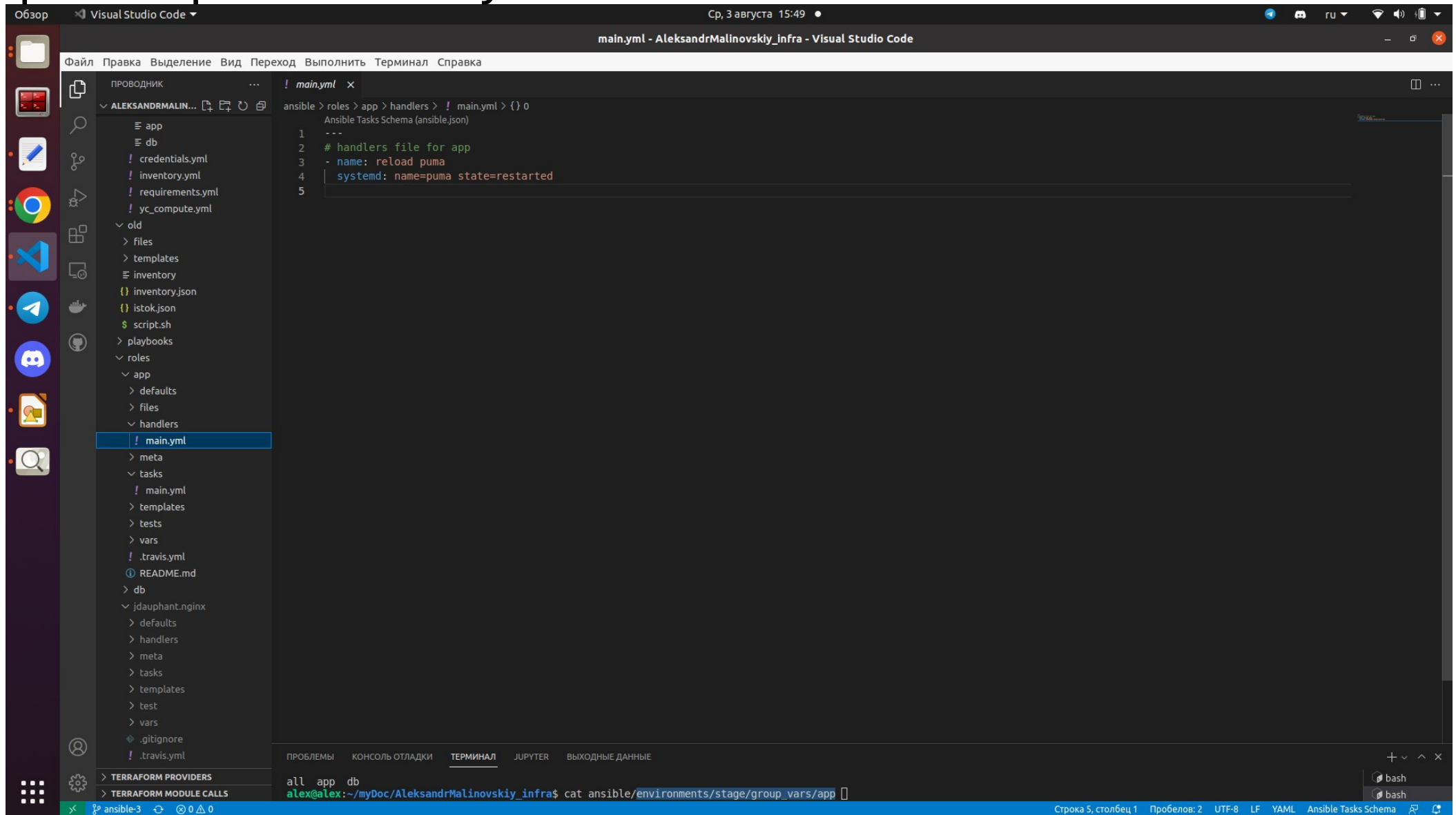
```
all app db
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
```

Файл `all` расположенный в `environments/stage/group_vars/`.  
В файле указан именно тот `env` по которому мы будем  
видеть в каком окружении мы находимся.





Файл main директория roles/app/handlers/.  
В файле указан перезапуск puma.service по триггеру reload  
puma прописанному в tasks/main.



main.yml - AleksandrMalinovskiy\_infra - Visual Studio Code

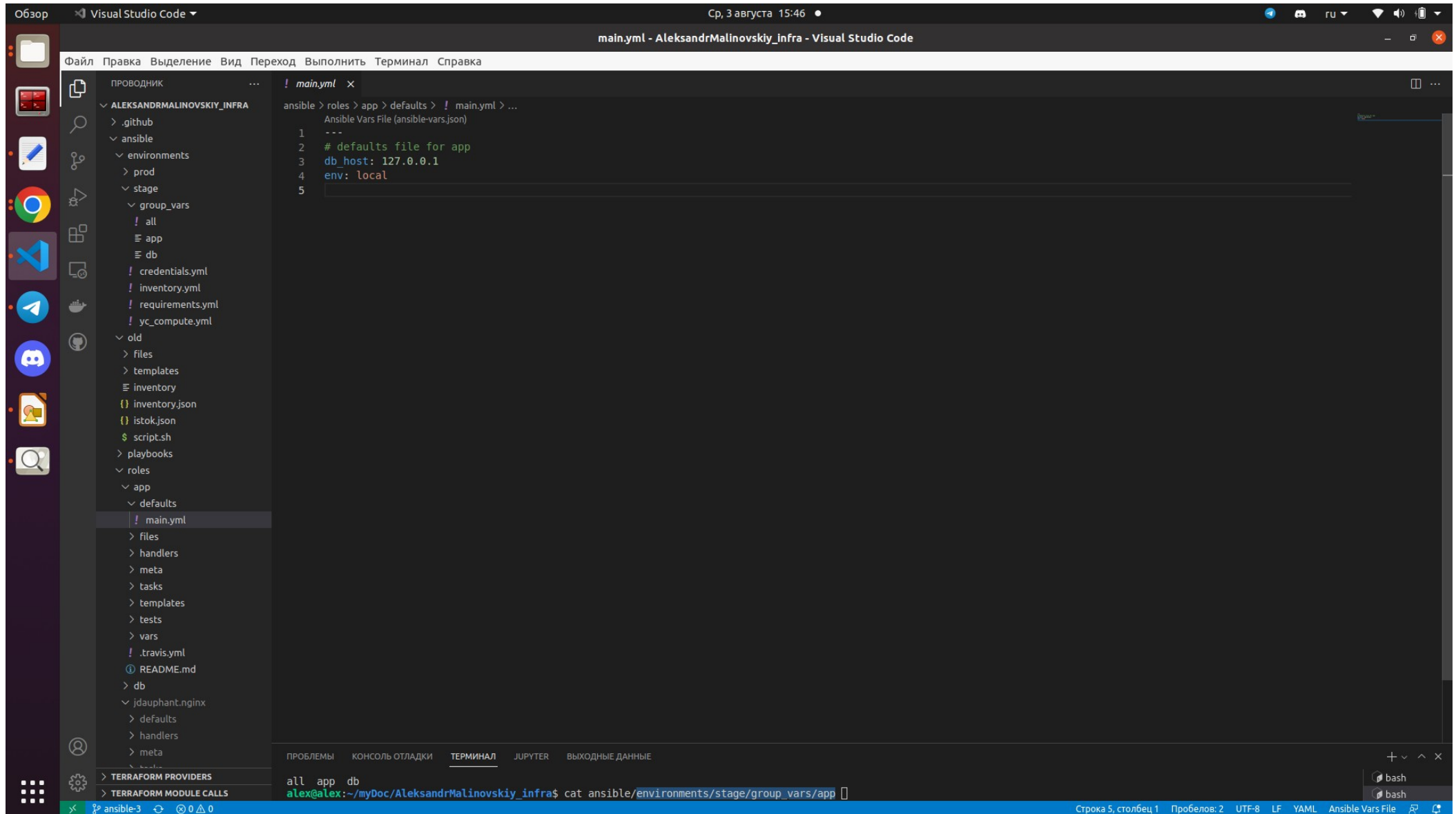
```
1 ---
2 # handlers file for app
3 - name: reload puma
4   systemd: name=puma state=restarted
5
```

ПРОБЛЕМЫ КОНСОЛЬ ОТЛАДКИ **ТЕРМИНАЛ** JUPYTER ВЫХОДНЫЕ ДАННЫЕ

```
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
```

Строка 5, столбец 1 Пробелов: 2 UTF-8 LF YAML Ansible Tasks Schema

Файл main в деректории roles/app/defaults.  
В файле указаны переменные по умолчанию.



