

# Pgbouncer/patroni/postgres

Дано:

- 3 сервера с ОС Debian

Задание:

- Написать плейбуки для ansible на установку кластера pgbouncer/patroni/postgres

## План выполнения:

- написать роль etcd
- написать роль haproxy
- написать роль postgres
- написать роль patroni
- написать роль patroni 2
- написать роль pgbouncer
- написать плейбук разворачивающий кластер используя ранее написанные роли.
- развернуть 3 сервера
- создать инвентори файл и указать там все переменные которые используются в ролях.
- запустить плейбук и проверить работу кластера

Кластер будет состоять из:

Первый сервер.

Etcd — это распределенное надежное хранилище ключей и значений для наиболее важных данных распределенной системы. Он используется Patroni для хранения информации о состоянии кластера и параметрах конфигурации PostgreSQL.

Нароку это бесплатное, очень быстрое и надежное решение, предлагающее высокую доступность, балансировку нагрузки и прокси для приложений на основе TCP и HTTP.

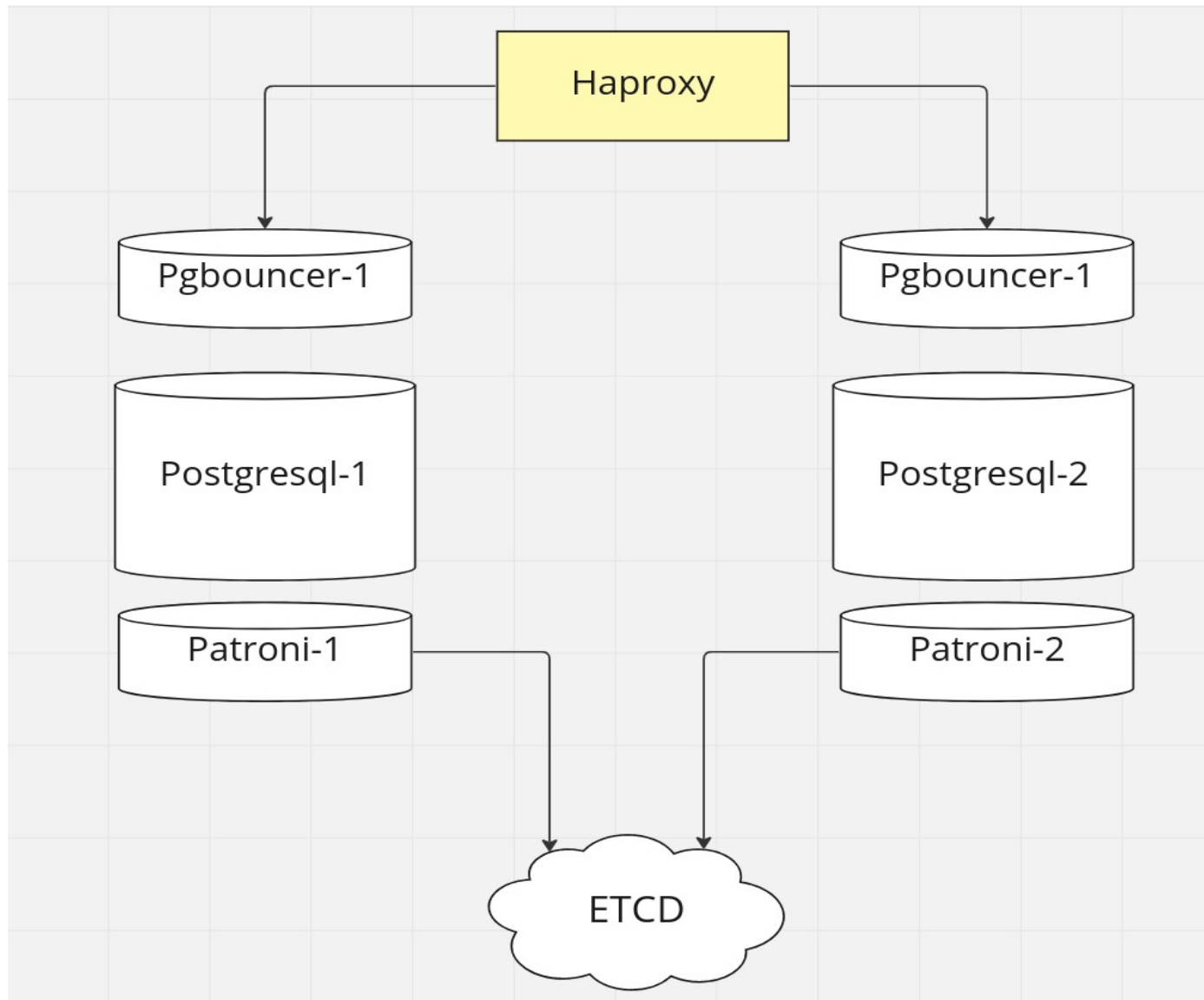
Второй и третий сервер.

PostgreSQL — свободная объектно-реляционная система управления базами данных.

Patroni — это шаблон для создания собственного настроенного высокодоступного решения с использованием Python и — для максимальной доступности — распределенного хранилища конфигураций, такого как ZooKeeper, etcd, Consul или Kubernetes. Используется для автоматизации управления экземплярами PostgreSQL и автоматического аварийного переключения.

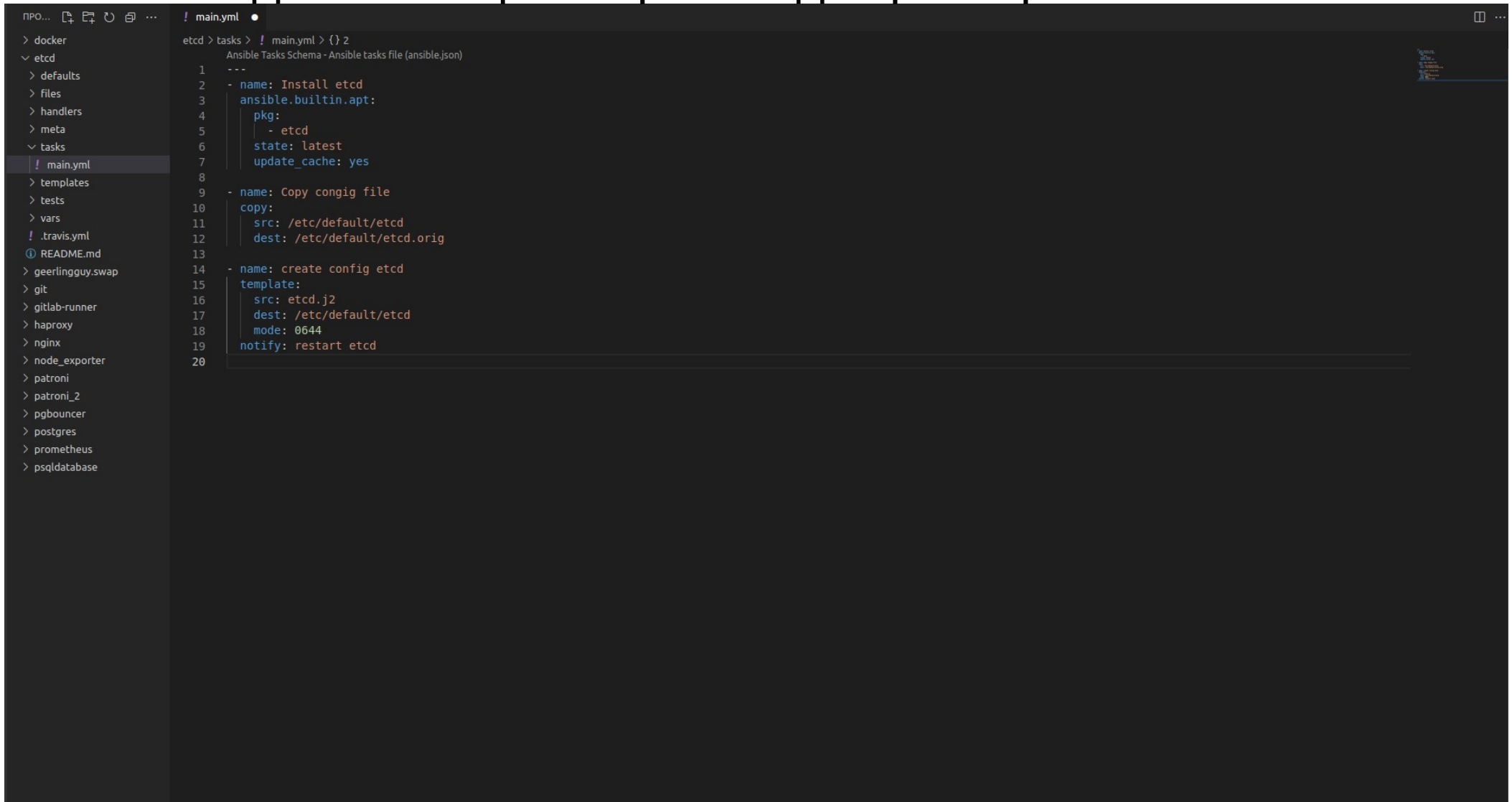
PgBouncer — это программа, управляющая пулом соединений Postgres. Любое конечное приложение может подключиться к pgbouncer, как если бы это был непосредственно сервер Postgres, и pgbouncer создаст подключение к реальному серверу, либо задействует одно из ранее установленных подключений.

# Принципиальная схема кластера.



# Инициализировал роль etcd

В директории task в файле main.yml прописал установку etcd и создание конфига из шаблона j2 с сохранением оригинала. После создания конфига происходит рестарт etcd.

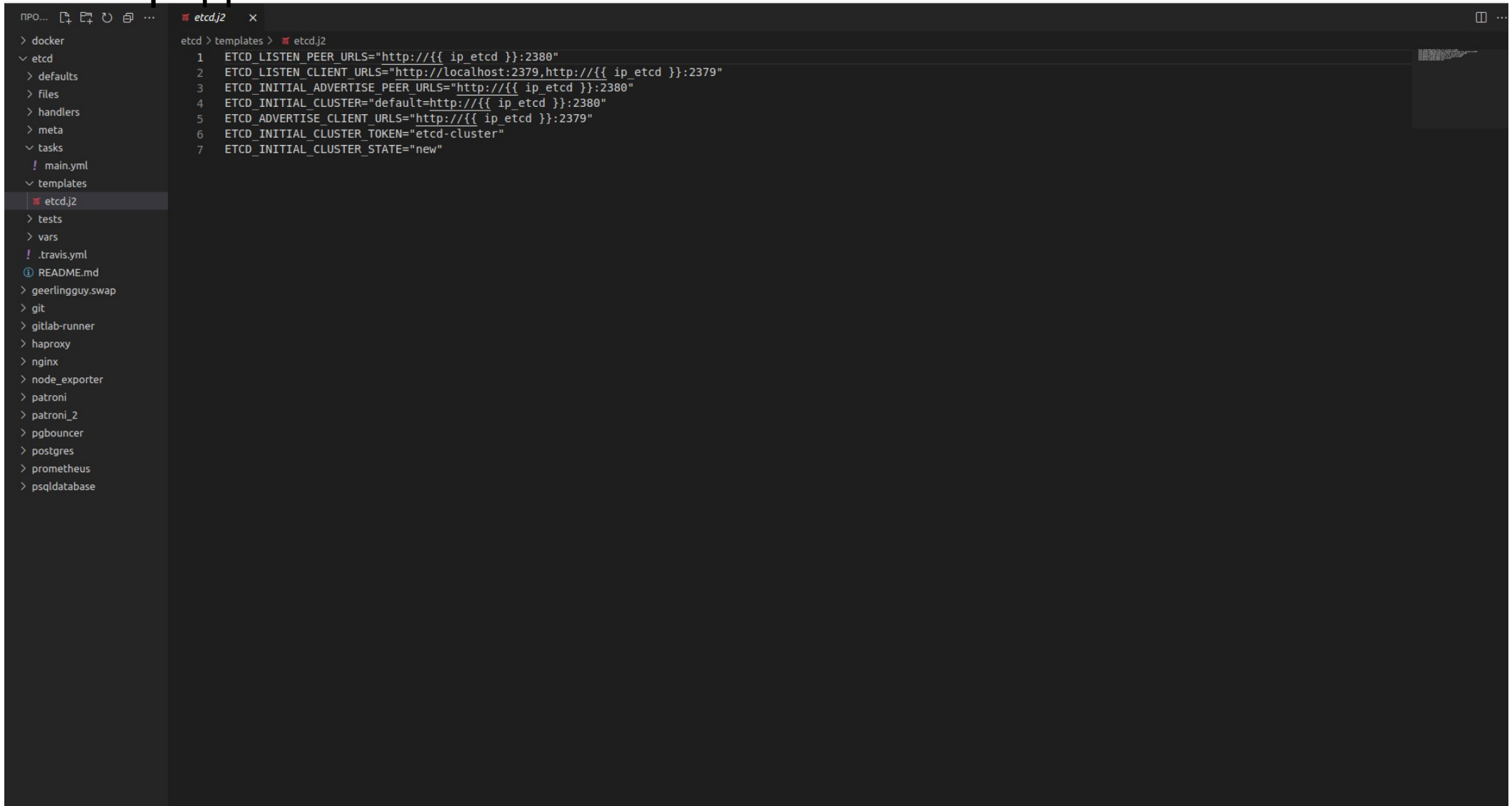


```
etcd > tasks > ! main.yml > {} 2
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1  ---
2  - name: Install etcd
3    ansible.builtin.apt:
4      pkg:
5        - etcd
6      state: latest
7      update_cache: yes
8
9  - name: Copy congig file
10    copy:
11      src: /etc/default/etcd
12      dest: /etc/default/etcd.orig
13
14  - name: create config etcd
15    template:
16      src: etcd.j2
17      dest: /etc/default/etcd
18      mode: 0644
19    notify: restart etcd
20
```

В директории templates прописал шаблон для конфиг файла etcd.

