Pgbouncer/patroni/postgres

Дано:
- 3 сервера с ОС Debian

Задание:
- Написать плэйбуки для ansible на установку кластера pgbouncer/patroni/postgres

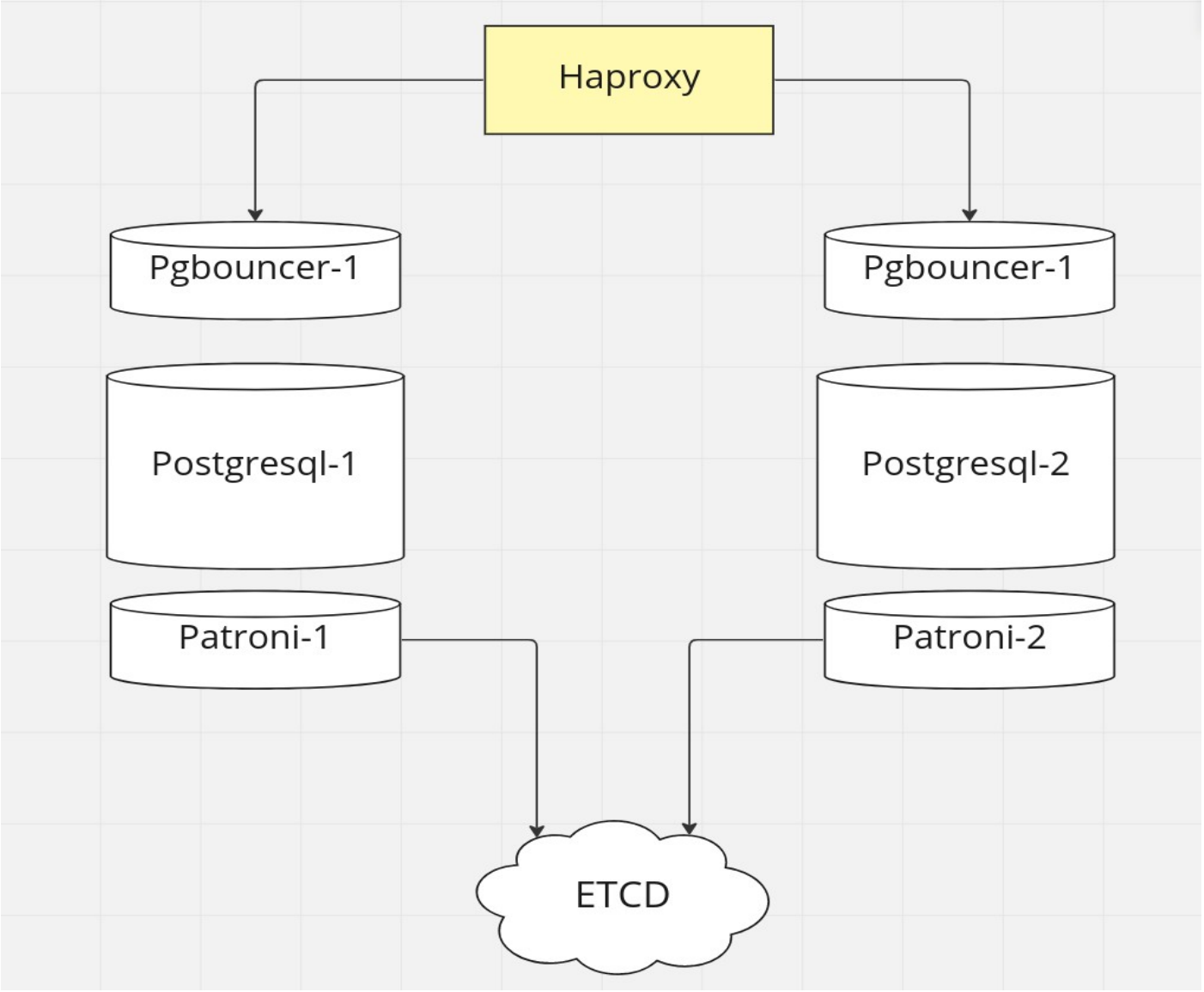
План выполнения:
- написать роль etcd
- написать роль haproxy
- написать роль postgres
- написать роль patroni
- написать роль patroni 2
- написать роль pgbouncer
- написать плэйбук разворачивающий кластер используя ранее написанные роли.