Обзор Visual Studio Code

main.yml - AleksandrMalinovskiy\_infra - Visual Studio Code

Файл Правка Выделение Вид Переход Выполнить Терминал Справка

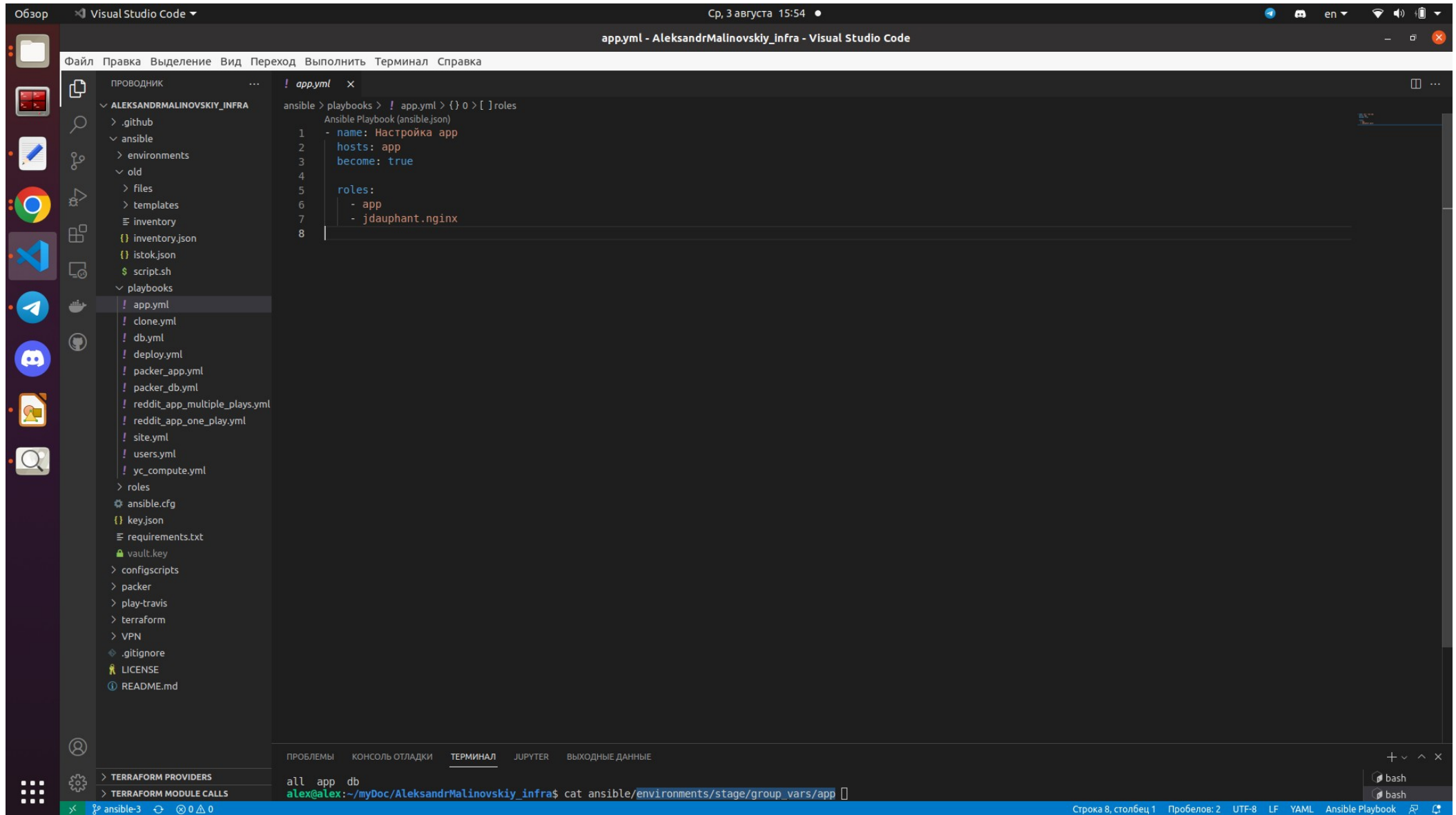
ПРОБЛЕМЫ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ JUPYTER ВЫХОДНЫЕ ДАННЫЕ

```
ansible > roles > app > defaults > ! main.yml > ...
Ansible Vars File (ansible-vars.json)
1 ---
2 # defaults file for app
3 db_host: 127.0.0.1
4 env: local
5
```

alex@alex:~/myDoc/AleksandrMalinovskiy\_infra\$ cat ansible/environments/stage/group\_vars/app

Строка 5, столбец 1 Пробелов: 2 UTF-8 LF YAML Ansible Vars File

Сам плейбук app с вызовом роли app и роли jdauphant.nginx.  
Роль jdauphant.nginx взята с коммьюнити. Используется для того  
что бы развернуть nginx на сервере app.



```
ansible > playbooks > ! app.yml > {} 0 > [ ] roles
Ansible Playbook (ansible.json)
1  - name: Настройка app
2    hosts: app
3    become: true
4
5    roles:
6      - app
7      - jdauphant.nginx
8
```

ПРОБЛЕМЫ КОНСОЛЬ ОТЛАДКИ **ТЕРМИНАЛ** JUPYTER ВЫХОДНЫЕ ДАННЫЕ

```
all app db
alex@alex:~/myDoc/AleksandrMalinovsky_infra$ cat ansible/environments/stage/group_vars/app
```

Строка 8, столбец 1 Пробелов: 2 UTF-8 LF YAML Ansible Playbook

Создадим файл requirements.yml в директории environments/stage/ для разделение зависимостей ролей по окружениям. Сейчас у нас уже два окружения stage и prod. Оба окружения одинаковые.

The screenshot shows the Visual Studio Code interface with the file explorer on the left, the editor in the center, and the terminal at the bottom. The file explorer shows the project structure with the 'requirements.yml' file highlighted in the 'environments/stage/' directory. The editor shows the content of 'requirements.yml' with the following YAML structure:

```
ansible > environments > stage > ! requirements.yml > {} 0 > version
Ansible Requirements (ansible-requirements.json)
1 - src: jdauphant.nginx
2   version: v2.21.1
3
```

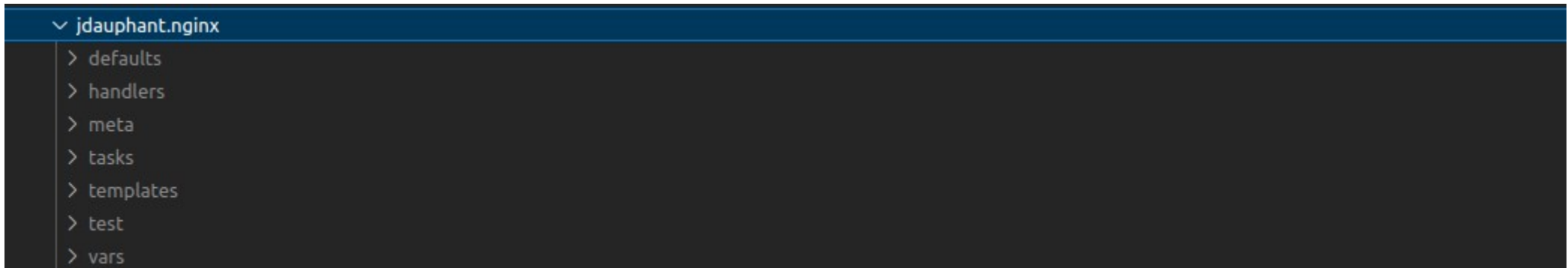
The terminal at the bottom shows the command executed to create the file:

```
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
```

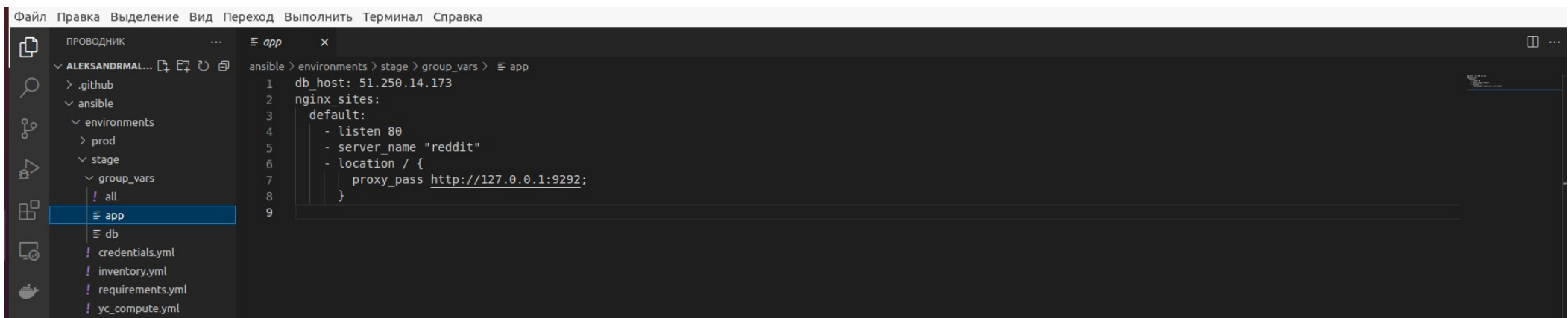
The status bar at the bottom indicates the file is 'requirements.yml' in the 'environments/stage/' directory, with a UTF-8 encoding and LF line endings.