Использую переменную ip\_etcd.

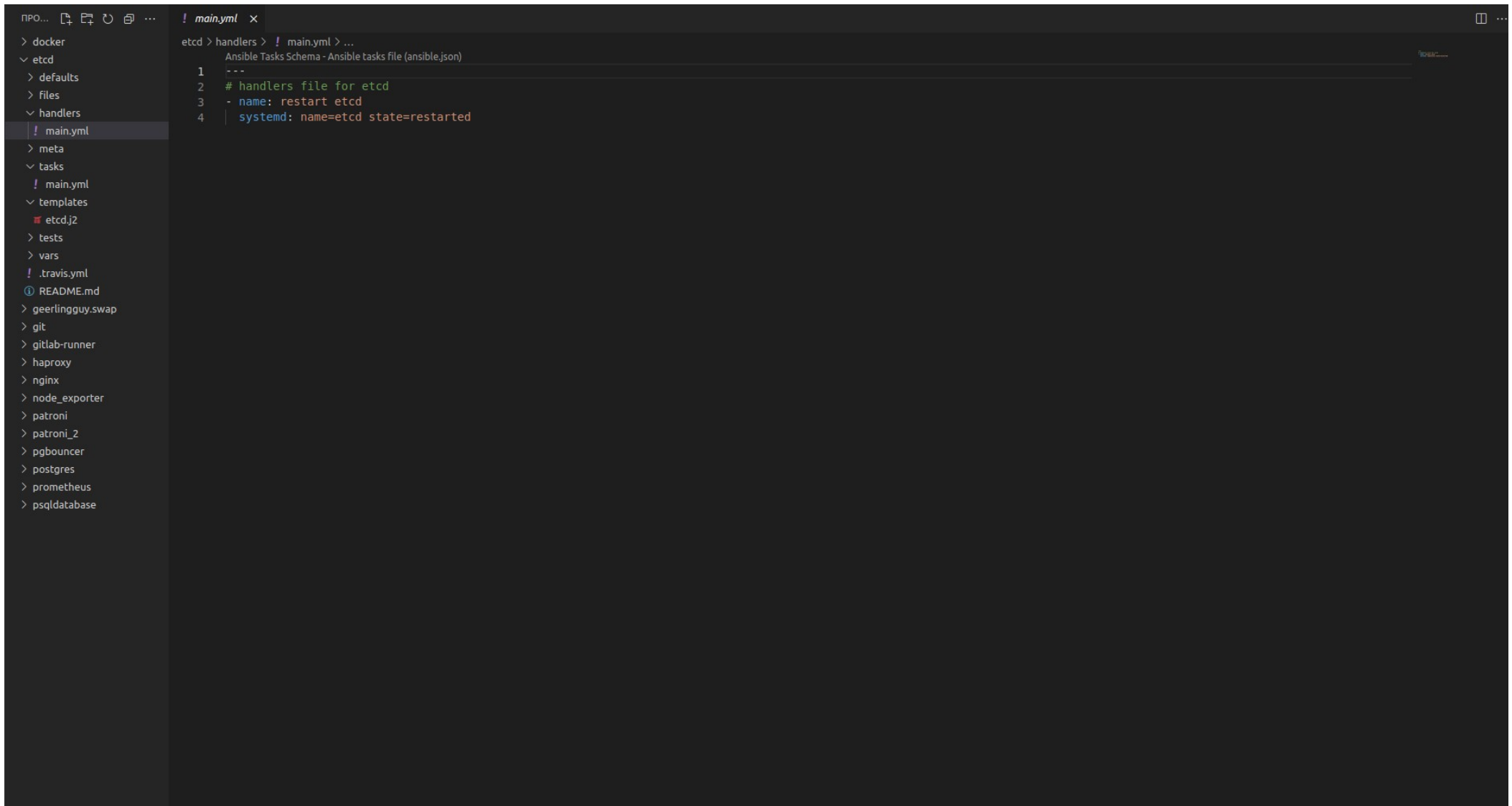
Её определил в inventori.



The screenshot shows a code editor with a dark theme. On the left is a sidebar with a file explorer showing a directory structure. The 'templates' directory is expanded, and 'etcd.j2' is selected. The main editor area displays the content of 'etcd.j2', which is a Jinja2 template for an etcd configuration file. The template uses the 'ip\_etcd' variable to define various URLs and cluster settings.

```
etcd > templates > etcd.j2
1 ETCD_LISTEN_PEER_URLS="http://{{ ip_etcd }}:2380"
2 ETCD_LISTEN_CLIENT_URLS="http://localhost:2379,http://{{ ip_etcd }}:2379"
3 ETCD_INITIAL_ADVERTISE_PEER_URLS="http://{{ ip_etcd }}:2380"
4 ETCD_INITIAL_CLUSTER="default=http://{{ ip_etcd }}:2380"
5 ETCD_ADVERTISE_CLIENT_URLS="http://{{ ip_etcd }}:2379"
6 ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
7 ETCD_INITIAL_CLUSTER_STATE="new"
```

В директории handlers в файле main.yml прописал рестарт etcd по notify: restart etcd

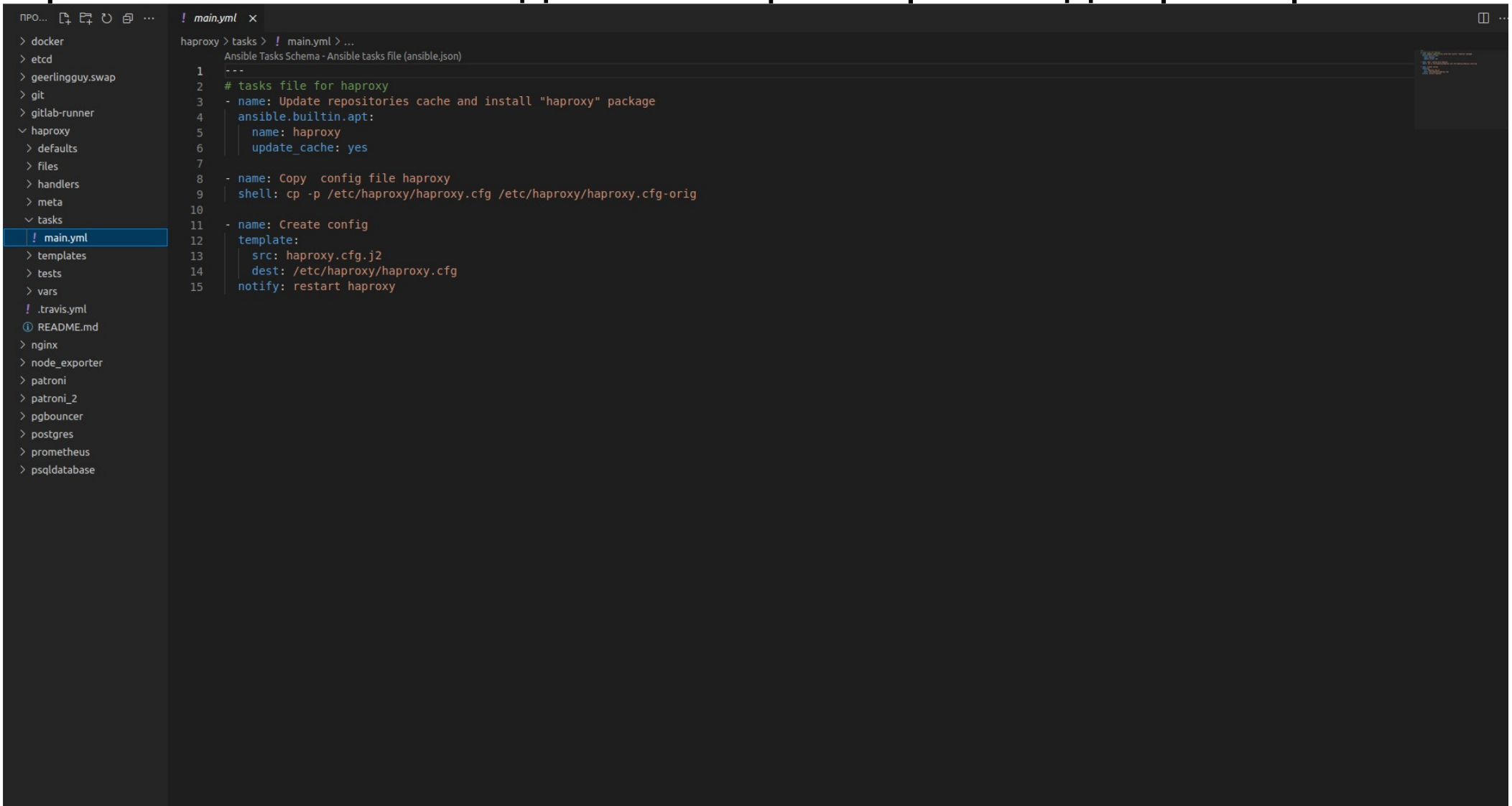


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with 'handlers' expanded, and 'main.yml' selected. The code editor shows the content of 'main.yml' with the following text:

```
etcd > handlers > ! main.yml > ...
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1 ---
2 # handlers file for etcd
3 - name: restart etcd
4   systemd: name=etcd state=restarted
```

Инициализировал роль haproxy

В директории task в файле main.yml прописал установку haproxy и создание конфига из шаблона j2 с сохранением оригинала. После создания конфига происходит рестарт.

A screenshot of a code editor with a dark theme. On the left is a file explorer showing a directory structure with folders like 'docker', 'etcd', 'haproxy', and 'tasks'. The 'tasks' folder is expanded, and 'main.yml' is selected. The main editor area shows the content of 'main.yml', which is an Ansible tasks file for haproxy. The file includes a comment, a task to update repositories and install haproxy, a task to copy the haproxy config file, and a task to create a new config from a j2 template and restart haproxy. Line numbers 1 through 15 are visible on the left of the code.