- развернуть 3 сервера
- создать инвентори файл и указать там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

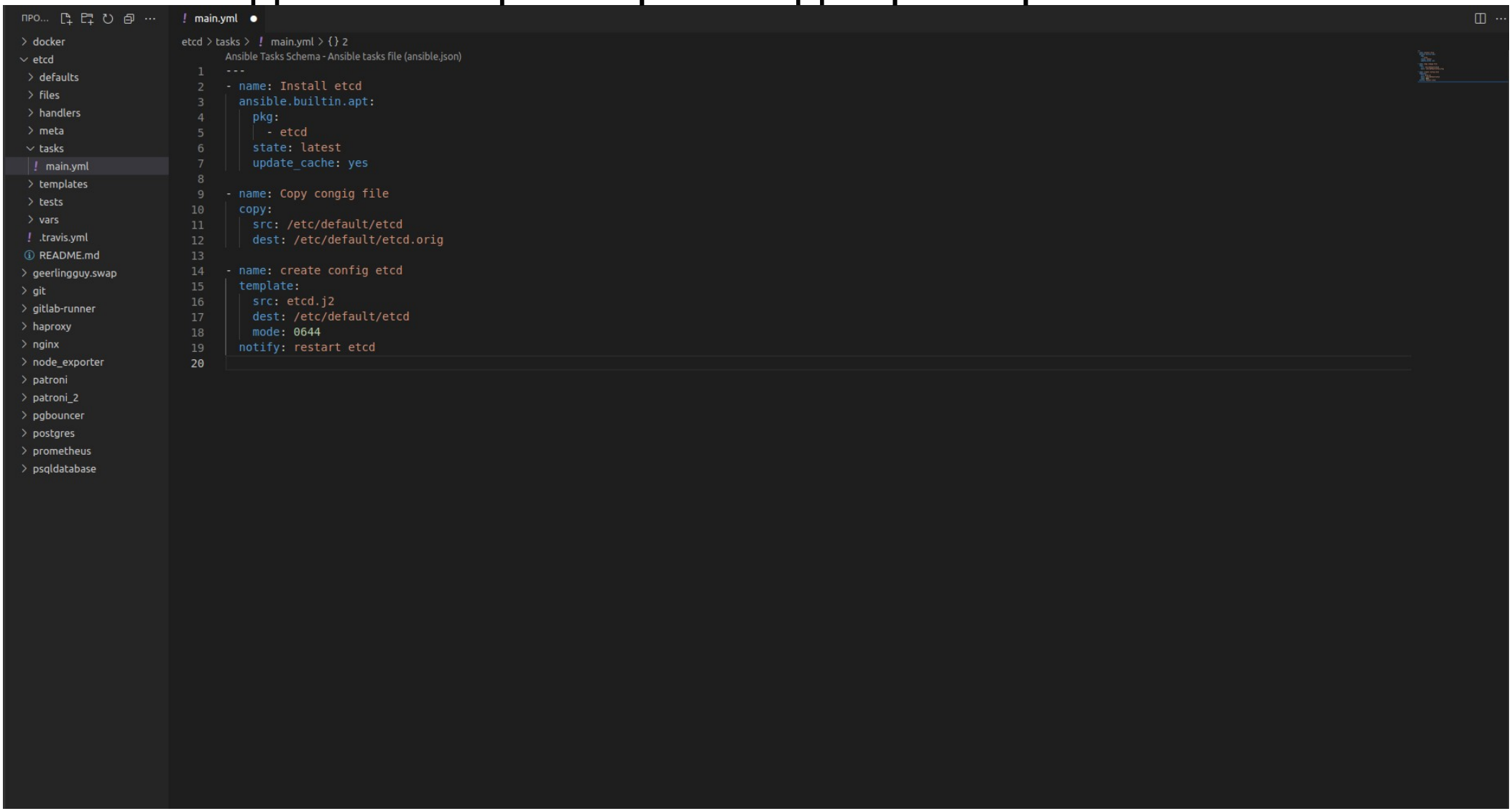
Haproxy это бесплатное, очень быстрое и надежное решение, предлагающее высокую доступность, балансировку нагрузки и прокси для приложений на основе TCP и HTTP.

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.

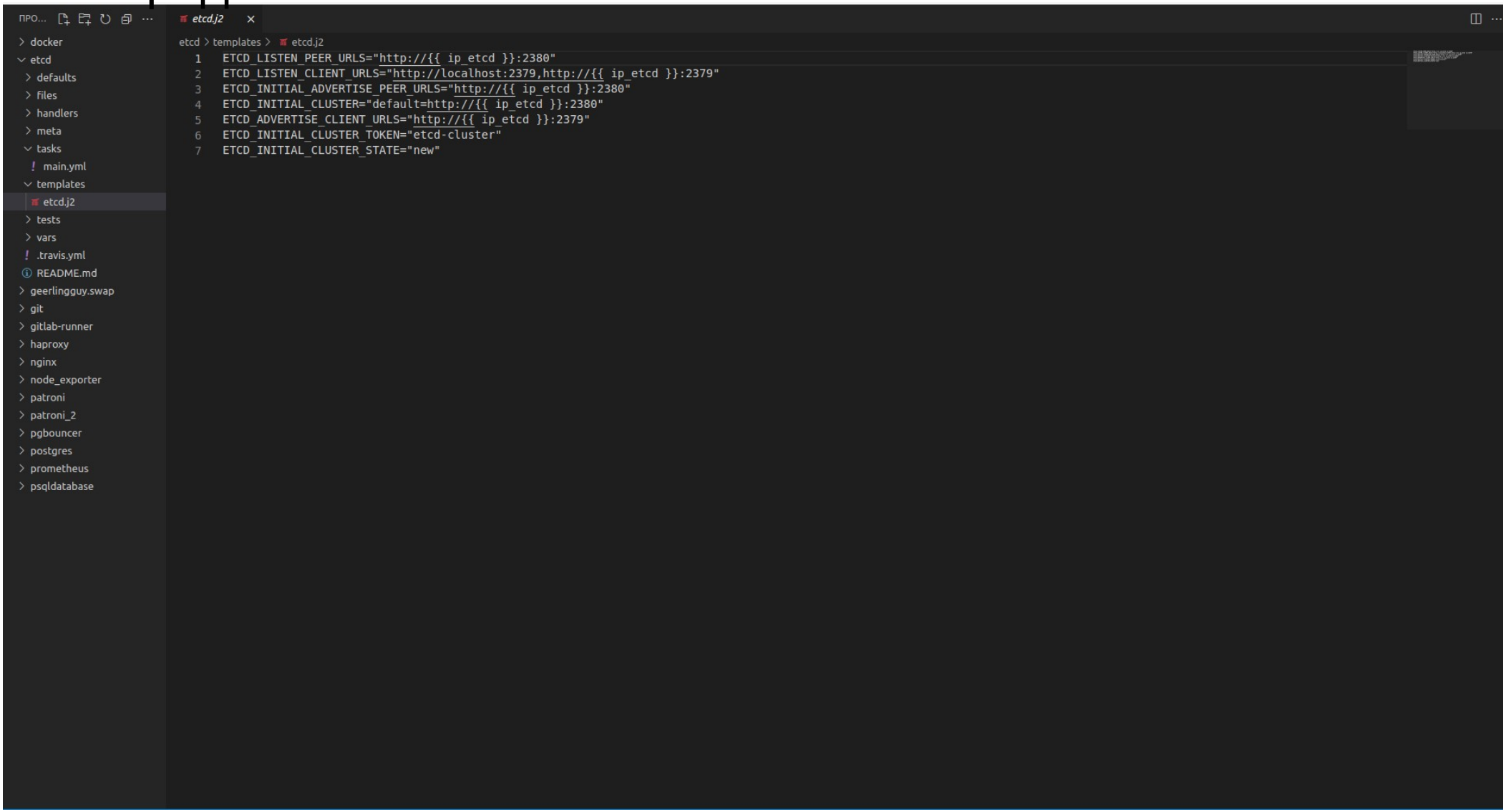
Принципиальная схема кластера.