## Команда

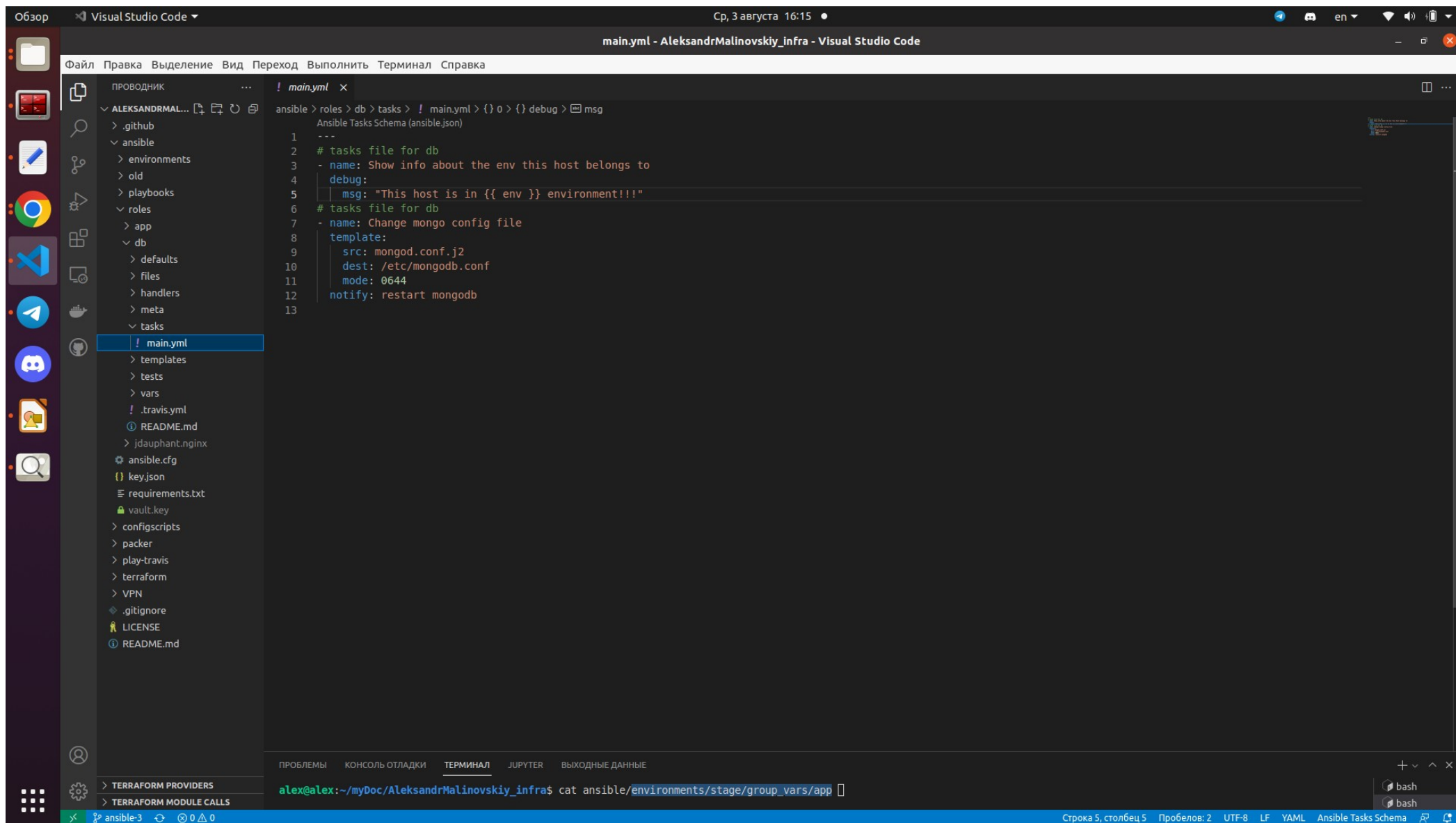
`ansible-galaxy install -r environments/stage/requirements.yml`  
установит роль и создаст директории с нужными файлами.  
Для минимальной настройки проксирования необходимо лишь  
добавить переменные в файл `app` в директории  
`environments/stage/group_vars/`



Файл `app` в директории `environments/stage/group_vars/`



В файл `roles/db/tasks/main` пишем task на которая будет выводить информацию о том, в каком окружении находится конфигурируемый хост, перепишет конфиг `mongodb` файлом `mongod.conf`. Информацию о том в каком мы окружении `ansible` возьмет из `env`.

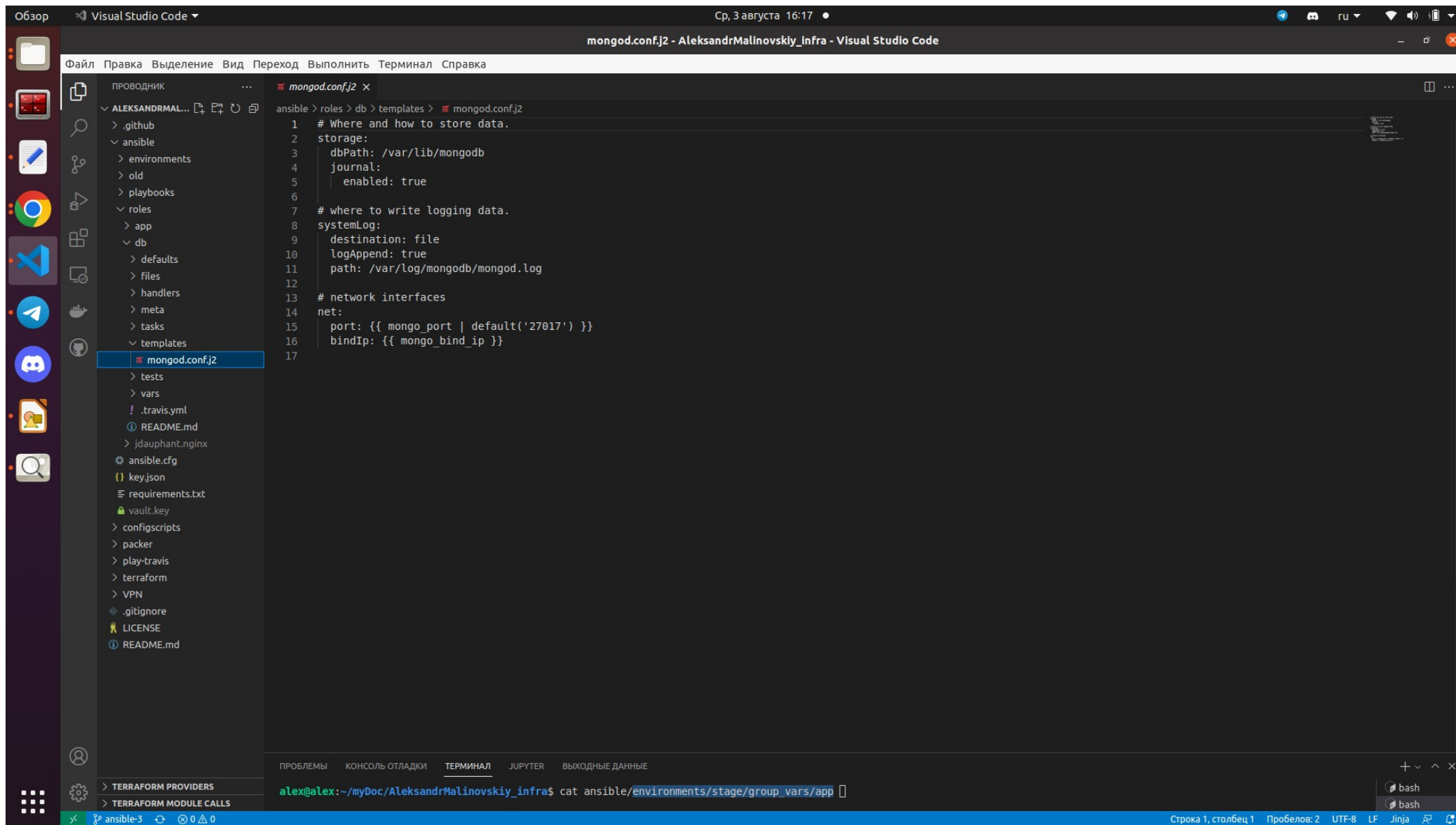


```
ansible > roles > db > tasks > ! main.yml > {} 0 > {} debug > msg
Ansible Tasks Schema (ansible.json)
1 ---
2 # tasks file for db
3 - name: Show info about the env this host belongs to
4   debug:
5     msg: "This host is in {{ env }} environment!!!"
6 # tasks file for db
7 - name: Change mongo config file
8   template:
9     src: mongod.conf.j2
10    dest: /etc/mongodb.conf
11    mode: 0644
12    notify: restart mongodb
13
```

```
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
```



Переносим файл шаблон `mongo.conf` в директорию `roles/db/templates/`. В данном файле мы указываем переменную `mongo_bind_ip`. Для хранения переменных в одном месте создана директория `environments/stage/group_vars/` и переменная описана в файле `db`.



Обзор Visual Studio Code

mongod.conf.j2 - AleksandrMalinovskiy\_infra - Visual Studio Code

Файл Правка Выделение Вид Переход Выполнить Терминал Справка

ПРОВОДНИК

- ALEKSANDRMALINOVSKIY\_INFRA
  - .github
  - ansible
    - environments
    - old
    - playbooks
  - roles
    - app
    - db
      - defaults
      - files
      - handlers
      - meta
      - tasks
      - templates
        - mongod.conf.j2
  - tests
  - vars
  - .travis.yml
  - README.md
  - jdauphant.nginx
  - ansible.cfg
  - key.json
  - requirements.txt
  - vault.key
  - configs
  - packer
  - play-travis
  - terraform
  - VPN
  - .gitignore
  - LICENSE
  - README.md

ansible > roles > db > templates > mongod.conf.j2

```
1 # Where and how to store data.
2 storage:
3   dbPath: /var/lib/mongodb
4   journal:
5     enabled: true
6
7 # where to write logging data.
8 systemLog:
9   destination: file
10  logAppend: true
11  path: /var/log/mongodb/mongod.log
12
13 # network interfaces
14 net:
15   port: {{ mongo_port | default('27017') }}
16   bindIp: {{ mongo_bind_ip }}
```

ПРОБЛЕМЫ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ JUPYTER ВЫХОДНЫЕ ДАННЫЕ

alex@alex:~/myDoc/AleksandrMalinovskiy\_infra\$ cat ansible/environments/stage/group\_vars/app

Строка 1, столбец 1 Пробелов: 2 UTF-8 LF Jinja

Файл db расположенный в environments/stage/group\_vars/.  
В файле указаны переменные используемые для хостов db.