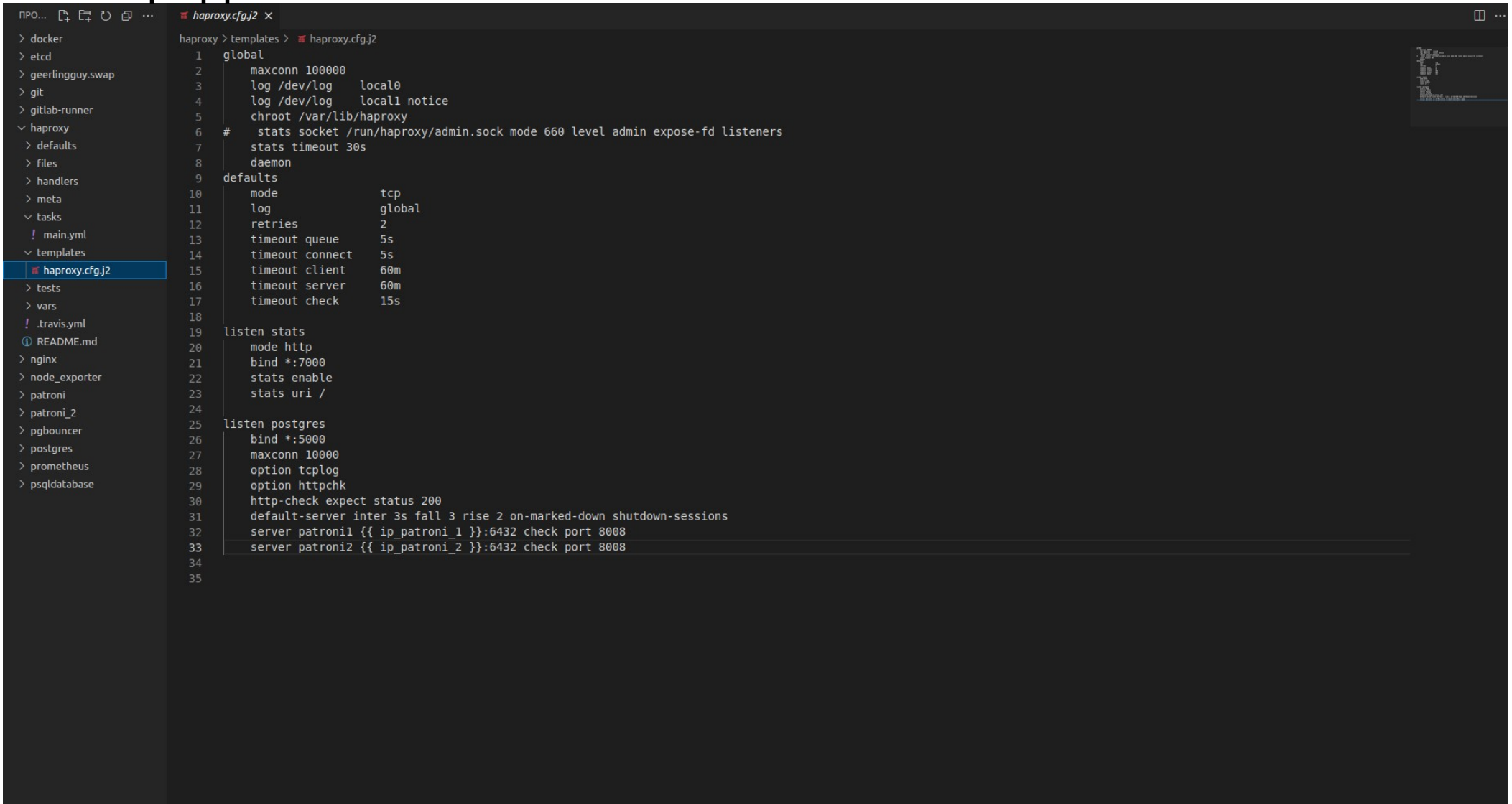
```
1  ---
2  # tasks file for haproxy
3  - name: Update repositories cache and install "haproxy" package
4    ansible.builtin.apt:
5      name: haproxy
6      update_cache: yes
7
8  - name: Copy config file haproxy
9    shell: cp -p /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg-orig
10
11 - name: Create config
12   template:
13     src: haproxy.cfg.j2
14     dest: /etc/haproxy/haproxy.cfg
15   notify: restart haproxy
```



В директории templates прописал шаблон для конфиг файла haproxy.

Использую переменную ip\_patroni 1 и 2.

Её определил в inventori.

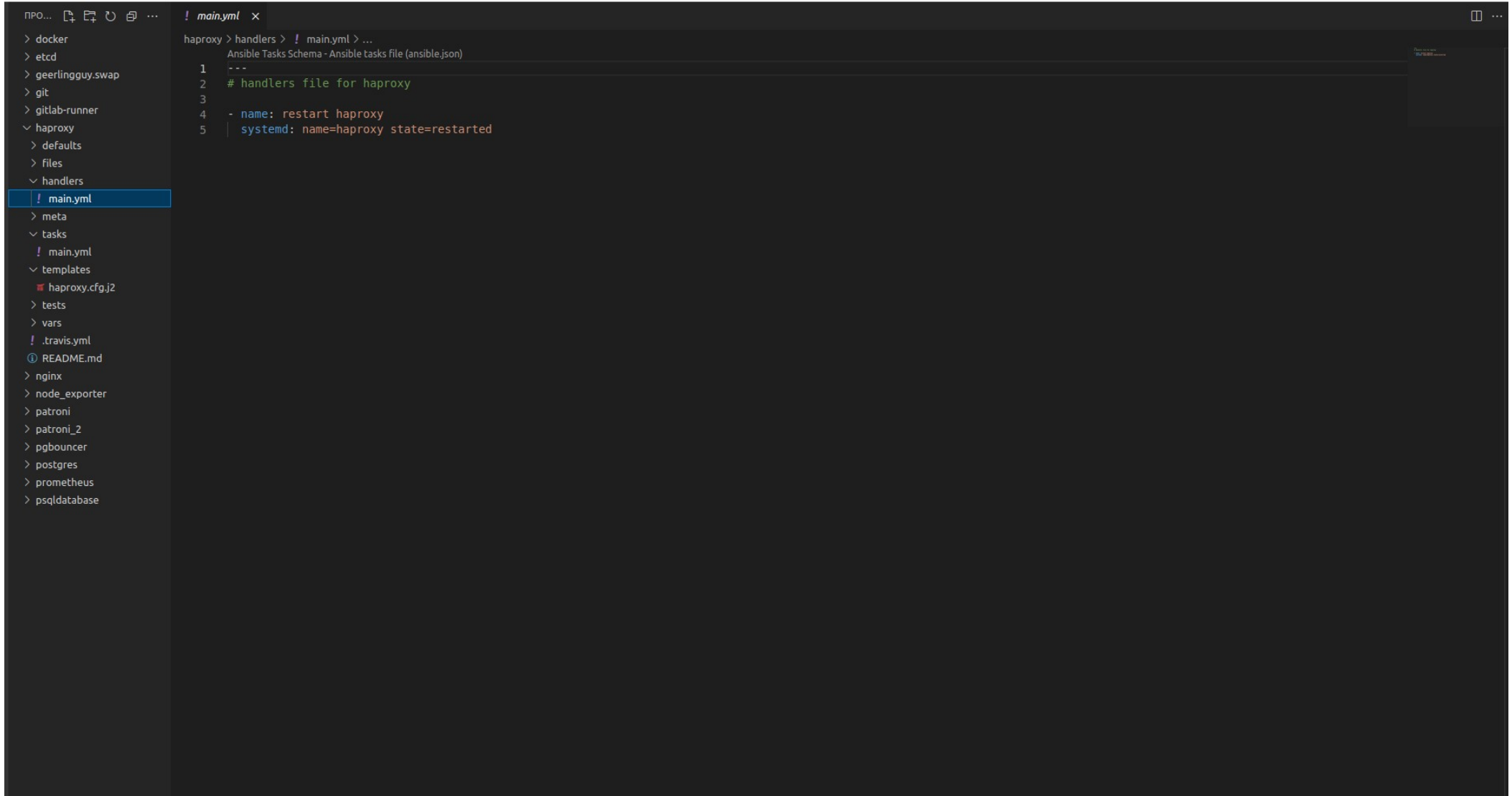


The screenshot shows a code editor with a sidebar on the left containing a file tree. The tree includes folders like 'docker', 'etcd', 'geerlingguy.swap', 'git', 'gitlab-runner', 'haproxy', 'defaults', 'files', 'handlers', 'meta', 'tasks', 'templates', 'tests', 'vars', and files like 'main.yml', '.travis.yml', 'README.md', 'nginx', 'node\_exporter', 'patroni', 'patroni\_2', 'pgbouncer', 'postgres', 'prometheus', and 'psqldatabase'. The 'haproxy.cfg.j2' file in the 'templates' folder is selected and highlighted in blue.

The main editor area displays the content of 'haproxy.cfg.j2' with line numbers 1 through 35. The configuration is a Jinja2 template for a haproxy config file, using the variable 'ip\_patroni' to define server addresses.

```
1 global
2     maxconn 100000
3     log /dev/log      local0
4     log /dev/log      local1 notice
5     chroot /var/lib/haproxy
6     # stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
7     stats timeout 30s
8     daemon
9
10    defaults
11        mode                tcp
12        log                  global
13        retries              2
14        timeout queue        5s
15        timeout connect      5s
16        timeout client        60m
17        timeout server        60m
18        timeout check        15s
19
20    listen stats
21        mode http
22        bind *:7000
23        stats enable
24        stats uri /
25
26    listen postgres
27        bind *:5000
28        maxconn 10000
29        option tcplog
30        option httpchk
31        http-check expect status 200
32        default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
33        server patroni1 {{ ip_patroni_1 }}:6432 check port 8008
34        server patroni2 {{ ip_patroni_2 }}:6432 check port 8008
35
```

В директории handlers в файле main.yml прописал рестарт etcd по notify: restart haproxy

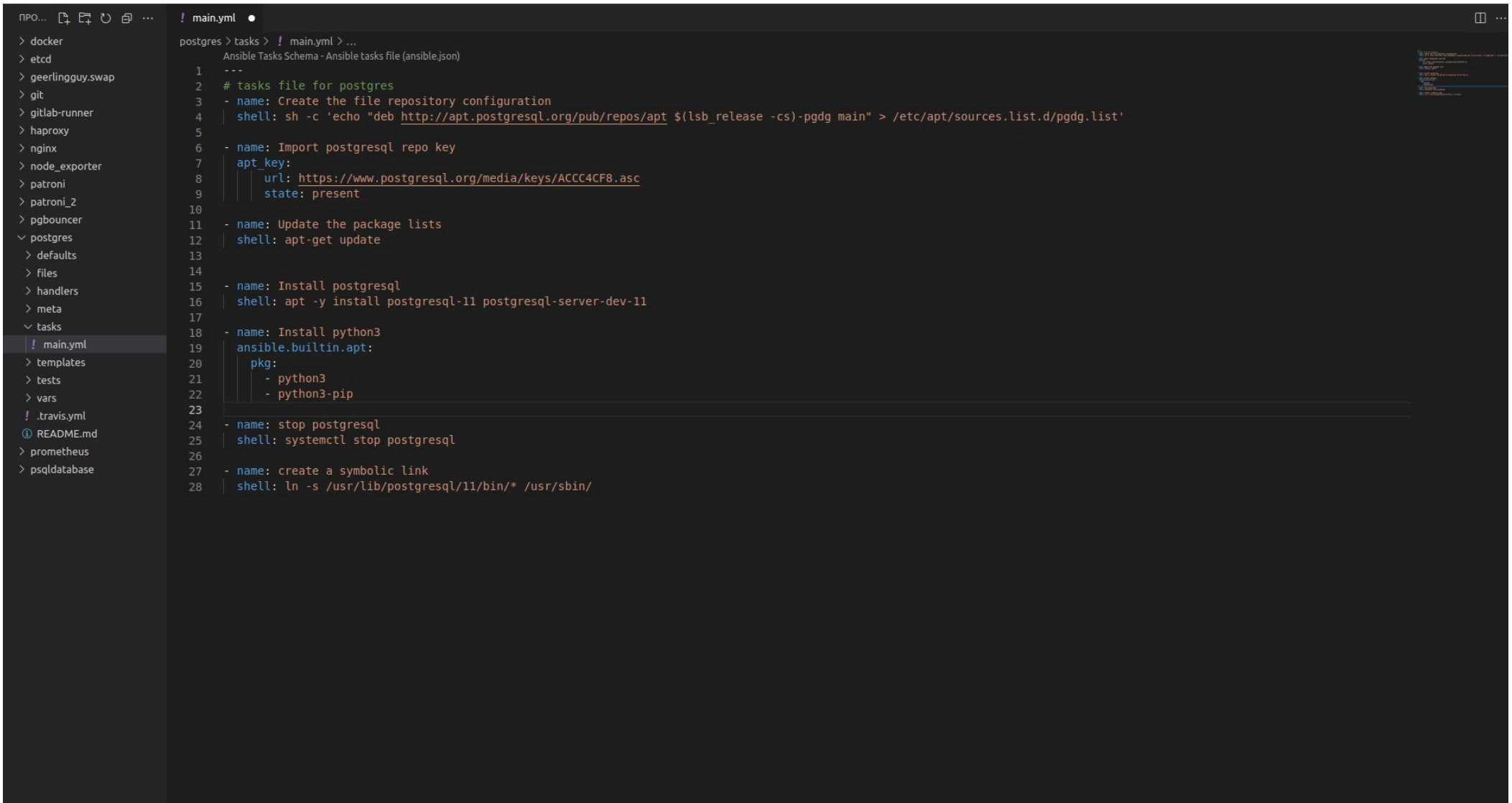


```
1 ---
2 # handlers file for haproxy
3
4 - name: restart haproxy
5   systemd: name=haproxy state=restarted
```

The screenshot shows a code editor with a sidebar on the left displaying a file tree. The 'handlers' directory is expanded, and 'main.yml' is selected. The main editor area shows the content of 'main.yml', which is an Ansible handlers file for haproxy. It contains a single task named 'restart haproxy' that uses the 'systemd' module to restart the haproxy service.

# Инициализировал роль postgres

В директории task в файле main.yml прописал установку postgres и создание symbolic link и остановка его.