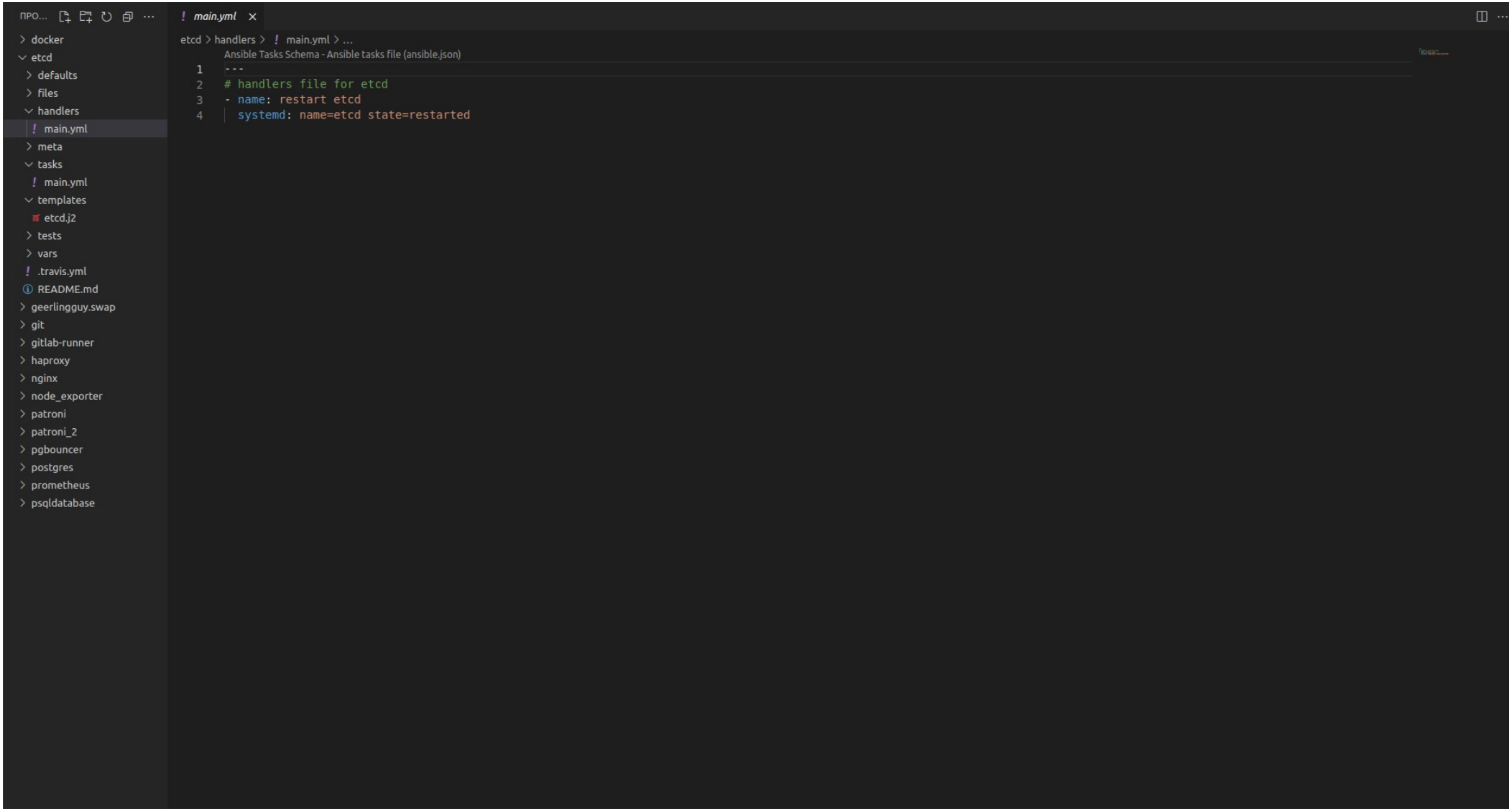
Инициализировал роль etcd
В директории task в файле main.yaml прописал установку etcd и создание конфига из шаблона j2 с сохранением оригинала. После создания конфига происходит рестарт etcd.