Visual Studio Code interface showing the file explorer on the left and the code editor in the center. The file explorer displays the project structure, with the file `db` selected under the path `environments/stage/group_vars/`. The code editor shows the content of the `db` file, which is an Ansible inventory snippet defining the `mongo_bind_ip` variable as `0.0.0.0`.

```
ansible > environments > stage > group_vars > db
1  mongo_bind_ip: 0.0.0.0
2  |
```

The terminal at the bottom shows the command `cat ansible/environments/stage/group_vars/app` being executed, with the output `alex@alex:~/myDoc/AleksandrMalinovskiy_infra$`.



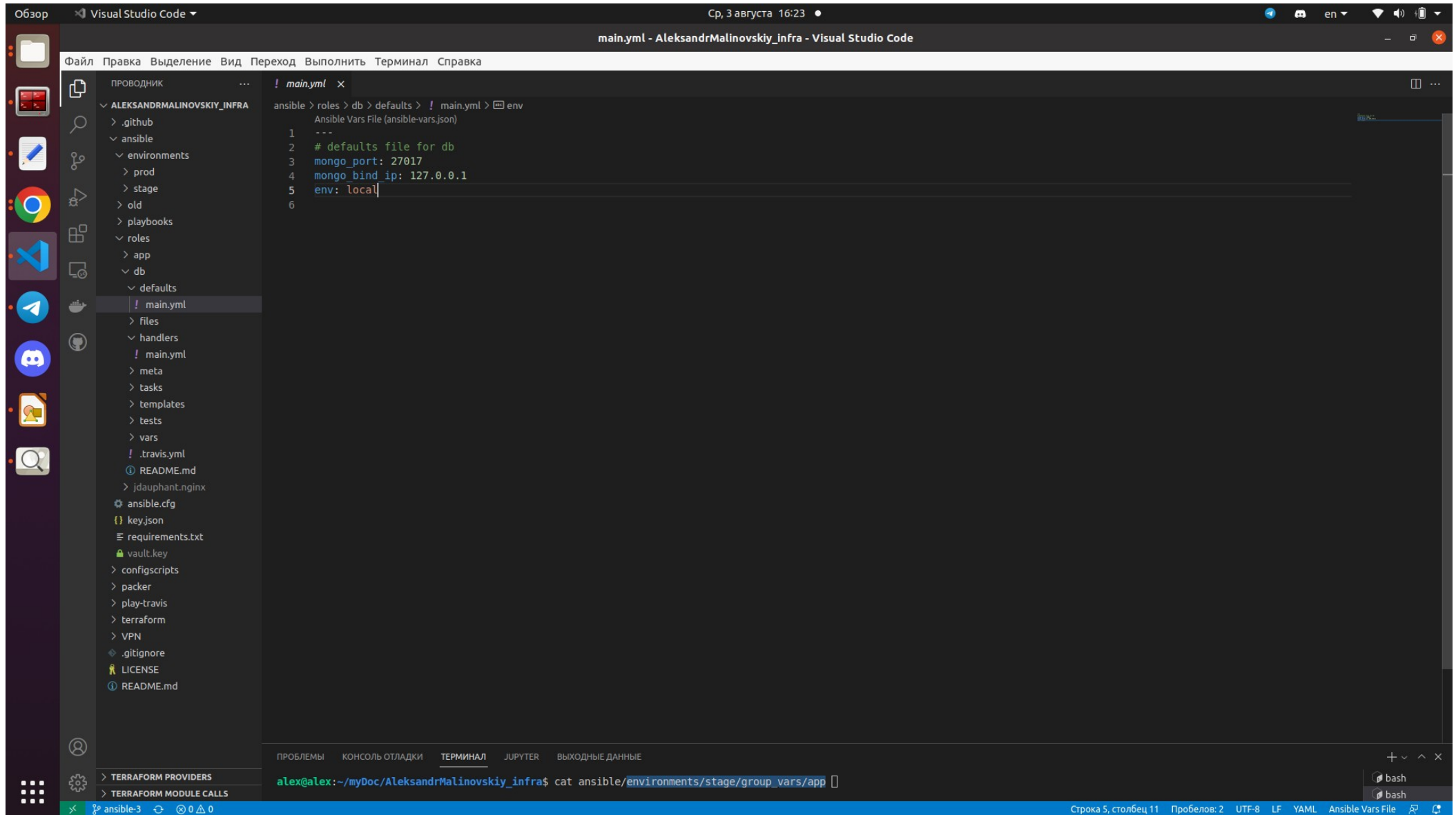
Файл main директория roles/db/handlers/.  
В файле указан перезапуск mongodb по триггеру restart  
mongodb прописанному в tasks/main.

The image shows a Visual Studio Code window titled "main.yml - AleksandrMalinovskiy\_infra - Visual Studio Code". The editor is open to a file named "main.yml" in the "handlers" directory of the "db" role. The file content is as follows:

```
ansible: roles: db: handlers: ! main.yml: {} 0: service:
  Ansible Tasks Schema (ansible.json)
  ---
  1:
  2: # handlers file for db
  3: - name: restart mongodb
  4: | service: name=mongodb state=restarted
  5:
```

The left sidebar shows the file explorer with the project structure. The "handlers" directory is selected, and "main.yml" is highlighted. The bottom status bar shows the terminal output: "alex@alex:~/myDoc/AleksandrMalinovskiy\_infra\$ cat ansible/environments/stage/group\_vars/app". The status bar also displays: "Строка 4, столбец 39 (выбрано 1) Пробелов: 2 UTF-8 LF YAML Ansible Tasks Schema".

Файл main в директории roles/db/defaults.  
В файле указаны переменные по умолчанию.



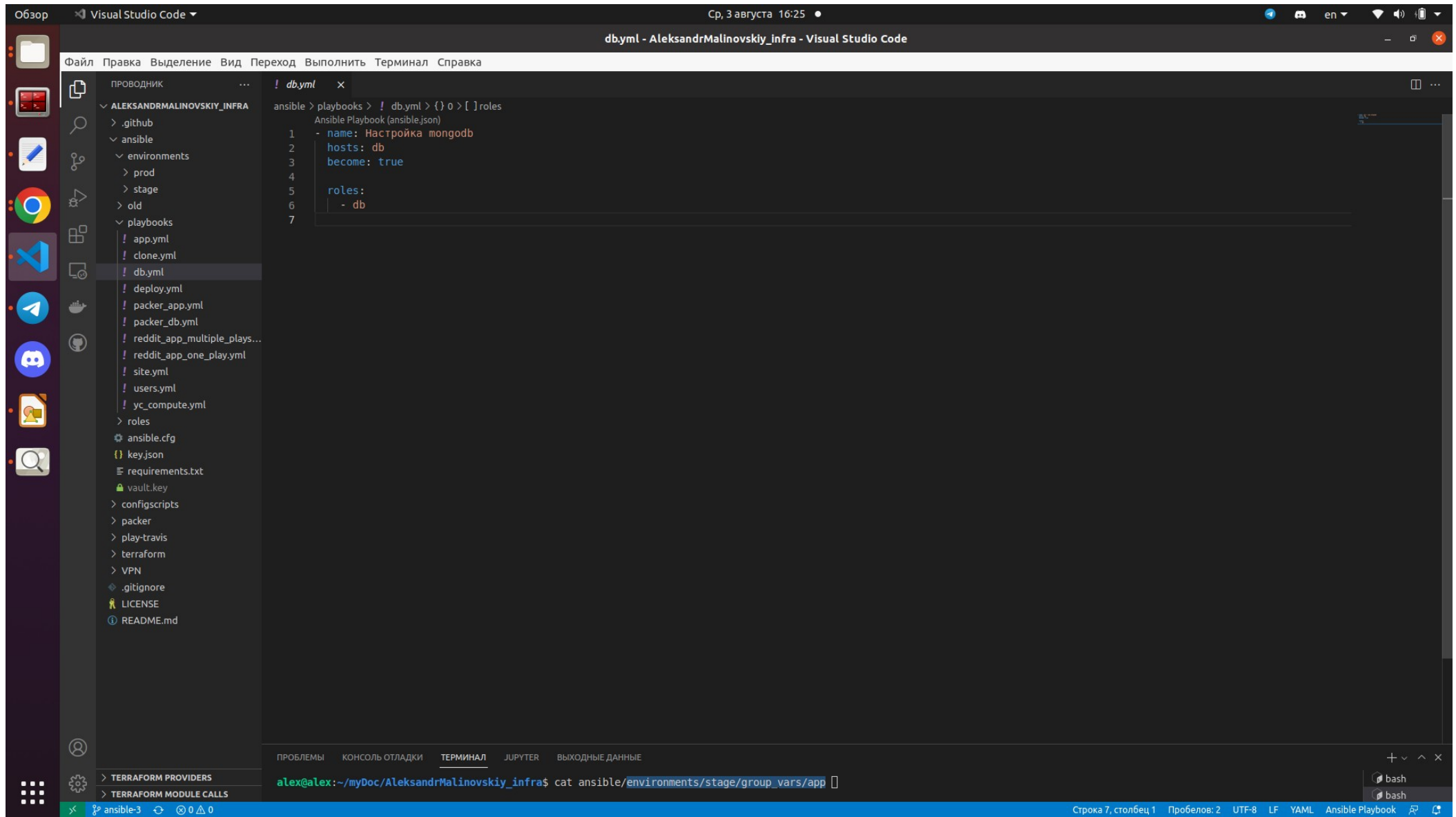
The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the directory structure. The file `main.yml` is selected under `roles/db/defaults`. The main editor shows the content of `main.yml`, which defines default variables for MongoDB. The terminal at the bottom shows a command to view the group\_vars/app file.

```
ansible > roles > db > defaults > ! main.yml > env
Ansible Vars File (ansible-vars.json)
1 ---
2 # defaults file for db
3 mongo_port: 27017
4 mongo_bind_ip: 127.0.0.1
5 env: local
6
```