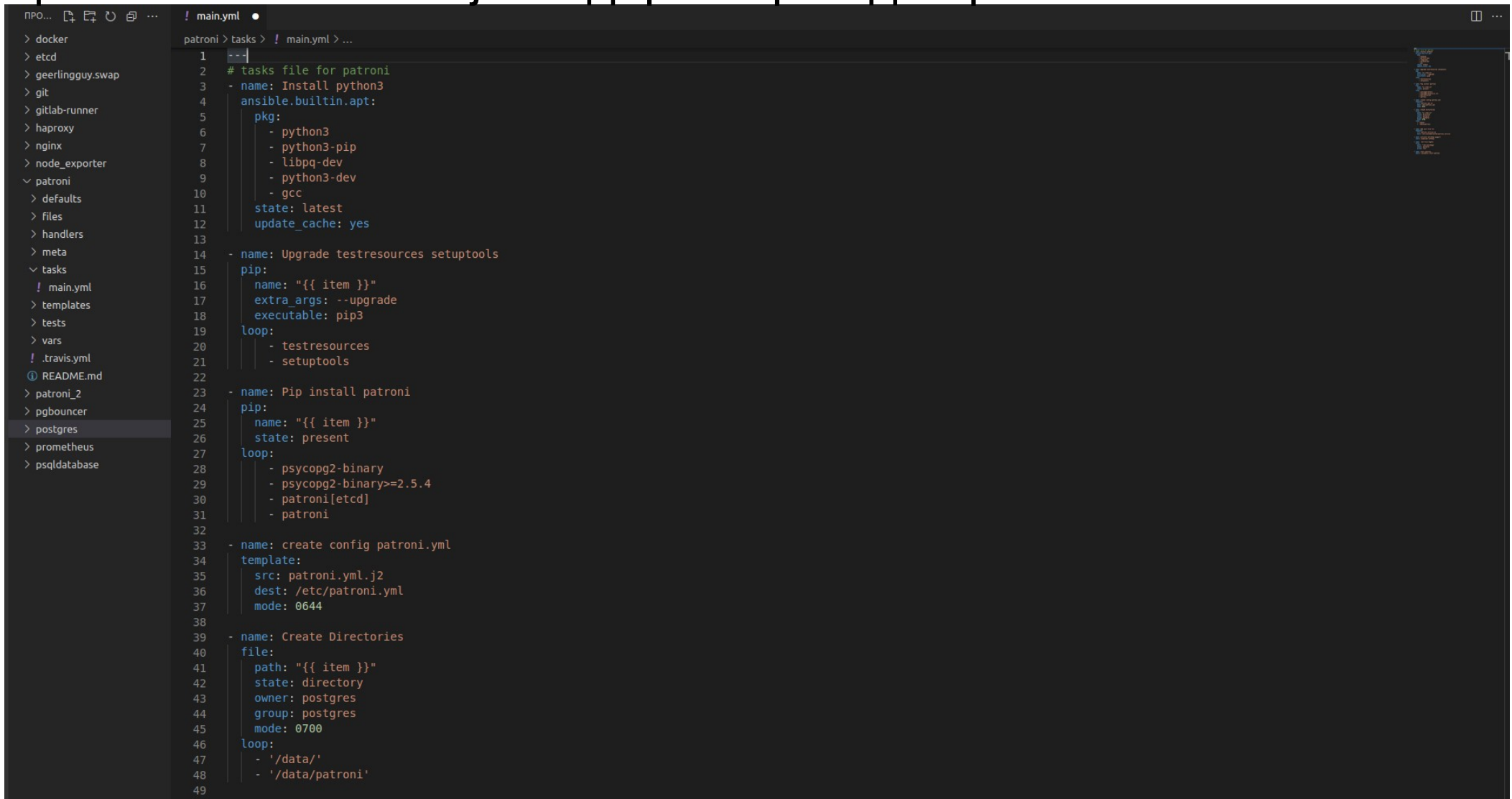


The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with various files and folders, including 'postgres' and 'tasks'. The 'tasks' folder is expanded, showing 'main.yml'. The code editor displays the content of 'main.yml', which is an Ansible tasks file for PostgreSQL installation. The tasks include creating a file repository configuration, importing the PostgreSQL repo key, updating package lists, installing PostgreSQL and Python3, stopping PostgreSQL, and creating a symbolic link.

```
postgres > tasks > ! main.yml > ...
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1  ---
2  # tasks file for postgres
3  - name: Create the file repository configuration
4    shell: sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
5
6  - name: Import postgresql repo key
7    apt_key:
8      url: https://www.postgresql.org/media/keys/ACCC4CF8.asc
9      state: present
10
11 - name: Update the package lists
12   shell: apt-get update
13
14
15 - name: Install postgresql
16   shell: apt -y install postgresql-11 postgresql-server-dev-11
17
18 - name: Install python3
19   ansible.builtin.apt:
20     pkg:
21       - python3
22       - python3-pip
23
24 - name: stop postgresql
25   shell: systemctl stop postgresql
26
27 - name: create a symbolic link
28   shell: ln -s /usr/lib/postgresql/11/bin/* /usr/sbin/
```

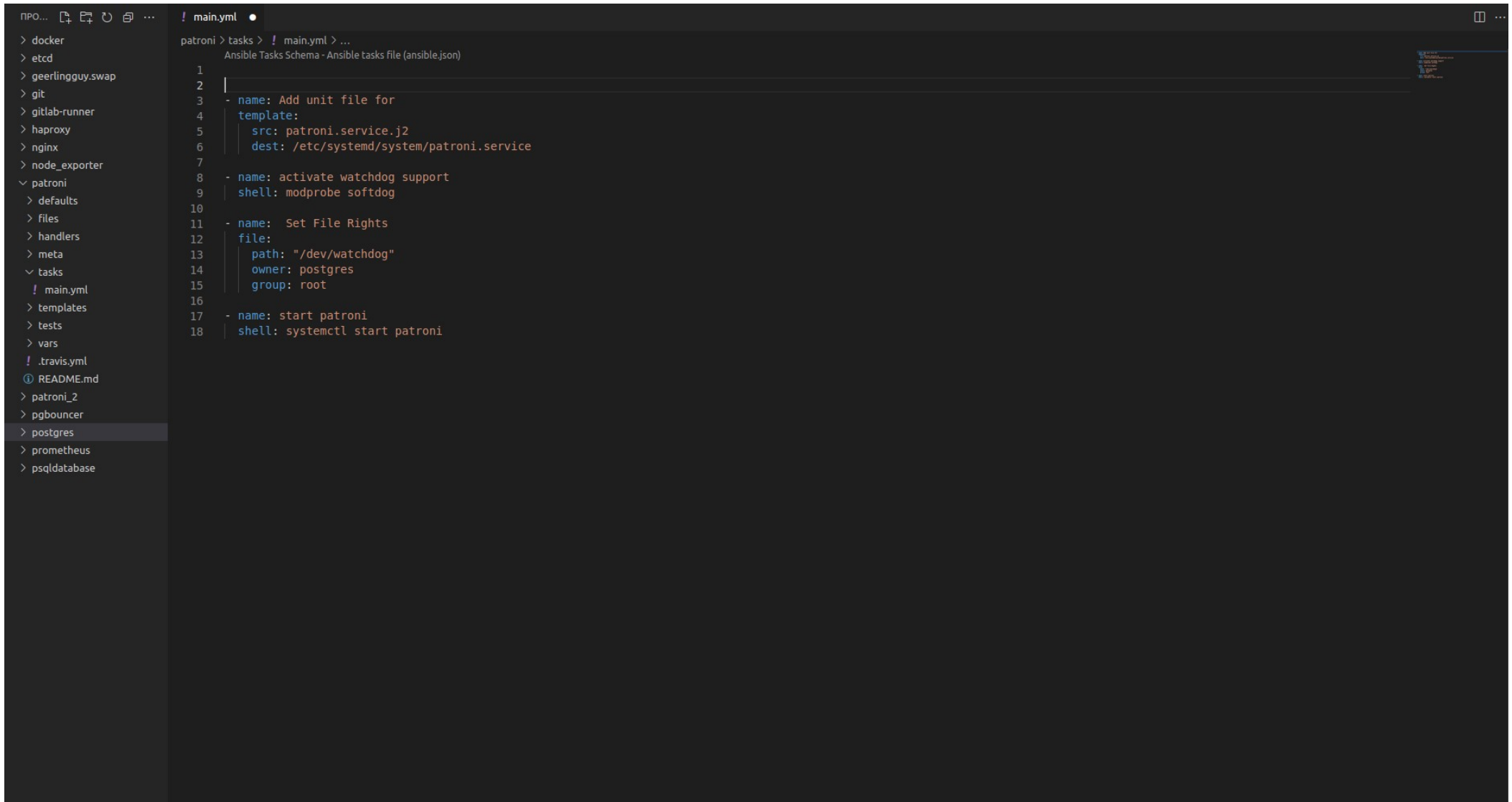
Инициализировал роль patroni 1.

В директории task в файле main.yml прописал установку всего необходимого для работы patroni. Создание конфиг файла из шаблона j2. И директории для patroni.



```
1  --|
2  # tasks file for patroni
3  - name: Install python3
4    ansible.builtin.apt:
5      pkg:
6        - python3
7        - python3-pip
8        - libpq-dev
9        - python3-dev
10       - gcc
11      state: latest
12      update_cache: yes
13
14  - name: Upgrade testresources setuptools
15    pip:
16      name: "{{ item }}"
17      extra_args: --upgrade
18      executable: pip3
19    loop:
20      - testresources
21      - setuptools
22
23  - name: Pip install patroni
24    pip:
25      name: "{{ item }}"
26      state: present
27    loop:
28      - psycopg2-binary
29      - psycopg2-binary>=2.5.4
30      - patroni[etcd]
31      - patroni
32
33  - name: create config patroni.yml
34    template:
35      src: patroni.yml.j2
36      dest: /etc/patroni.yml
37      mode: 0644
38
39  - name: Create Directories
40    file:
41      path: "{{ item }}"
42      state: directory
43      owner: postgres
44      group: postgres
45      mode: 0700
46    loop:
47      - /data/
48      - /data/patroni'
```

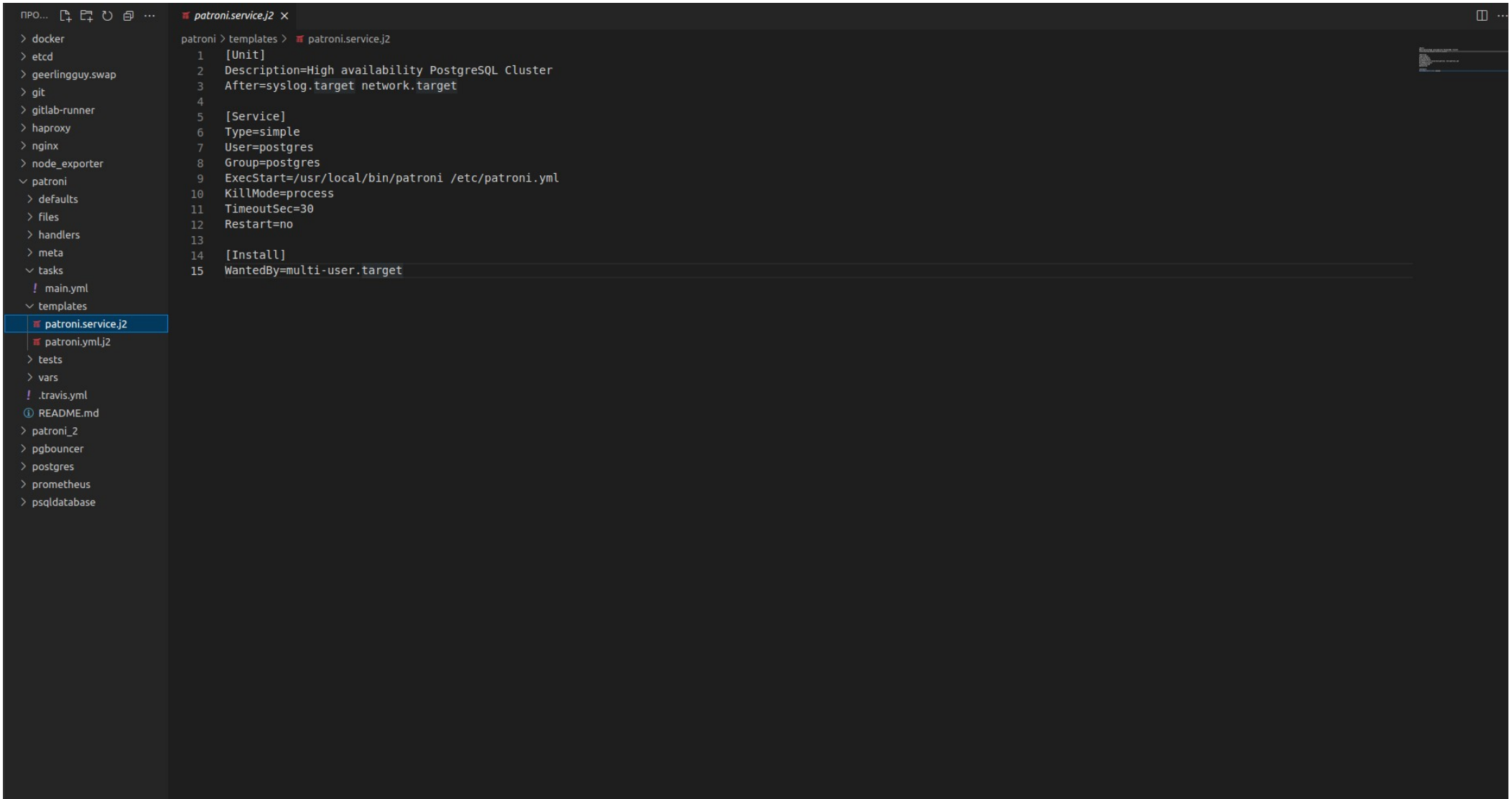
# Прописал создание юнит файла для patroni из шаблона j2. Запуск patroni.