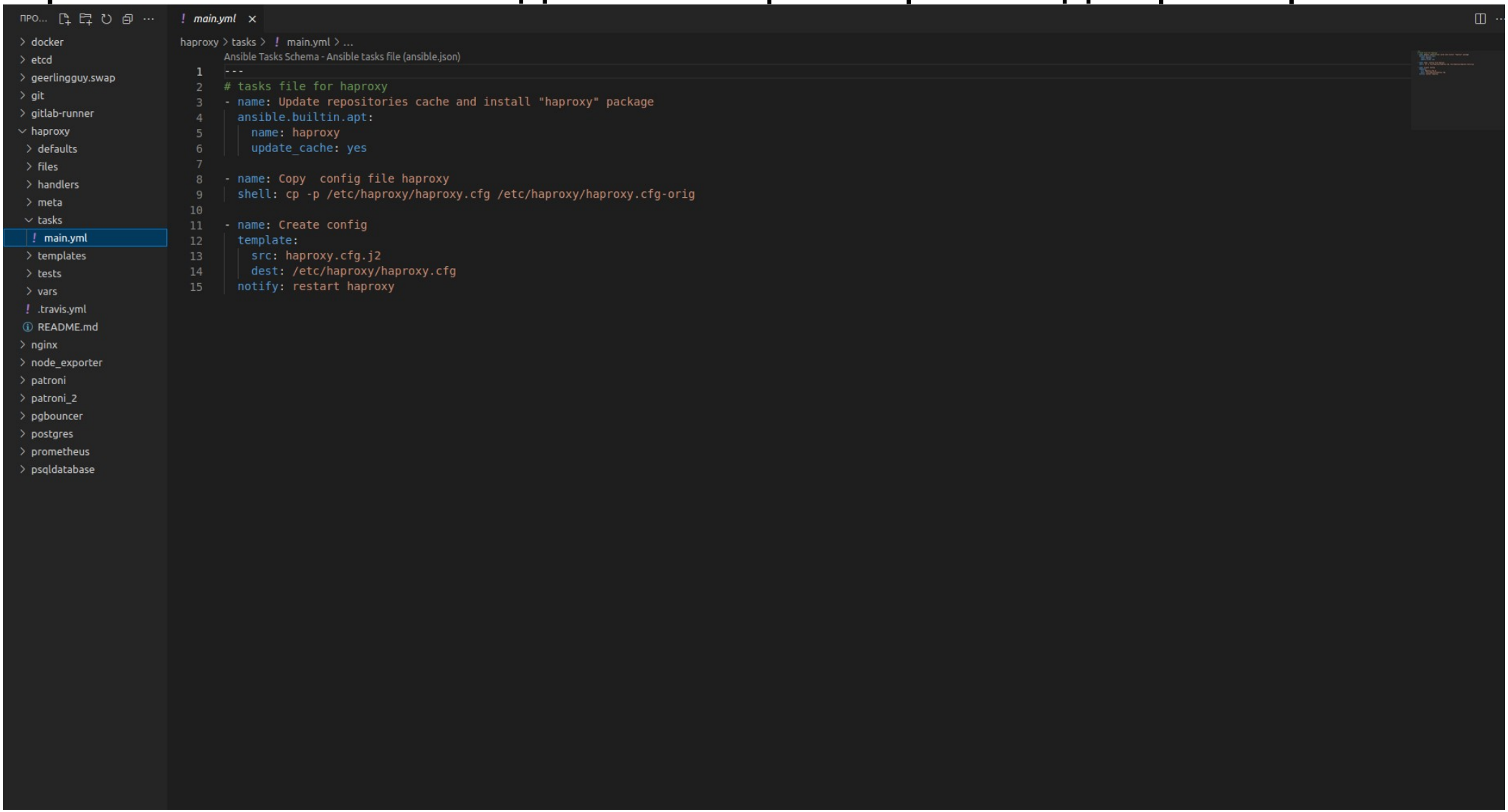
В директории templates прописал шаблон для конфиг файла etcd.
Использую переменную ip_etcd.
Её определил в inventori.



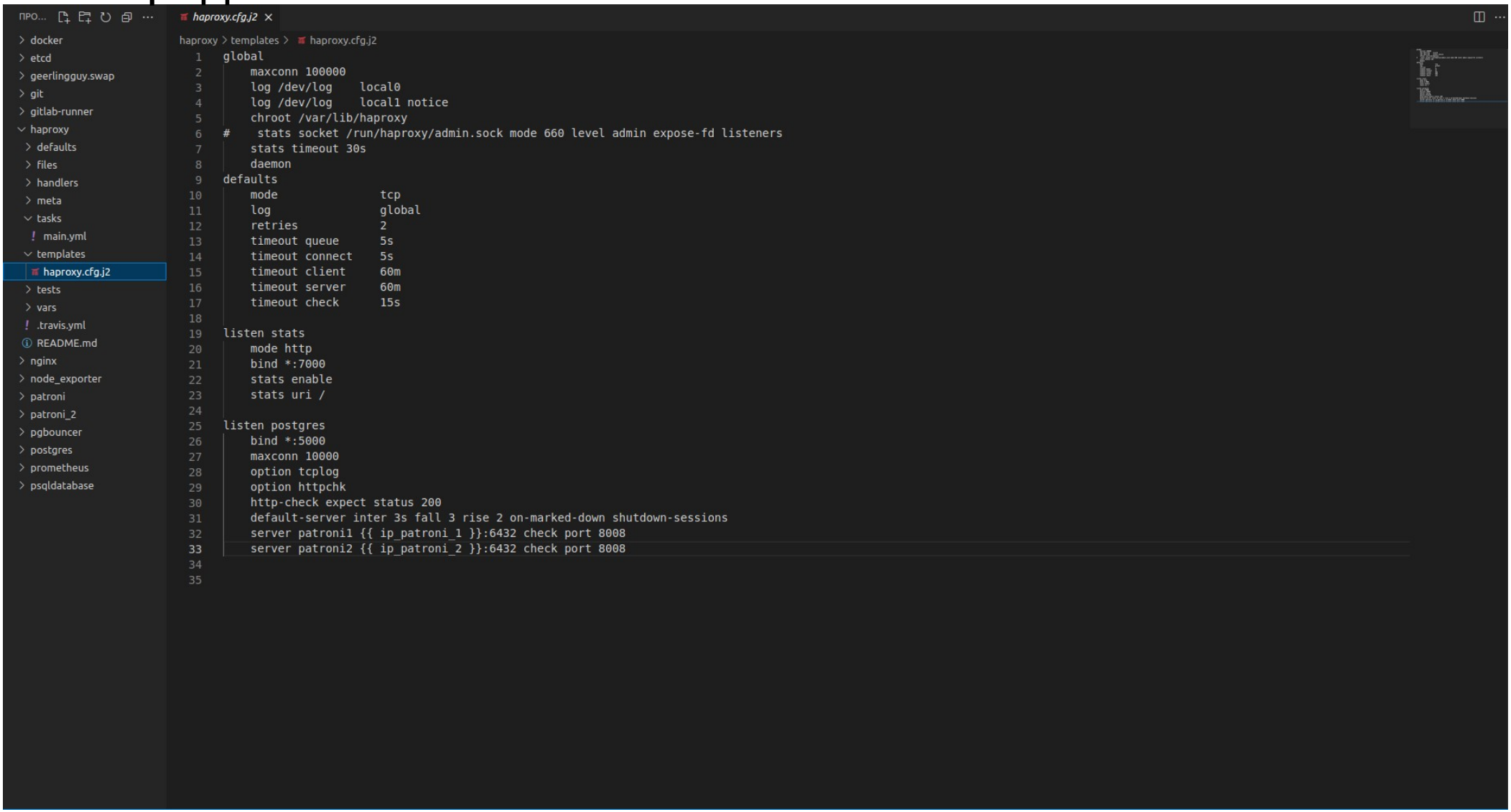
В директории handlers в файле main.yml прописал рестарт etcd по notify: restart etcd