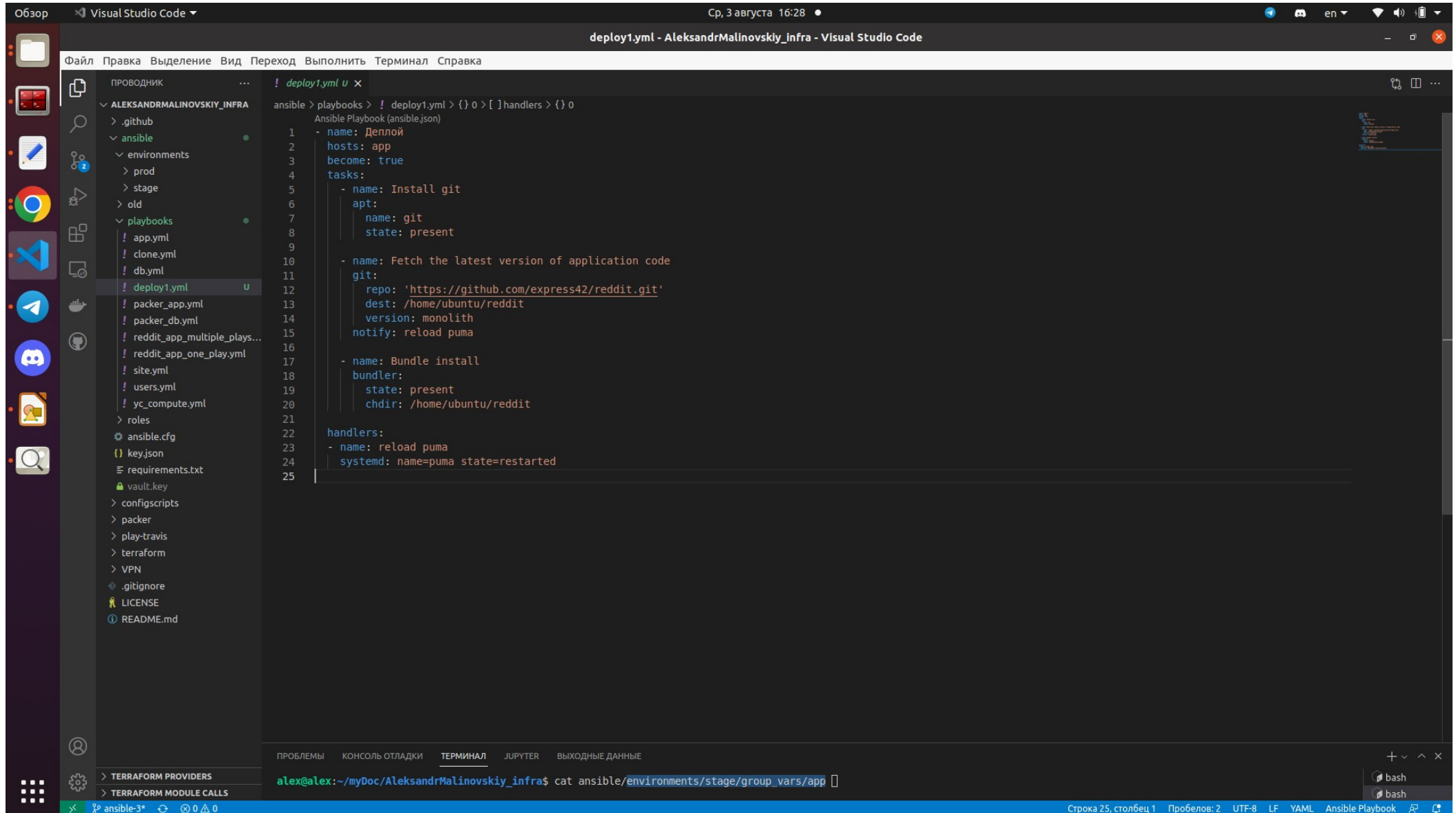
```
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
```

Строка 5, столбец 11 Пробелов: 2 UTF-8 LF YAML Ansible Vars File

# Сам плейбук db с вызовом роли db.



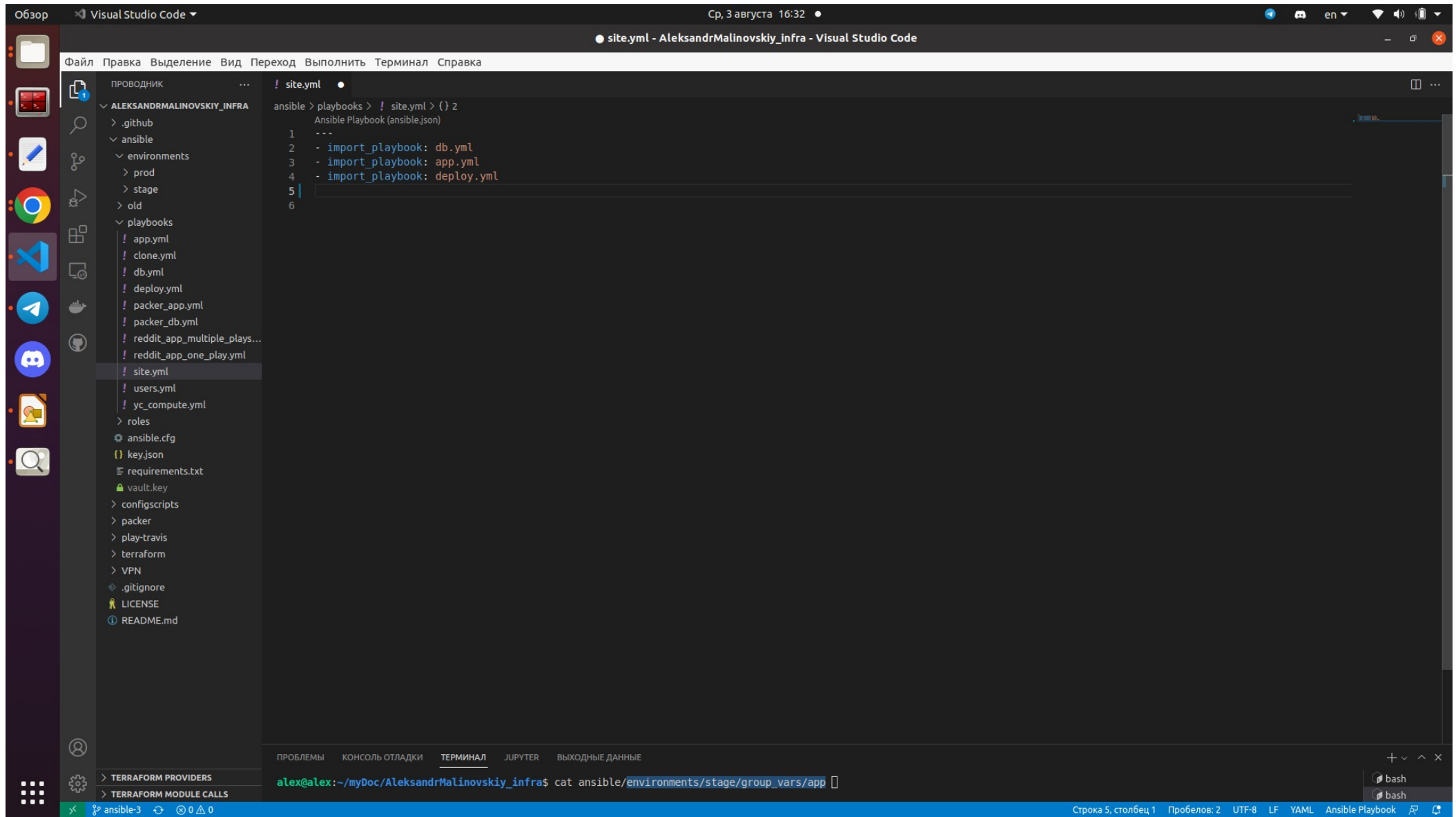
Плэйбук deploy остается без изменений Он устанавливает git, скачивает код приложения с гита и собирает его на сервере app. После перезапускает puma.service.



```
ansible-playbook deploy1.yml
ansible > playbooks > ! deploy1.yml > {} 0 > [] handlers > {} 0
Ansible Playbook (ansible.json)
1  - name: Деплой
2  hosts: app
3  become: true
4  tasks:
5    - name: Install git
6      apt:
7        name: git
8        state: present
9
10   - name: Fetch the latest version of application code
11     git:
12       repo: 'https://github.com/express42/reddit.git'
13       dest: /home/ubuntu/reddit
14       version: monolith
15     notify: reload puma
16
17   - name: Bundle install
18     bundler:
19       state: present
20       chdir: /home/ubuntu/reddit
21
22 handlers:
23   - name: reload puma
24     systemd: name=puma state=restarted
25
```

alex@alex:~/myDoc/AleksandrMalinovskiy\_infra\$ cat ansible/environments/stage/group\_vars/app