The screenshot shows a code editor with a file explorer on the left and a main editor area. The file explorer lists various files and directories, including 'docker', 'etcd', 'geerlingguy.swap', 'git', 'gitlab-runner', 'haproxy', 'nginx', 'node\_exporter', 'patroni', 'defaults', 'files', 'handlers', 'meta', 'tasks', 'main.yml', 'templates', 'tests', 'vars', '.travis.yml', 'README.md', 'patroni\_2', 'pgbouncer', 'postgres', 'prometheus', and 'psqldatabase'. The 'main.yml' file is selected and its content is displayed in the main editor. The content is an Ansible tasks file for 'patroni' that defines tasks for adding a unit file, activating watchdog support, setting file rights, and starting the patroni service.

```
! main.yml
patroni > tasks > ! main.yml > ...
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1
2
3 - name: Add unit file for
4   template:
5     src: patroni.service.j2
6     dest: /etc/systemd/system/patroni.service
7
8 - name: activate watchdog support
9   shell: modprobe softdog
10
11 - name: Set File Rights
12   file:
13     path: "/dev/watchdog"
14     owner: postgres
15     group: root
16
17 - name: start patroni
18   shell: systemctl start patroni
```

В директории templates прописал шаблон для юнит файл patroni.

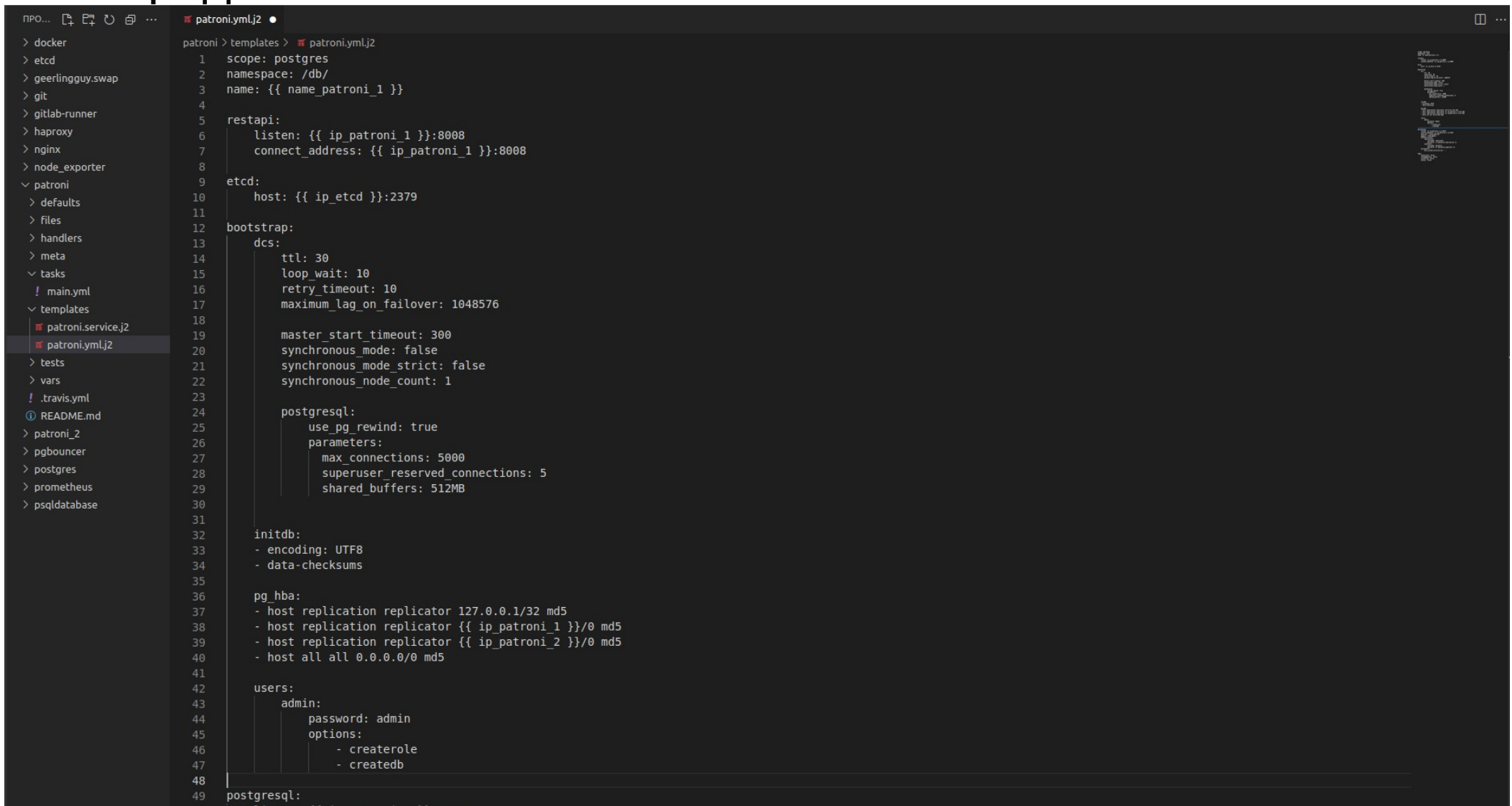


The image shows a code editor with a dark theme. On the left is a file explorer sidebar showing a directory structure. The 'templates' directory is expanded, and 'patroni.service.j2' is selected. The main editor area displays the content of this file, which is a Jinja2 template for a systemd unit. The template includes sections for [Unit], [Service], and [Install].

```
1 [Unit]
2 Description=High availability PostgreSQL Cluster
3 After=syslog.target network.target
4
5 [Service]
6 Type=simple
7 User=postgres
8 Group=postgres
9 ExecStart=/usr/local/bin/patroni /etc/patroni.yml
10 KillMode=process
11 TimeoutSec=30
12 Restart=no
13
14 [Install]
15 WantedBy=multi-user.target
```

В директории templates прописал шаблон для конфиг файл patroni.

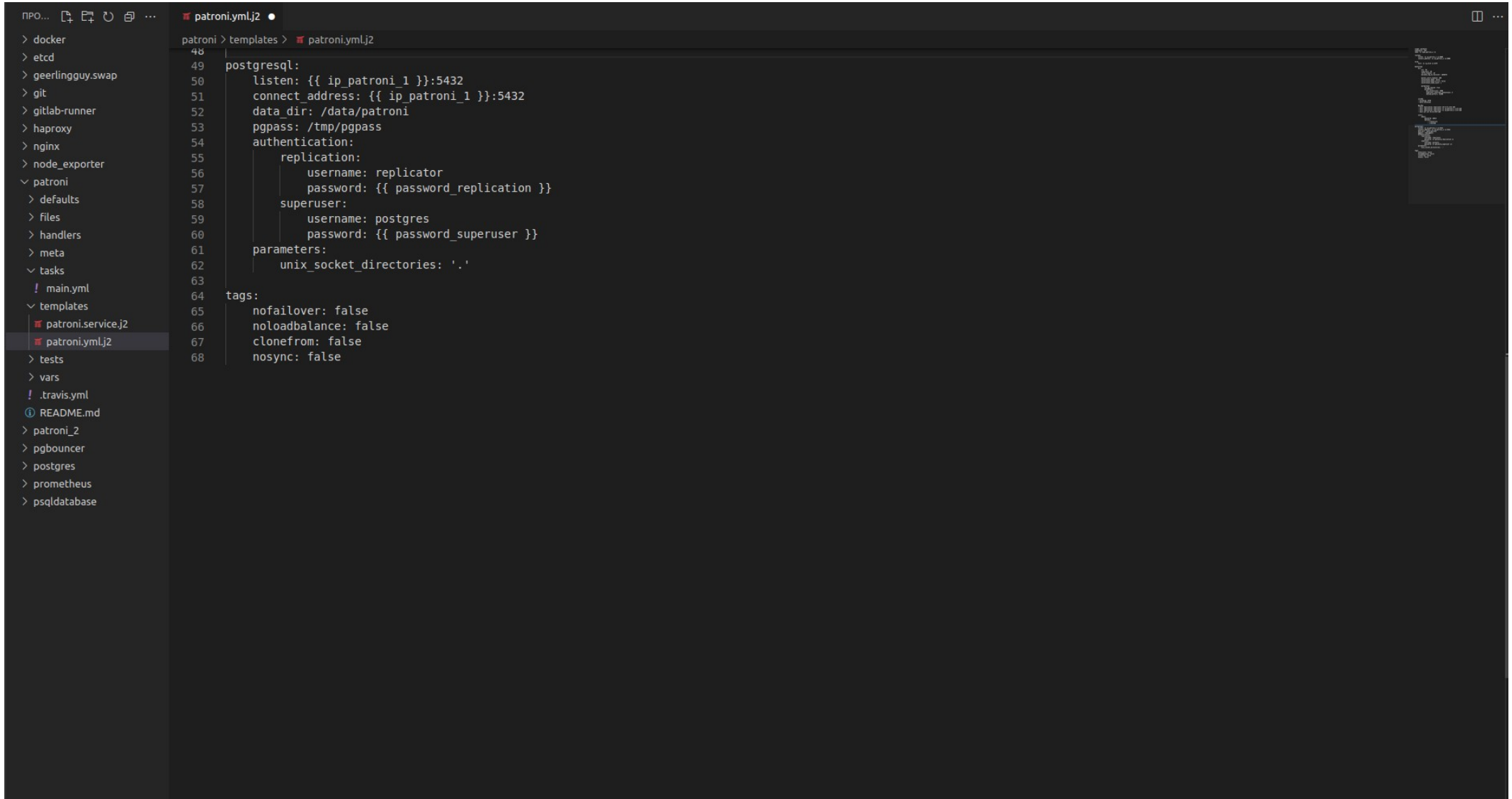
Использую переменную ip\_etcd, ip\_patroni 1, 2)  
Их определил в inventori.