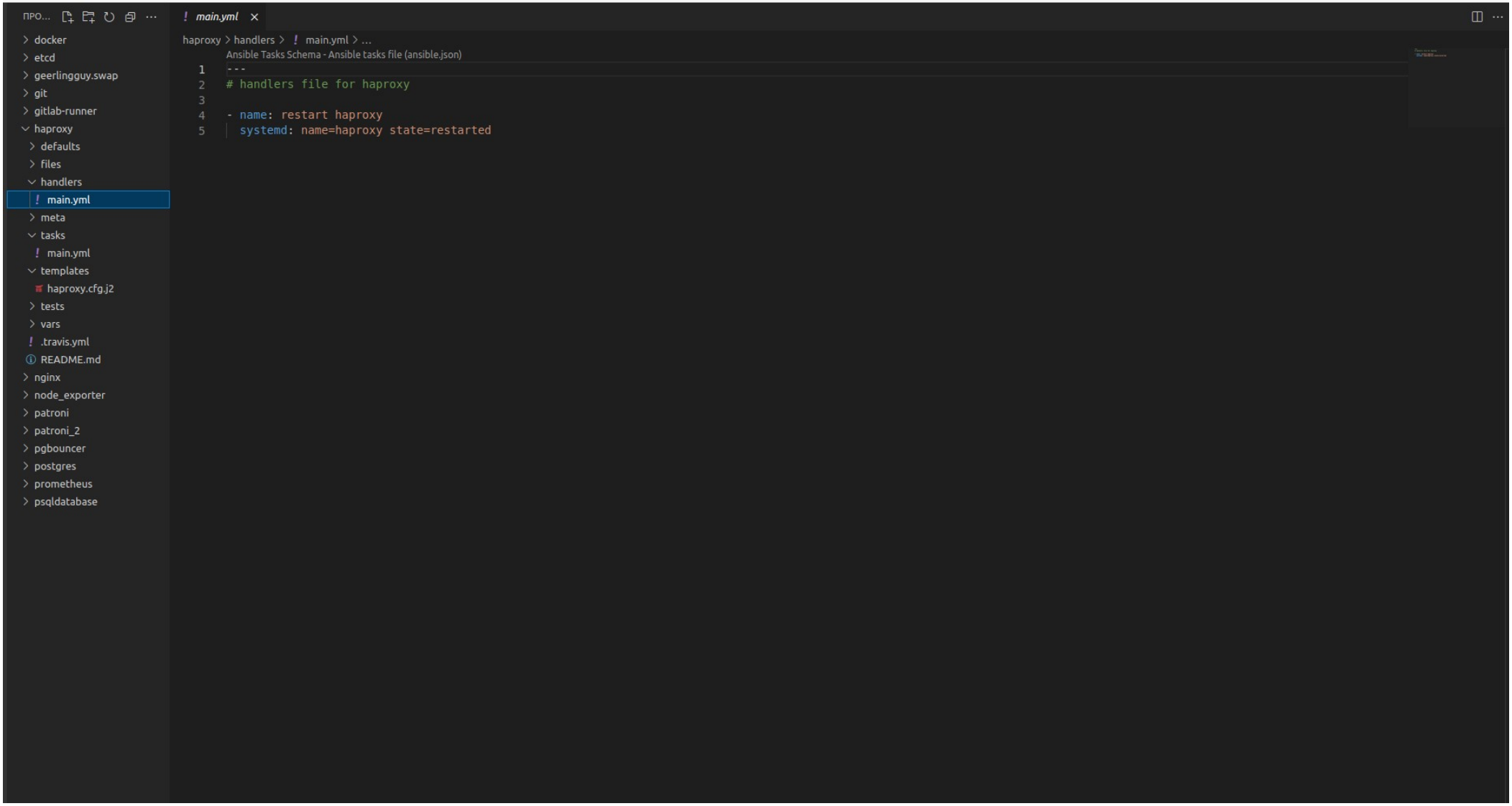
Инициализировал роль haproxy
В директории task в файле main.yaml прописал установку haproxy и создание конфига из шаблона j2 с сохранением оригинала. После создания конфига происходит рестарт.