# Плэйбук site который запускает поочередно все плэйбуки.



The screenshot shows the Visual Studio Code interface with the file explorer on the left, the editor in the center, and a terminal at the bottom. The file explorer shows a project structure with a folder named 'ALEKSANDRMALINOVSKIY\_INFRA' containing subfolders like 'ansible', 'environments', 'playbooks', and 'roles'. The 'playbooks' folder is expanded, showing files like 'app.yml', 'clone.yml', 'db.yml', 'deploy.yml', 'packer\_app.yml', 'packer\_db.yml', 'reddit\_app\_multiple\_plays...', 'reddit\_app\_one\_play.yml', 'site.yml', 'users.yml', and 'yc\_compute.yml'. The 'site.yml' file is selected and its content is displayed in the editor. The terminal shows the command 'cat ansible/environments/stage/group\_vars/app' being executed.

```
ansible > playbooks > ! site.yml > {} 2
Ansible Playbook (ansible.json)
1 ---
2 - import_playbook: db.yml
3 - import_playbook: app.yml
4 - import_playbook: deploy.yml
5
6
```

```
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
```

Строка 5, столбец 1 Пробелов: 2 UTF-8 LF YAML Ansible Playbook

# Перенесем динамический инвентори файл в директорию environments/stage/

Обзор Visual Studio Code Cp, 3 августа 16:34 yc\_compute.yml - AleksandrMalinovskiy\_infra - Visual Studio Code

Файл Правка Выделение Вид Переход Выполнить Терминал Справка

ПРОВОДНИК ... ! yc\_compute.yml x

- ALEKSANDRMALINOVSKIY\_INFRA
  - > .github
  - > ansible
    - > environments
      - > prod
      - > stage
        - > group\_vars
          - ! credentials.yml
          - ! inventory.yml
          - ! requirements.yml
          - ! yc\_compute.yml
      - > old
      - > playbooks
      - > roles
        - ansible.cfg
        - key.json
        - requirements.txt
        - vault.key
      - > configscripts
      - > packer
      - > play-travis
      - > terraform
      - > VPN
      - .gitignore
      - LICENSE
      - README.md

```
ansible > environments > stage > ! yc_compute.yml > [ ] keyed_groups > { } 0 > separator
1 plugin: community.general.yc_compute
2
3 folders:
4 | - blgv4l14uqeaftgpcpr
5
6 auth_kind: serviceaccountfile
7
8 service_account_file: ./key.json
9
10 hostnames:
11 # - fqdn
12 | - "{{name}}"
13
14 compose:
15 | ansible_host: network_interfaces[0].primary_v4_address.one_to_one_nat.address
16
17
18 keyed_groups:
19 | - key: labels['tags']
20 |   prefix: ''
21 |   separator: ''
22
```

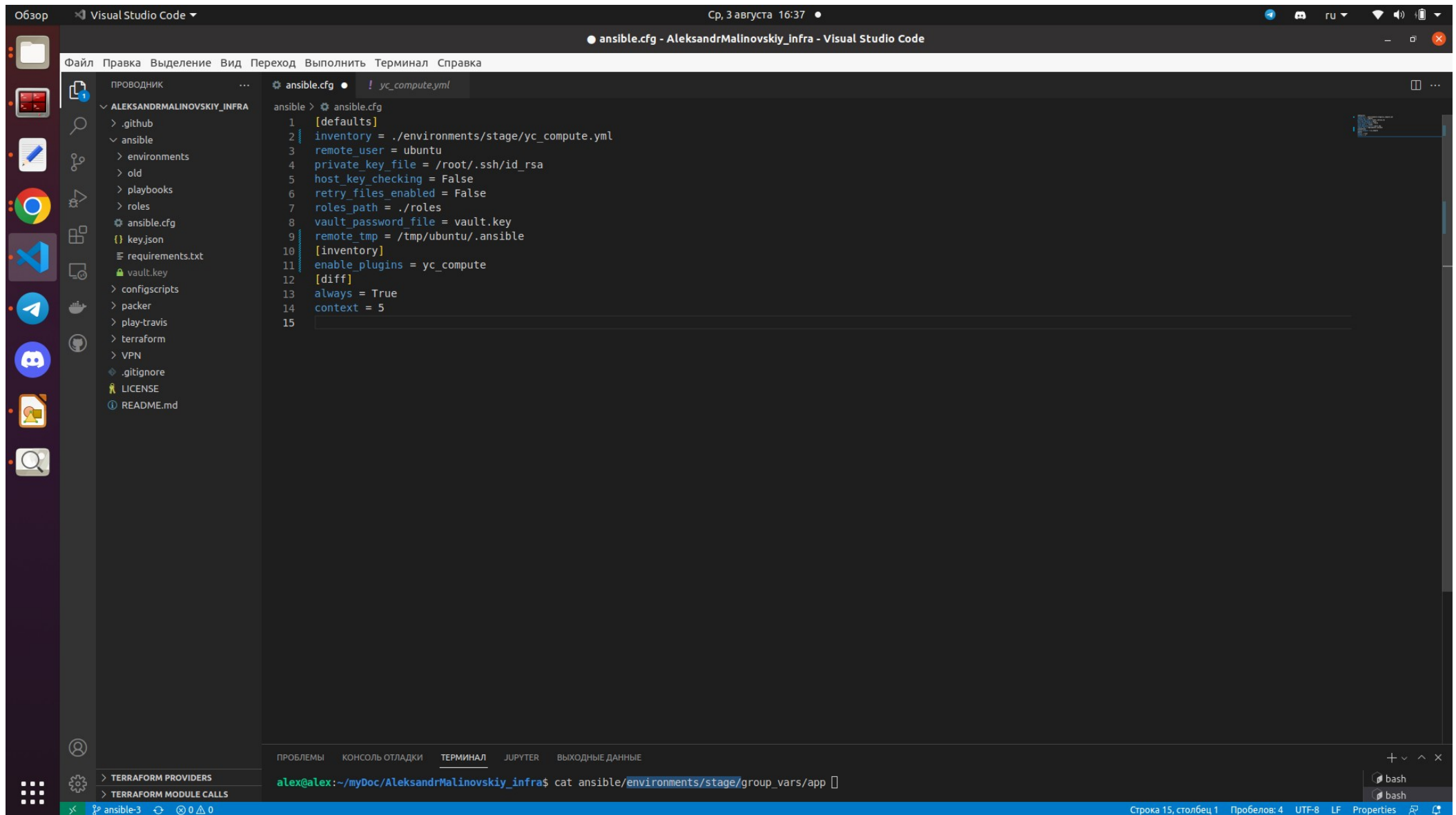
ПРОБЛЕМЫ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ JUPYTER ВЫХОДНЫЕ ДАННЫЕ

alex@alex:~/myDoc/AleksandrMalinovskiy\_infra\$ cat ansible/environments/stage/group\_vars/app

Строка 21, столбец 17 Пробелов: 2 UTF-8 LF YAML No JSON Schema



В ansible.cfg пропишем по умолчанию использовать динамический инвентори окружения stage.



The screenshot shows the Visual Studio Code interface with the file `ansible.cfg` open in the editor. The file content is as follows:

```
1 [defaults]
2 inventory = ./environments/stage/yc_compute.yml
3 remote_user = ubuntu
4 private_key_file = /root/.ssh/id_rsa
5 host_key_checking = False
6 retry_files_enabled = False
7 roles_path = ./roles
8 vault_password_file = vault.key
9 remote_tmp = /tmp/ubuntu/.ansible
10 [inventory]
11 enable_plugins = yc_compute
12 [diff]
13 always = True
14 context = 5
15
```

The left sidebar shows the file explorer with the following structure:

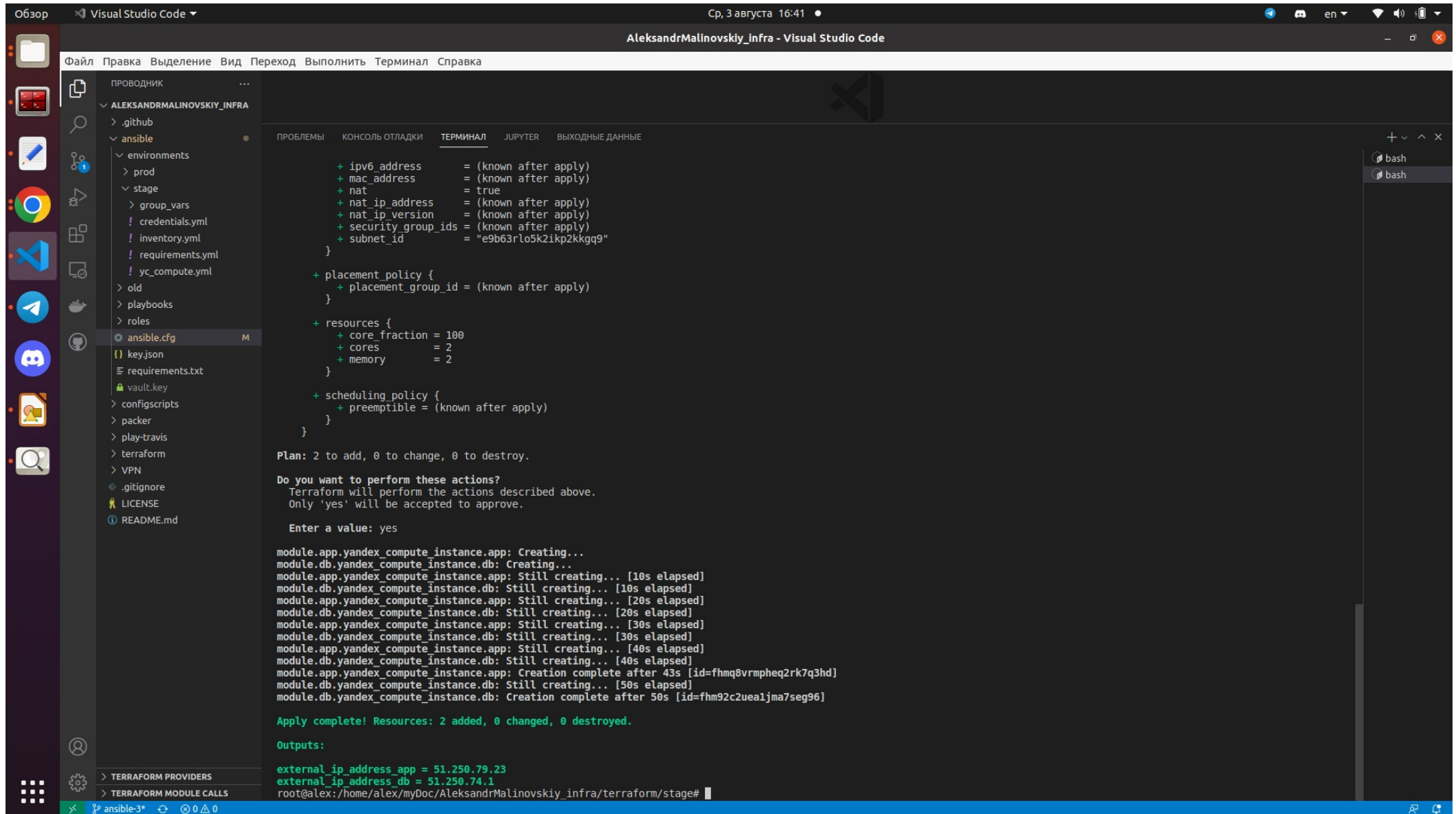
- PROBODNIK
- ALEKSANDRMALINOVSKIY\_INFRA
  - .github
  - ansible
    - environments
    - old
    - playbooks
    - roles
  - ansible.cfg
  - key.json
  - requirements.txt
  - vault.key
  - configscripts
  - packer
  - play-travis
  - terraform
  - VPN
  - .gitignore
  - LICENSE
  - README.md

The bottom terminal shows the command:

```
alex@alex:~/myDoc/AleksandrMalinovskiy_infra$ cat ansible/environments/stage/group_vars/app
```

The status bar at the bottom indicates the current line and column: "Строка 15, столбец 1".

Для проверки работоспособности плэйбуков создадим заново VM на yandexcloud. Осталось только указать новый ip переменной в файле app в директории environments/stage/group\_vars/



The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying the project structure. The main editor shows the `ansible.cfg` file, and the terminal on the right displays the output of a Terraform command.

**File Explorer:**

- PROБОННИК
- ▼ ALEKSANDRMALINOVSKIY\_INFRA
  - ▼ .github
  - ▼ ansible
    - ▼ environments
      - ▼ prod
      - ▼ stage
        - ▼ group\_vars
          - ! credentials.yml
          - ! inventory.yml
          - ! requirements.yml
          - ! yc\_compute.yml
      - old
      - playbooks
      - roles
    - ansible.cfg
    - key.json
    - requirements.txt
    - vault.key
    - configs
    - packer
    - play-travis
    - terraform
    - VPN
    - .gitignore
    - LICENSE
    - README.md

**Terminal Output:**

```
+ ipv6_address = (known after apply)
+ mac_address = (known after apply)
+ nat = true
+ nat_ip_address = (known after apply)
+ nat_ip_version = (known after apply)
+ security_group_ids = (known after apply)
+ subnet_id = "e9b63rlo5k2ikp2kkgq9"
}

+ placement_policy {
+ placement_group_id = (known after apply)
}

+ resources {
+ core_fraction = 100
+ cores = 2
+ memory = 2
}

+ scheduling_policy {
+ preemptible = (known after apply)
}
}

Plan: 2 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

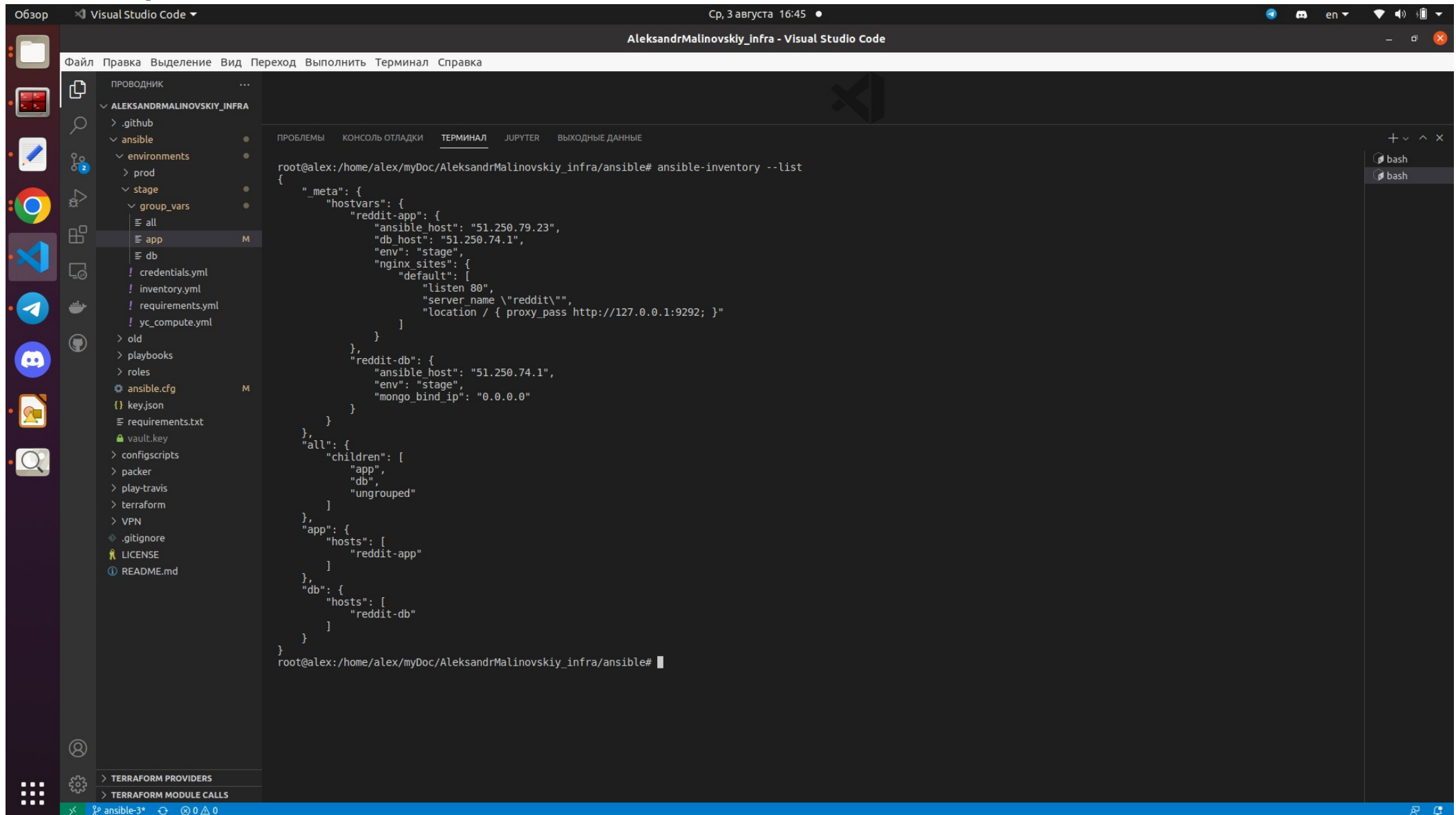
module.app.yandex_compute_instance.app: Creating...
module.db.yandex_compute_instance.db: Creating...
module.app.yandex_compute_instance.app: Still creating... [10s elapsed]
module.db.yandex_compute_instance.db: Still creating... [10s elapsed]
module.app.yandex_compute_instance.app: Still creating... [20s elapsed]
module.db.yandex_compute_instance.db: Still creating... [20s elapsed]
module.app.yandex_compute_instance.app: Still creating... [30s elapsed]
module.db.yandex_compute_instance.db: Still creating... [30s elapsed]
module.app.yandex_compute_instance.app: Still creating... [40s elapsed]
module.db.yandex_compute_instance.db: Still creating... [40s elapsed]
module.app.yandex_compute_instance.app: Creation complete after 43s [id=fhm92c2uea1jma7seg96]
module.db.yandex_compute_instance.db: Still creating... [50s elapsed]
module.db.yandex_compute_instance.db: Creation complete after 50s [id=fhm92c2uea1jma7seg96]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:
external_ip_address_app = 51.250.79.23
external_ip_address_db = 51.250.74.1
root@alex:/home/alex/myDoc/AleksandrMalinovskiy_infra/terraform/stage#
```