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure with files like docker, etcd, geerlingguy.swap, git, gitlab-runner, haproxy, nginx, node\_exporter, patroni, defaults, files, handlers, meta, tasks, main.yml, templates, patroni.service.j2, patroni.yml.j2, tests, vars, .travis.yml, README.md, patroni\_2, pgbouncer, postgres, prometheus, and psqlatabase. The code editor shows the content of the patroni.yml.j2 file, which is a Jinja2 template for a Patroni configuration. The template includes sections for scope, namespace, name, restapi, etcd, bootstrap, dcs, postgresql, initdb, pg\_hba, users, and postgresql. The template uses variables like ip\_etcd, ip\_patroni\_1, and ip\_patroni\_2 to define configuration values.

```
1 scope: postgres
2 namespace: /db/
3 name: {{ name_patroni_1 }}
4
5 restapi:
6   listen: {{ ip_patroni_1 }}:8008
7   connect_address: {{ ip_patroni_1 }}:8008
8
9 etcd:
10   host: {{ ip_etcd }}:2379
11
12 bootstrap:
13   dcs:
14     ttl: 30
15     loop_wait: 10
16     retry_timeout: 10
17     maximum_lag_on_failover: 1048576
18
19     master_start_timeout: 300
20     synchronous_mode: false
21     synchronous_mode_strict: false
22     synchronous_node_count: 1
23
24   postgresql:
25     use_pg_rewind: true
26     parameters:
27       max_connections: 5000
28       superuser_reserved_connections: 5
29       shared_buffers: 512MB
30
31   initdb:
32     - encoding: UTF8
33     - data-checksums
34
35   pg_hba:
36     - host replication replicator 127.0.0.1/32 md5
37     - host replication replicator {{ ip_patroni_1 }}/0 md5
38     - host replication replicator {{ ip_patroni_2 }}/0 md5
39     - host all all 0.0.0.0/0 md5
40
41   users:
42     admin:
43       password: admin
44       options:
45         - createrole
46         - createdb
47
48   postgresql:
49     listen: {{ ip_patroni_1 }}:5432
```

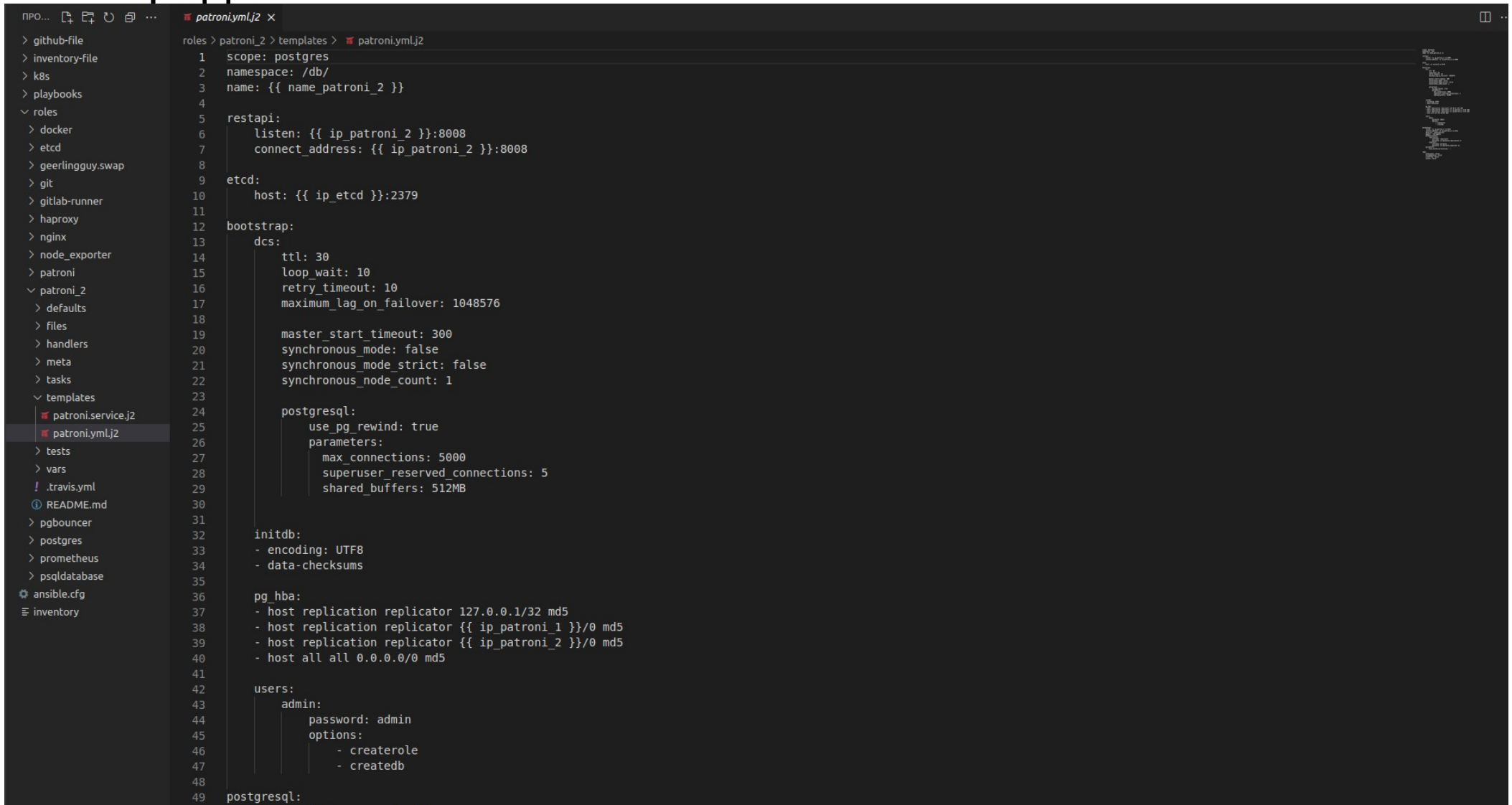
Так же использую переменные:  
password\_replication  
password\_superuser.  
Они так же прописаны в inventori.



```
48
49 postgresql:
50   listen: {{ ip_patroni_1 }}:5432
51   connect_address: {{ ip_patroni_1 }}:5432
52   data_dir: /data/patroni
53   pgpass: /tmp/pgpass
54   authentication:
55     replication:
56       username: replicator
57       password: {{ password_replication }}
58     superuser:
59       username: postgres
60       password: {{ password_superuser }}
61   parameters:
62     unix_socket_directories: '.'
63
64   tags:
65     nofailover: false
66     noloadbalance: false
67     clonefrom: false
68     nosync: false
```



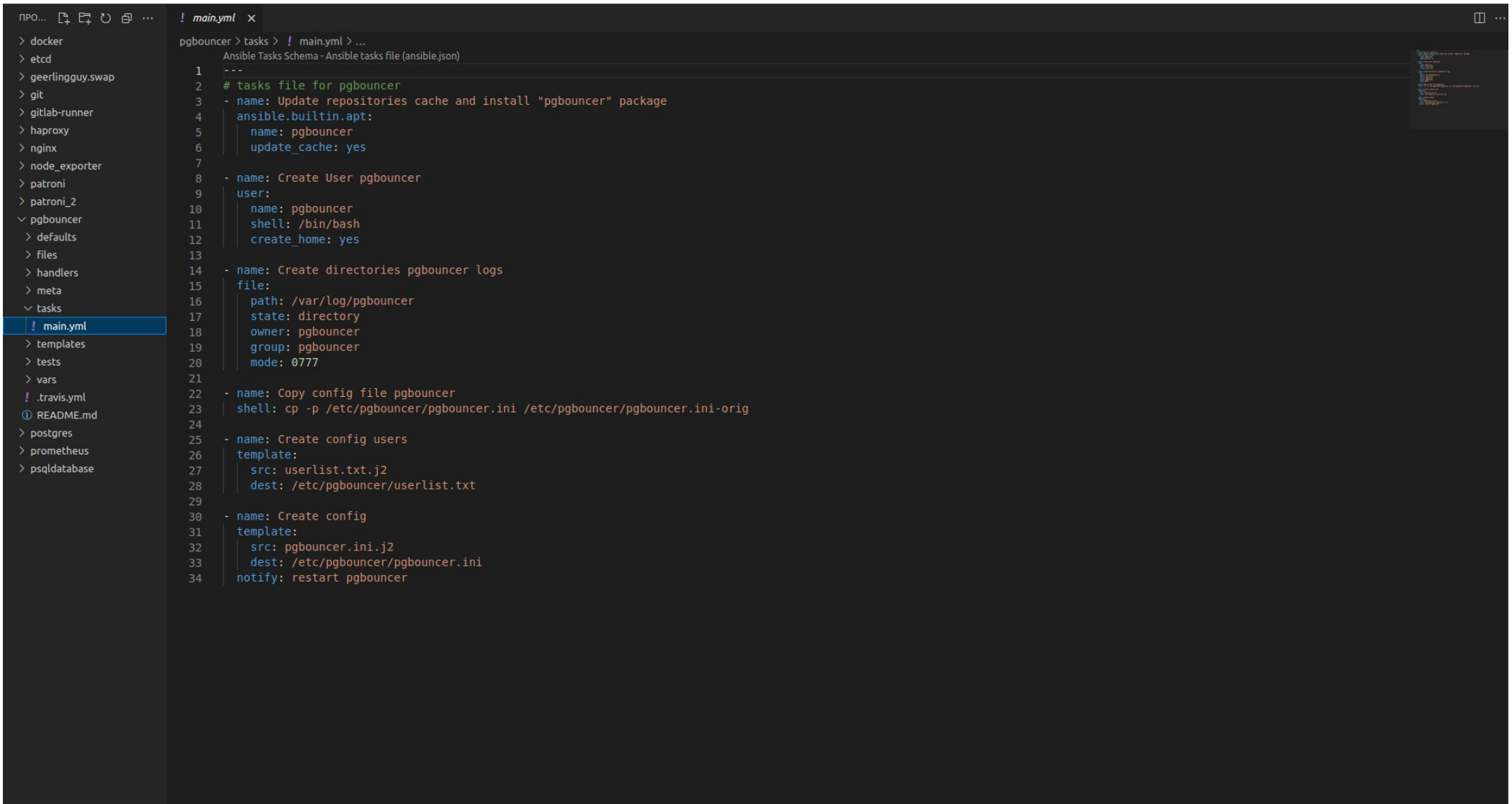
Инициализировал роль patroni 2. Отличие этой роли от роли patroni 1 только в шаблоне для конфиг файл patroni. Использую переменную ip\_etcd, ip\_patroni 1, 2 Их определил в inventori.



```
roles > patroni_2 > templates > patroni.ymlj2
1  scope: postgres
2  namespace: /db/
3  name: {{ name_patroni_2 }}
4
5  restapi:
6    listen: {{ ip_patroni_2 }}:8008
7    connect_address: {{ ip_patroni_2 }}:8008
8
9  etcd:
10    host: {{ ip_etcd }}:2379
11
12  bootstrap:
13    dcs:
14      ttl: 30
15      loop_wait: 10
16      retry_timeout: 10
17      maximum_lag_on_failover: 1048576
18
19      master_start_timeout: 300
20      synchronous_mode: false
21      synchronous_mode_strict: false
22      synchronous_node_count: 1
23
24      postgresql:
25        use_pg_rewind: true
26        parameters:
27          max_connections: 5000
28          superuser_reserved_connections: 5
29          shared_buffers: 512MB
30
31      initdb:
32        - encoding: UTF8
33        - data-checksums
34
35      pg_hba:
36        - host replication replicator 127.0.0.1/32 md5
37        - host replication replicator {{ ip_patroni_1 }}/0 md5
38        - host replication replicator {{ ip_patroni_2 }}/0 md5
39        - host all all 0.0.0.0/0 md5
40
41      users:
42        admin:
43          password: admin
44          options:
45            - createrole
46            - createdb
47
48      postgresql:
49
```