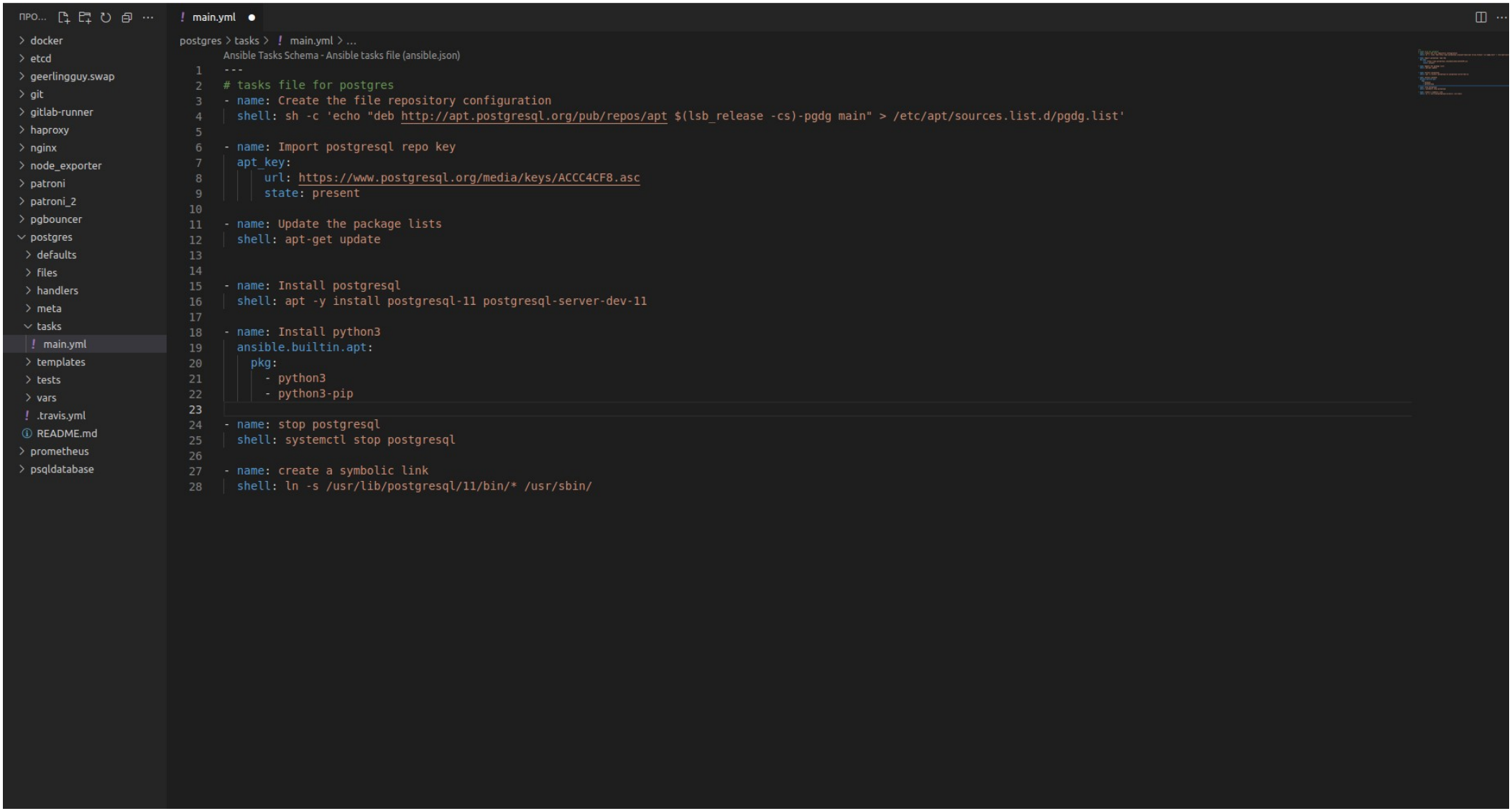
В директории templates прописал шаблон для конфиг файла haproxy.
Использую переменную ip_patroni 1 и 2.
Её определил в inventori.