Вывод команды `ansible-inventory --list`.  
Вывод показывает stage окружение, наименование хостов и их ip.



```
root@alex:/home/alex/myDoc/AleksandrMalinovskiy_infra/ansible# ansible-inventory --list
{
  "_meta": {
    "hostvars": {
      "reddit-app": {
        "ansible_host": "51.250.79.23",
        "db_host": "51.250.74.1",
        "env": "stage",
        "nginx_sites": {
          "default": {
            "listen": 80,
            "server_name": "reddit",
            "location": { proxy_pass http://127.0.0.1:9292; }
          }
        }
      },
      "reddit-db": {
        "ansible_host": "51.250.74.1",
        "env": "stage",
        "mongo_bind_ip": "0.0.0.0"
      }
    }
  },
  "all": {
    "children": [
      "app",
      "db",
      "ungrouped"
    ]
  },
  "app": {
    "hosts": [
      "reddit-app"
    ]
  },
  "db": {
    "hosts": [
      "reddit-db"
    ]
  }
}
```

# Итог выполнения плейбука site.

Обзор Visual Studio Code

АleksandrMalinovskiy\_Infra - Visual Studio Code

Файл Правка Выделение Вид Переход Выполнить Терминал Справка

ПРОБЛЕМЫ КОНСОЛЬ ОТЛАДКИ **ТЕРМИНАЛ** JUPYTER ВЫХОДНЫЕ ДАННЫЕ

PROVODNIK ...

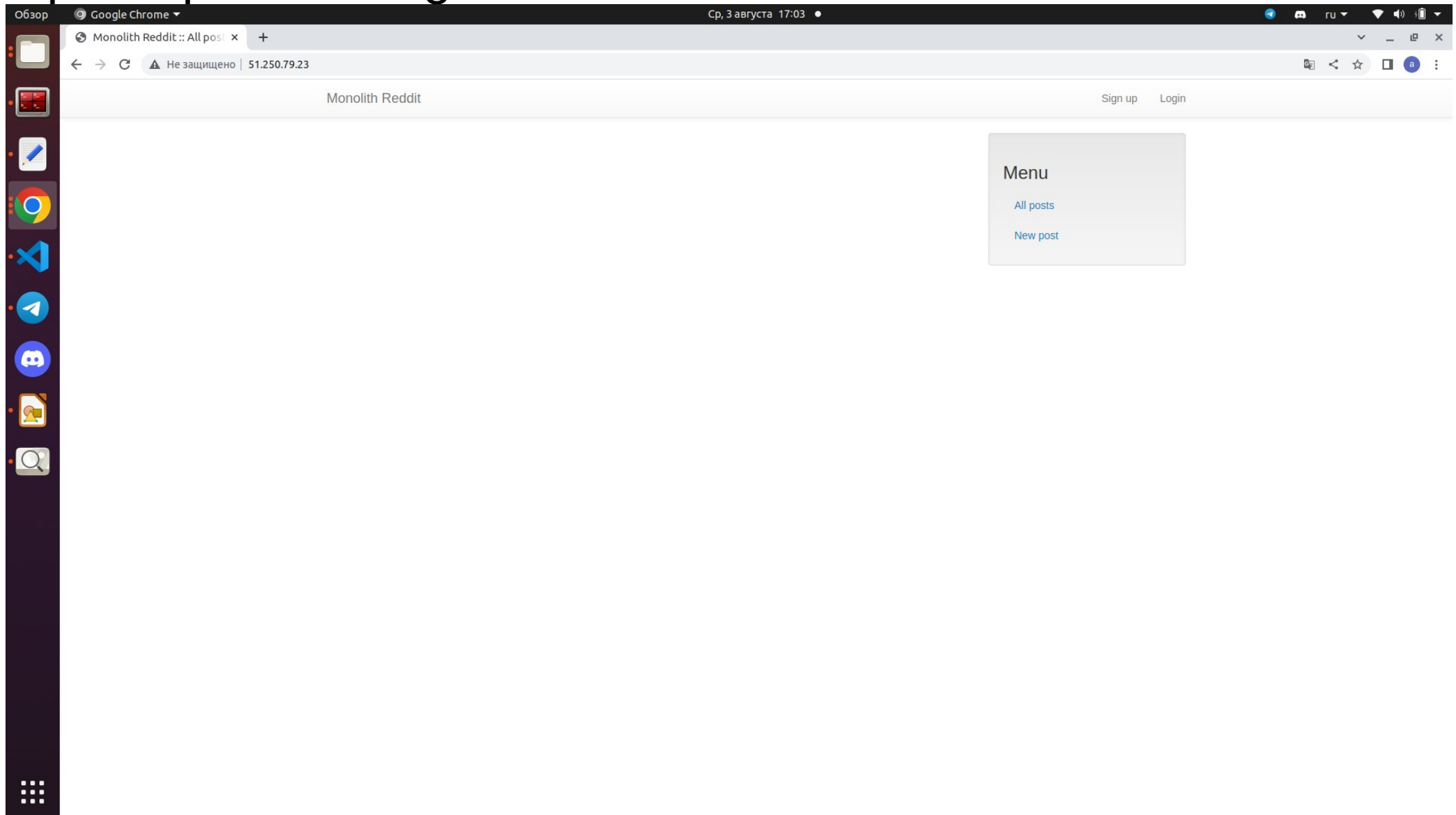
- ▼ ALEKSANDRMALINOVSKIY\_INFRA
  - > .github
  - ▼ ansible
    - > environments
    - > old
    - ▼ playbooks
      - ! app.yml
      - ! clone.yml
      - ! db.yml
      - ! deploy.yml
      - ! packer\_app.yml
      - ! packer\_db.yml
      - ! reddit\_app\_multiple\_plays...
      - ! reddit\_app\_one\_play.yml
      - ! site.yml M
      - ! users.yml
      - ! yc\_compute.yml
    - > roles
    - ! ansible.cfg M
    - ! key.json
    - ! requirements.txt
    - ! vault.key
    - > configscripts
    - > packer
    - > play-travis
    - > terraform
    - > VPN
    - > .gitignore
    - ! LICENSE
    - ! README.md
  - > TERRAFORM PROVIDERS
  - > TERRAFORM MODULE CALLS

```
TASK [jdauphant.nginx : Create the configurations for independent config file for streams] *****
TASK [jdauphant.nginx : Create links for modules-enabled] *****
TASK [jdauphant.nginx : include_tasks] *****
skipping: [reddit-app]
TASK [jdauphant.nginx : include_tasks] *****
skipping: [reddit-app]
TASK [jdauphant.nginx : Start the nginx service] *****
changed: [reddit-app]
RUNNING HANDLER [app : reload puma] *****
changed: [reddit-app]
RUNNING HANDLER [jdauphant.nginx : restart nginx] *****
changed: [reddit-app] => {
  "msg": "checking config first"
}
RUNNING HANDLER [jdauphant.nginx : reload nginx] *****
changed: [reddit-app] => {
  "msg": "checking config first"
}
RUNNING HANDLER [jdauphant.nginx : check nginx configuration] *****
ok: [reddit-app]
RUNNING HANDLER [jdauphant.nginx : restart nginx - after config check] *****
changed: [reddit-app]
RUNNING HANDLER [jdauphant.nginx : reload nginx - after config check] *****
changed: [reddit-app]
PLAY [Деплой] *****
TASK [Gathering Facts] *****
ok: [reddit-app]
TASK [Install git] *****
ok: [reddit-app]
TASK [Fetch the latest version of application code] *****
>> Newly checked out 5c217c565c1122c5343dc0514c116ae816c17ca2
changed: [reddit-app]
TASK [Bundle install] *****
changed: [reddit-app]
RUNNING HANDLER [reload puma] *****
changed: [reddit-app]
PLAY RECAP *****
reddit-app      : ok=28  changed=18  unreachable=0  failed=0  skipped=17  rescued=0  ignored=0
reddit-db      : ok=4    changed=2   unreachable=0  failed=0  skipped=0   rescued=0  ignored=0
root@alex:/home/alex/myDoc/AleksandrMalinovskiy_infra/ansible#
```

bash

ansible-3\* 0 0

В итоге на сервере app с ip 51.250.79.23 крутится приложение на 80 порту, а не на 9292 как ранее. За счет проксирования nginx.



Для проверки смотрим что на сервере app работает nginx.

A terminal window titled 'ubuntu@fhmq8vrmpheq2rk7q3hd: ~' with a top bar showing 'Ср, 3 августа 16:59'. The terminal shows a user 'alex' logging in as 'root' and running 'systemctl status nginx'. The output shows Nginx is active and running. Then, the user runs 'systemctl restart nginx' and the output shows Nginx is restarted successfully. The terminal also shows the user running 'systemctl status nginx' again, confirming it is active and running. The terminal window has a dark theme and a sidebar with application icons on the left.