# Инициализировал роль pgbouncer

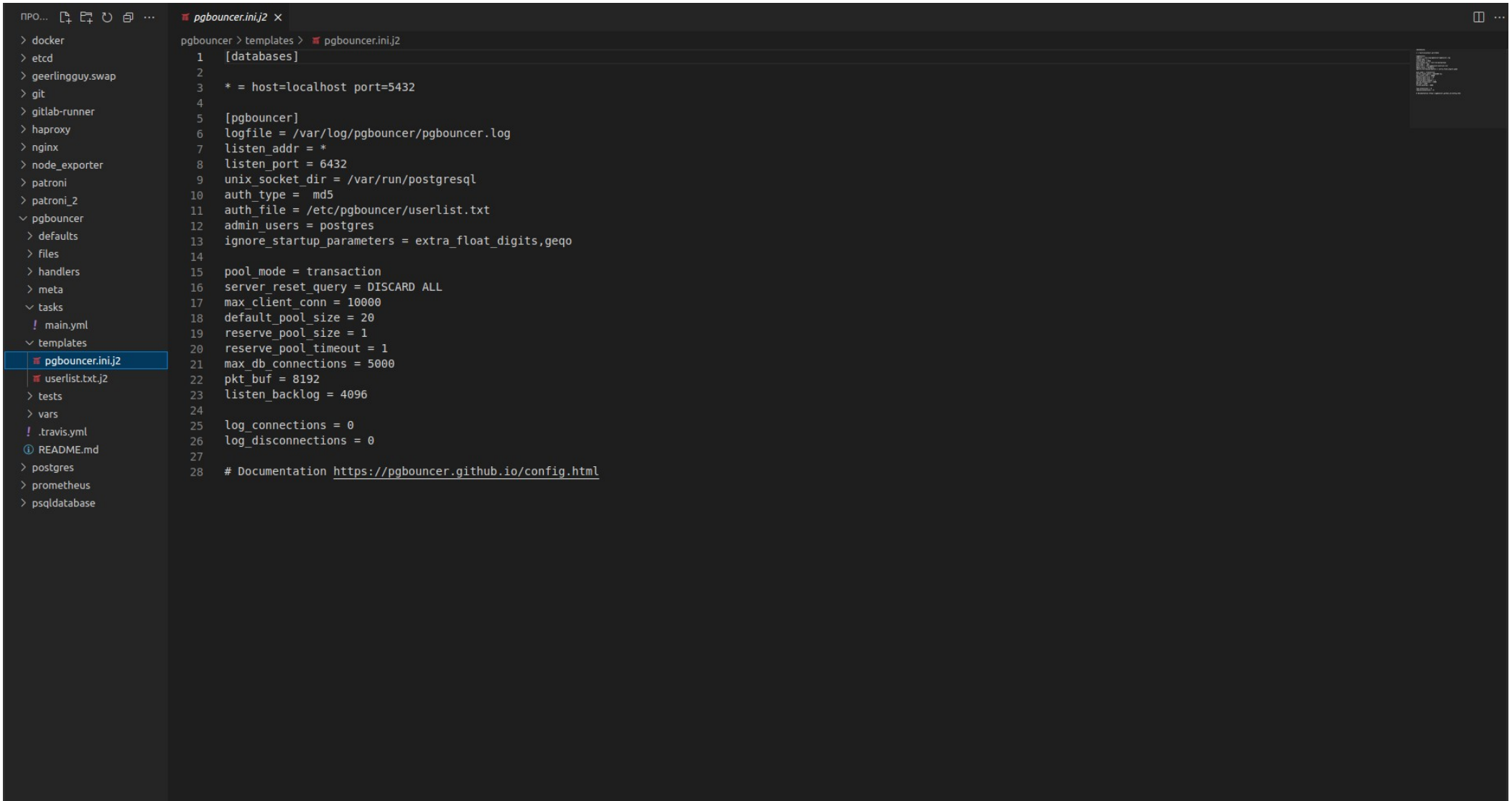
В директории task в файле main.yml прописал установку pgbouncer и создание для него юзера, директории логов и конфиг файлов из шаблонов j2.



```
! main.yml x
> docker
> etcd
> geerlingguy.swap
> git
> gitlab-runner
> haproxy
> nginx
> node_exporter
> patroni
> patroni_2
> pgbouncer
  > defaults
  > files
  > handlers
  > meta
  > tasks
! main.yml
  > templates
  > tests
  > vars
! .travis.yml
① README.md
> postgres
> prometheus
> psqldatabase

pgbouncer > tasks > ! main.yml > ...
Ansible Tasks Schema - Ansible tasks file (ansible.json)
1  ---
2  # tasks file for pgbouncer
3  - name: Update repositories cache and install "pgbouncer" package
4    ansible.builtin.apt:
5      name: pgbouncer
6      update_cache: yes
7
8  - name: Create User pgbouncer
9    user:
10     name: pgbouncer
11     shell: /bin/bash
12     create_home: yes
13
14  - name: Create directories pgbouncer logs
15    file:
16     path: /var/log/pgbouncer
17     state: directory
18     owner: pgbouncer
19     group: pgbouncer
20     mode: 0777
21
22  - name: Copy config file pgbouncer
23    shell: cp -p /etc/pgbouncer/pgbouncer.ini /etc/pgbouncer/pgbouncer.ini-orig
24
25  - name: Create config users
26    template:
27     src: userlist.txt.j2
28     dest: /etc/pgbouncer/userlist.txt
29
30  - name: Create config
31    template:
32     src: pgbouncer.ini.j2
33     dest: /etc/pgbouncer/pgbouncer.ini
34    notify: restart pgbouncer
```

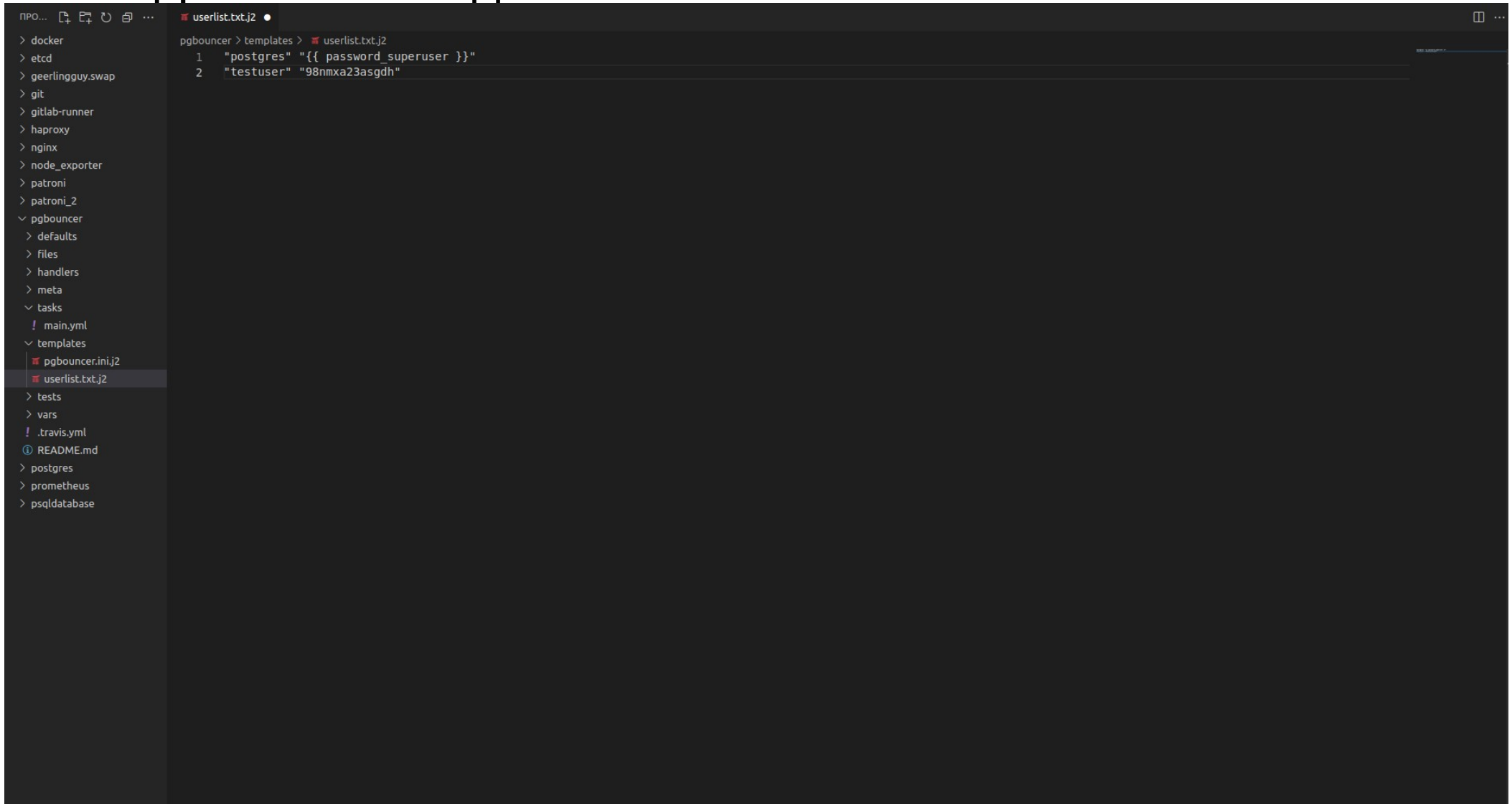
В директории templates прописал шаблон для конфиг файл pgbouncer.



The screenshot shows a code editor with a sidebar on the left containing a file tree. The tree includes folders like 'docker', 'etcd', 'geerlingguy.swap', 'git', 'gitlab-runner', 'haproxy', 'nginx', 'node\_exporter', 'patroni', 'patroni\_2', 'pgbouncer', 'tasks', and 'templates'. The 'pgbouncer' folder is expanded, showing 'defaults', 'files', 'handlers', 'meta', 'tasks', and 'templates'. The 'templates' folder is selected, and the file 'pgbouncer.ini.j2' is highlighted. The main editor area displays the content of 'pgbouncer.ini.j2', which is a Jinja2 template for the pgbouncer configuration file. The template includes sections for '[databases]', '[pgbouncer]', and a documentation link at the bottom.

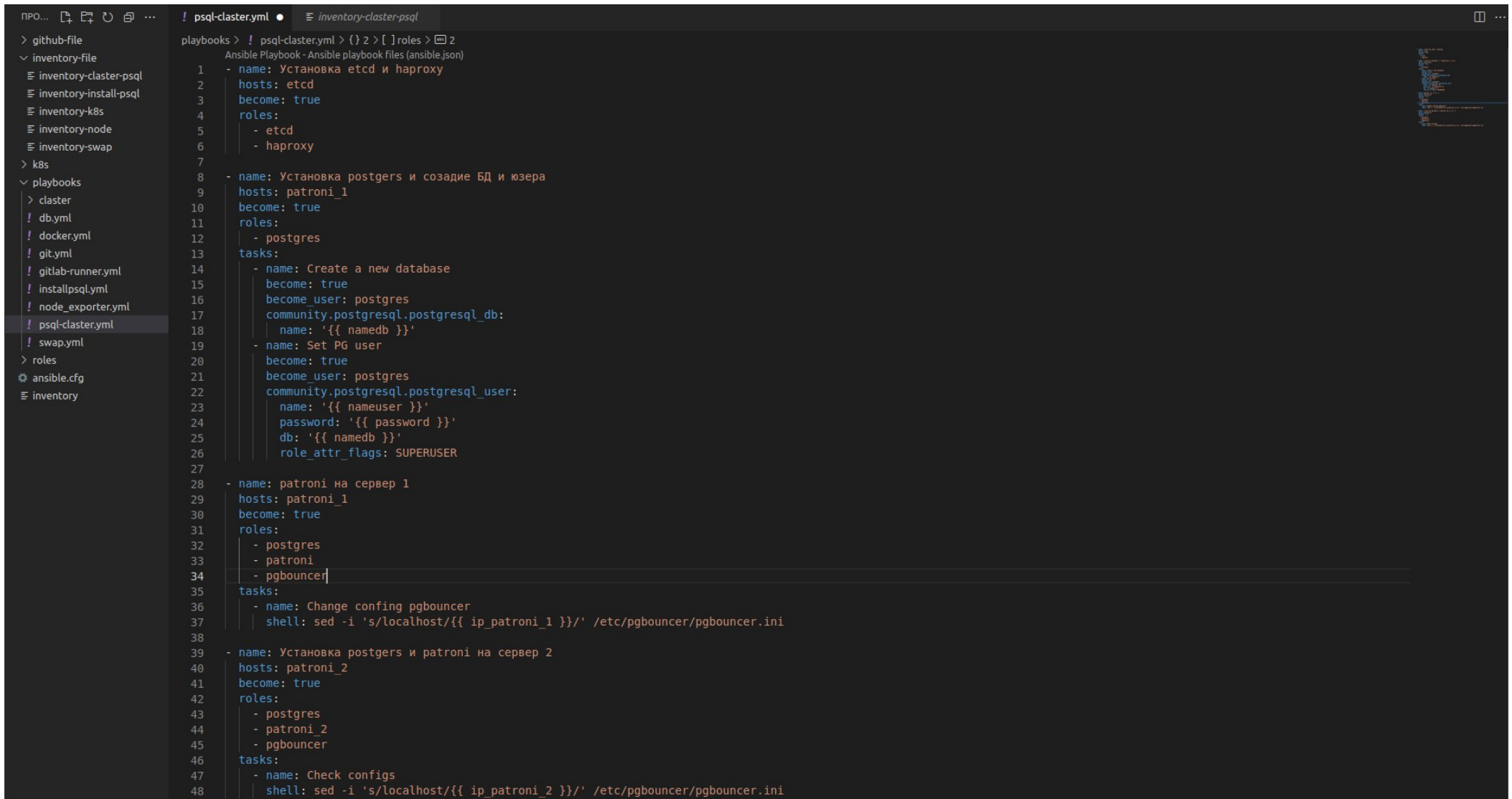
```
1 [databases]
2
3 * = host=localhost port=5432
4
5 [pgbouncer]
6 logfile = /var/log/pgbouncer/pgbouncer.log
7 listen_addr = *
8 listen_port = 6432
9 unix_socket_dir = /var/run/postgresql
10 auth_type = md5
11 auth_file = /etc/pgbouncer/userlist.txt
12 admin_users = postgres
13 ignore_startup_parameters = extra_float_digits,geqo
14
15 pool_mode = transaction
16 server_reset_query = DISCARD ALL
17 max_client_conn = 10000
18 default_pool_size = 20
19 reserve_pool_size = 1
20 reserve_pool_timeout = 1
21 max_db_connections = 5000
22 pkt_buf = 8192
23 listen_backlog = 4096
24
25 log_connections = 0
26 log_disconnections = 0
27
28 # Documentation https://pgbouncer.github.io/config.html
```

Так же создал шаблон для списка юзеров.  
В нем используются переменная password\_superuser.  
Указал тестового юзера и пароль для него. Их можно указать в зашифрованном виде.



```
userlist.txt.j2
1 "postgres" "{{ password_superuser }}"
2 "testuser" "98nmxa23asgdh"
```