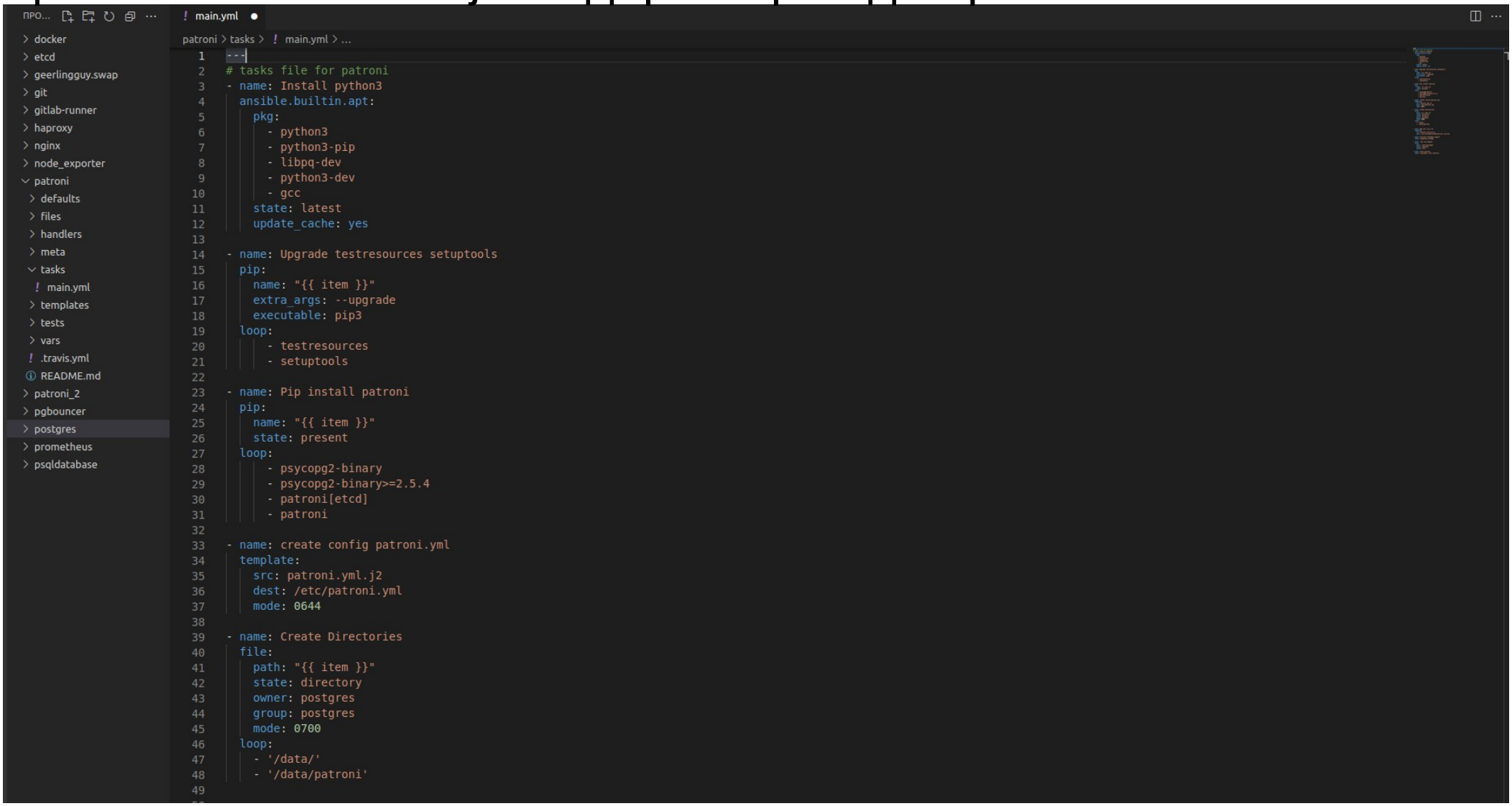
В директории handlers в файле main.yml прописал рестарт etcd по notify: restart haproxy



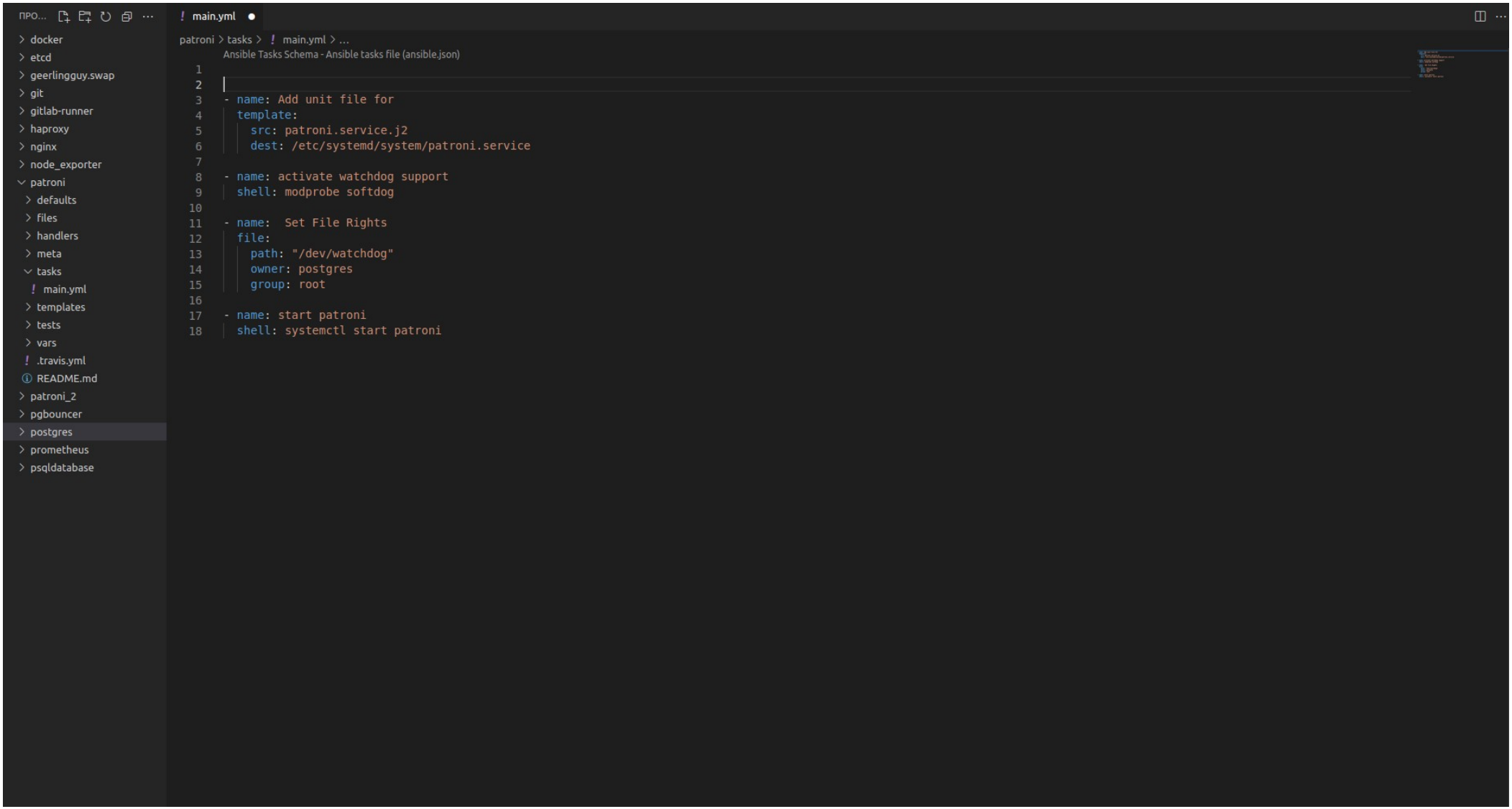
Инициализировал роль postgres
В директории task в файле main.yaml прописал установку postgres и создание symbolic link и остановка его.