Написал плейбук для установки кластера объединяющий в себе роли etcd, haproxy, postgres, patroni, pgbouncer, а так же создание тестовой БД и юзера.



```
! psql-cluster.yml • inventory-cluster-psql
playbooks > ! psql-cluster.yml > {} 2 > [ ] roles > 2
Ansible Playbook - Ansible playbook files (ansible.json)
1 - name: Установка etcd и haproxy
2   hosts: etcd
3   become: true
4   roles:
5     - etcd
6     - haproxy
7
8 - name: Установка postgers и созадие БД и юзера
9   hosts: patroni_1
10  become: true
11  roles:
12    - postgres
13  tasks:
14    - name: Create a new database
15      become: true
16      become_user: postgres
17      community.postgresql.postgresql_db:
18        name: '{{ namedb }}'
19
20    - name: Set PG user
21      become: true
22      become_user: postgres
23      community.postgresql.postgresql_user:
24        name: '{{ nameuser }}'
25        password: '{{ password }}'
26        db: '{{ namedb }}'
27        role_attr_flags: SUPERUSER
28
29    - name: patroni на сервер 1
30      hosts: patroni_1
31      become: true
32      roles:
33        - postgres
34        - patroni
35        - pgbouncer
36
37      tasks:
38        - name: Change confing pgbouncer
39          shell: sed -i 's/localhost/{{ ip_patroni_1 }}/ /etc/pgbouncer/pgbouncer.ini
40
41
42    - name: Установка postgers и patroni на сервер 2
43      hosts: patroni_2
44      become: true
45      roles:
46        - postgres
47        - patroni_2
48        - pgbouncer
49
50      tasks:
51        - name: Check configs
52          shell: sed -i 's/localhost/{{ ip_patroni_2 }}/ /etc/pgbouncer/pgbouncer.ini
```

# Развернул три сервера

<div>SERVERS</div> <div>SNAPSHOTS</div> <div>BACKUPS</div> <div>PLACEMENT GROUPS</div> <div>PRIMARY IPS</div>					<div></div> <div>Add Server</div>	
<input type="checkbox"/>	Name	Public IP	Location	Created		
<input type="checkbox"/>	<div><div><div></div></div><div><div>psql-1</div><div>CX11 / 20 GB / eu-central</div></div></div>	65.109.143.81	<div><div></div><div>Helsinki</div></div>	less than a minute ago	<div></div>	
<input type="checkbox"/>	<div><div><div></div></div><div><div>psql-2</div><div>CX11 / 20 GB / eu-central</div></div></div>	65.21.177.102	<div><div></div><div>Helsinki</div></div>	less than a minute ago	<div></div>	
<input type="checkbox"/>	<div><div><div></div></div><div><div>ectd-haproxy</div><div>CX11 / 20 GB / eu-central</div></div></div>	65.109.175.31	<div><div></div><div>Helsinki</div></div>	less than a minute ago	<div></div>	

# Создал inventory файл прописал в нем переменные

```
inventory-cluster-psql
ansible > inventory-file > inventory-cluster-psql
1  [Cluster]
2  etcd ansible_host=65.109.175.31
3  patroni_1 ansible_host=65.109.143.81
4  patroni_2 ansible_host=65.21.177.102
5
6  [Cluster:vars]
7  #####переменные etcd
8      ip_etcd=65.109.175.31
9  #####переменные haproxy
10     ip_patroni_1=65.109.143.81
11     ip_patroni_2=65.21.177.102
12 #####переменные для patroni и postgres
13     nameuser=testuser
14     namedb=testdb
15     password=98nmxa23asgdh
16     name_patroni_1=patroni_1
17     name_patroni_2=patroni_2
18     password_replication=QWEasdzxc
19     password_superuser=QWEasdzxc
20 #####для pgbouncer
21 #пользователь postgres пароль QWEasdzxc
22
```

# Запустил плейбук

```
ПРОБЛЕМЫ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ВЫХОДНЫЕ ДАННЫЕ

ansible
├── github-file
├── inventory-file
├── k8s
├── playbooks
├── roles
├── ansible.cfg
└── inventory

-;tcp_keepintvl = 0
-
-;; How long may transmitted data remain unacknowledged before TCP
-;; connection is closed (in milliseconds)
-;tcp_user_timeout = 0
-
-;; DNS lookup caching time
-;dns_max_ttl = 15
-
-;; DNS zone SOA lookup period
-;dns_zone_check_period = 0
-
-;; DNS negative result caching time
-;dns_nxdomain_ttl = 15
-
-;; Custom resolv.conf file, to set custom DNS servers or other options
-;; (default: empty = use OS settings)
-;resolv_conf = /etc/pgbouncer/resolv.conf
-
-;;;
-;;; Random stuff
-;;;
-
-;; Hackish security feature.  Helps against SQL injection: when PQexec
-;; is disabled, multi-statement cannot be made.
-;disable_pqexec = 0
-
-;; Config file to use for next RELOAD/SIGHUP
-;; By default contains config file from command line.
-;conffile
-
-;; Windows service name to register as.  job_name is alias for
-;; service_name, used by some Skytools scripts.
-;service_name = pgbouncer
-;job_name = pgbouncer
-
-;; Read additional config from other file
-;%include /etc/pgbouncer/pgbouncer-other.ini
-# Documentation https://pgbouncer.github.io/config.html
\ No newline at end of file

changed: [patroni_2]

TASK [Check configs] *****
[WARNING]: Skipping plugin (/usr/lib/python3/dist-packages/ansible/plugins/filter/core.py) as it seems to be invalid: cannot import name 'environmentfilter' from 'jinja2.filters'
(/usr/local/lib/python3.8/dist-packages/jinja2/filters.py)
[WARNING]: Skipping plugin (/usr/lib/python3/dist-packages/ansible/plugins/filter/mathstuff.py) as it seems to be invalid: cannot import name 'environmentfilter' from 'jinja2.filters'
(/usr/local/lib/python3.8/dist-packages/jinja2/filters.py)
changed: [patroni_2]

RUNNING HANDLER [pgbouncer : restart pgbouncer] *****
changed: [patroni_2]

PLAY RECAP *****
etcd                : ok=9   changed=8   unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
patroni_1           : ok=25  changed=24  unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
patroni_2           : ok=25  changed=24  unreachable=0   failed=0   skipped=0   rescued=0   ignored=0

alex@alex:~/myDoc/ansible$
```



По ip адресу сервера с haproxy на порту 7000 есть страница со статистикой.

Видим что patroni 1 сейчас master

Server · Hetzner Cloud

Statistics Report for HAPr

+

← → ↺ Не защищено | 65.109.175.31:7000

✉ Почта 📺 YouTube 📁 учебные 📁 git 📁 rabota 📁 прочее 📁 дом

# HAProxy version 2.2.9-2+deb11u3, released 2022/03/10

## Statistics Report for pid 2562

> General process information

pid = 2562 (process #1, nbproc = 1, nbthread = 1)

uptime = 0d 0h07m27s

system limits: memmax = unlimited; ulimit-n = 200041

maxsock = 200041; maxconn = 100000; maxpipes = 0

current conns = 1; current pipes = 0/0; conn rate = 1/sec; bit rate = 0.271 kbps

Running tasks: 1/12; idle = 100 %

active UP

active UP, going down

active DOWN, going up

active or backup DOWN

active or backup DOWN for maintenance (MAINT)

active or backup SOFT STOPPED for maintenance

backup UP

backup UP, going down

backup DOWN, going up

not checked

Display option:

Scope :

Hide 'DOWN' servers

Refresh now

CSV export

JSON export (schema)

External resources:

Primary site

Updates (v2.2)

Online manual

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

stats

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				1	1	-	1	1	100 000	1			0	0	0	0	0					OPEN									
Backend	0	0		0	0		0	0	10 000	0	0	0s	0	0	0	0		0	0	0	0	7m27s UP		0	0	0			0		

postgres

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				0	0	-	0	0	10 000	0			0	0	0	0	0					OPEN									
patroni1	0	0	-	0	0		0	0	-	0	0	?	0	0		0	0	0	0	0	0	4m57s UP	L7OK/200 in 3ms	1	Y	-	1	1	2m30s	-	
patroni2	0	0	-	0	0		0	0	-	0	0	?	0	0		0	0	0	0	0	0	7m26s DOWN	L7STS/503 in 3ms	1	Y	-	1	1	7m26s	-	
Backend	0	0		0	0		0	0	1 000	0	0	?	0	0	0	0		0	0	0	0	4m57s UP		1	1	0		1	2m29s		

Через patronictl посмотрел состояние кластера. Patroni 1 master.

```
root@psql-1:~# patronictl -c /etc/patroni.yml list
+ Cluster: postgres (7193225461450939510) -----+-----+-----+
| Member      | Host           | Role    | State  | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| patroni_1    | 65.109.143.81  | Leader  | running | 3 |           |
| patroni_2    | 65.21.177.102  | Replica | running | 3 |          0 |
+-----+-----+-----+-----+-----+-----+
```

Подключаюсь к БД, для проверки, на ip адрес hapроxy порт 5000 указав юзера и БД из инвентори. БД доступна.

Делаю рестарт patroni 1 и снова захожу в БД. БД доступна.

Рестарта достаточно для того что бы произошел переход.

Это видно по вкладке со статистикой. Patroni 1 стал репликой, а patroni 2 мастером. По этому доступ к БД остался.

```
root@etcd-haproxy:~# psql -h 65.109.175.31 -p 5000 -U testuser -d test
Password for user testuser:
psql (13.9 (Debian 13.9-0+deb11u1), server 11.18 (Debian 11.18-1.pgdg110+1))
Type "help" for help.
```

test=> \q

```
root@etcd-haproxy:~# ssh root@65.109.143.81 "service patroni restart"
```

```
root@etcd-haproxy:~# psql -h 65.109.175.31 -p 5000 -U testuser -d test
```

Password for user testuser:

```
psql (13.9 (Debian 13.9-0+deb11u1), server 11.18 (Debian 11.18-1.pgdg110+1))
```

Type "help" for help.

test=> \q

postgres																															
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Status	Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis		LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle	
Frontend				0	1	-	0	1	10 000	10			1 105	1 910	0	0	0				OPEN										
patroni1	0	0	-	0	1		0	1	-	7	7	6s	795	1 514		0		0	0	0	0	5s DOWN	L7STS/503 in 4ms	1	Y	-	10	4	5m32s	-	
patroni2	0	0	-	0	1		0	1	-	3	3	5m35s	310	396		0		0	0	0	0	8s UP	L7OK/200 in 4ms	1	Y	-	7	3	20m50s	-	
Backend	0	0		0	1		0	1	1 000	10	10	6s	1 105	1 910	0	0		0	0	0	0	21m33s UP		1	1	0		1	2m29s		

Через patronictl посмотрел состояние кластера  
Теперь Patroni 2 master.

```
root@psql-1:~# patronictl -c /etc/patroni.yml list
+ Cluster: postgres (7193225461450939510) -----+-----+
| Member   | Host           | Role   | State  | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| patroni_1 | 65.109.143.81 | Replica | running | 4 |          0 |
| patroni_2 | 65.21.177.102 | Leader  | running | 4 |          |
+-----+-----+-----+-----+-----+-----+
root@psql-1:~#
```

Делаю рестарт patroni 2 и захожу в БД. БД доступна.  
В окне со статистикой видно что произошел переход.  
Patroni 1 стал снова мастером, а patroni 2 репликой по этому  
доступ к БД остался.

```
root@etcd-haproxy:~# ssh root@65.21.177.102 "service patroni restart"
root@etcd-haproxy:~# psql -h 65.109.175.31 -p 5000 -U testuser -d test
Password for user testuser:
psql (13.9 (Debian 13.9-0+deb11u1), server 11.18 (Debian 11.18-1.pgdg110+1))
Type "help" for help.

test=> \q
root@etcd-haproxy:~#
```

postgres			Queue			Session rate			Sessions					Bytes		Denied		Errors		Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle		
Frontend				0	1	-	0	1	10 000	14			1 503	2 320	0	0	0					OPEN										
patroni1	0	0	-	0	1		0	1	-	9	9	1m13s	1 017	1 896		0		0	0	0	0	1m27s UP	L7OK/200 in 3ms	1	Y	-	10	4	6m12s	-		
patroni2	0	0	-	0	1		0	1	-	5	5	1m28s	486	424		0		0	0	0	0	1m26s DOWN	L7STS/503 in 3ms	1	Y	-	10	4	22m16s	-		
Backend	0	0		0	1		0	1	1 000	14	14	1m13s	1 503	2 320	0	0		0	0	0	0	23m40s UP		1	1	0		1	2m29s			