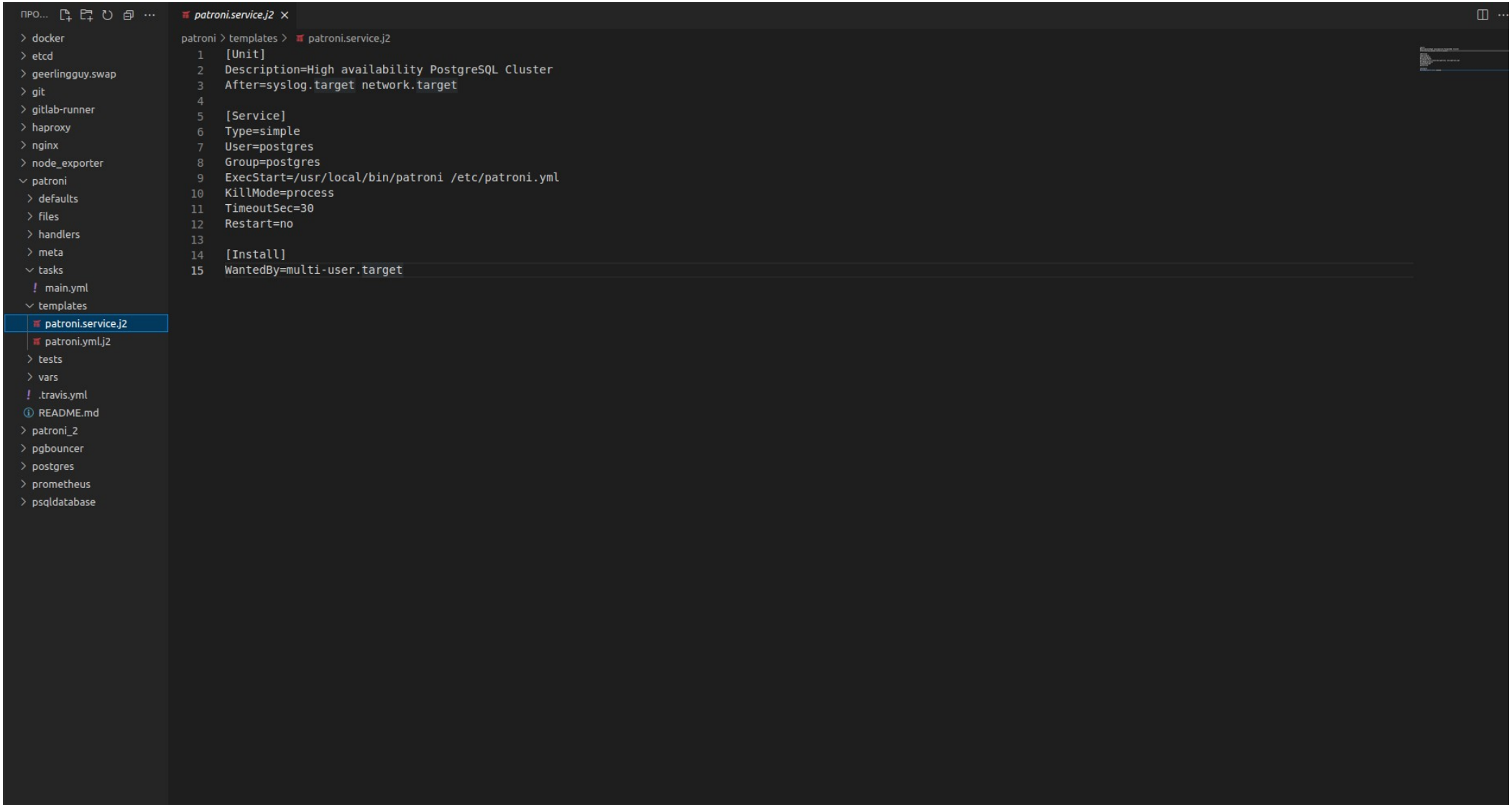
Инициализировал роль patroni 1.
В директории task в файле main.yaml прописал установку всего необходимого для работы patroni. Создание конфиг файла из шаблона j2. И директории для patroni.



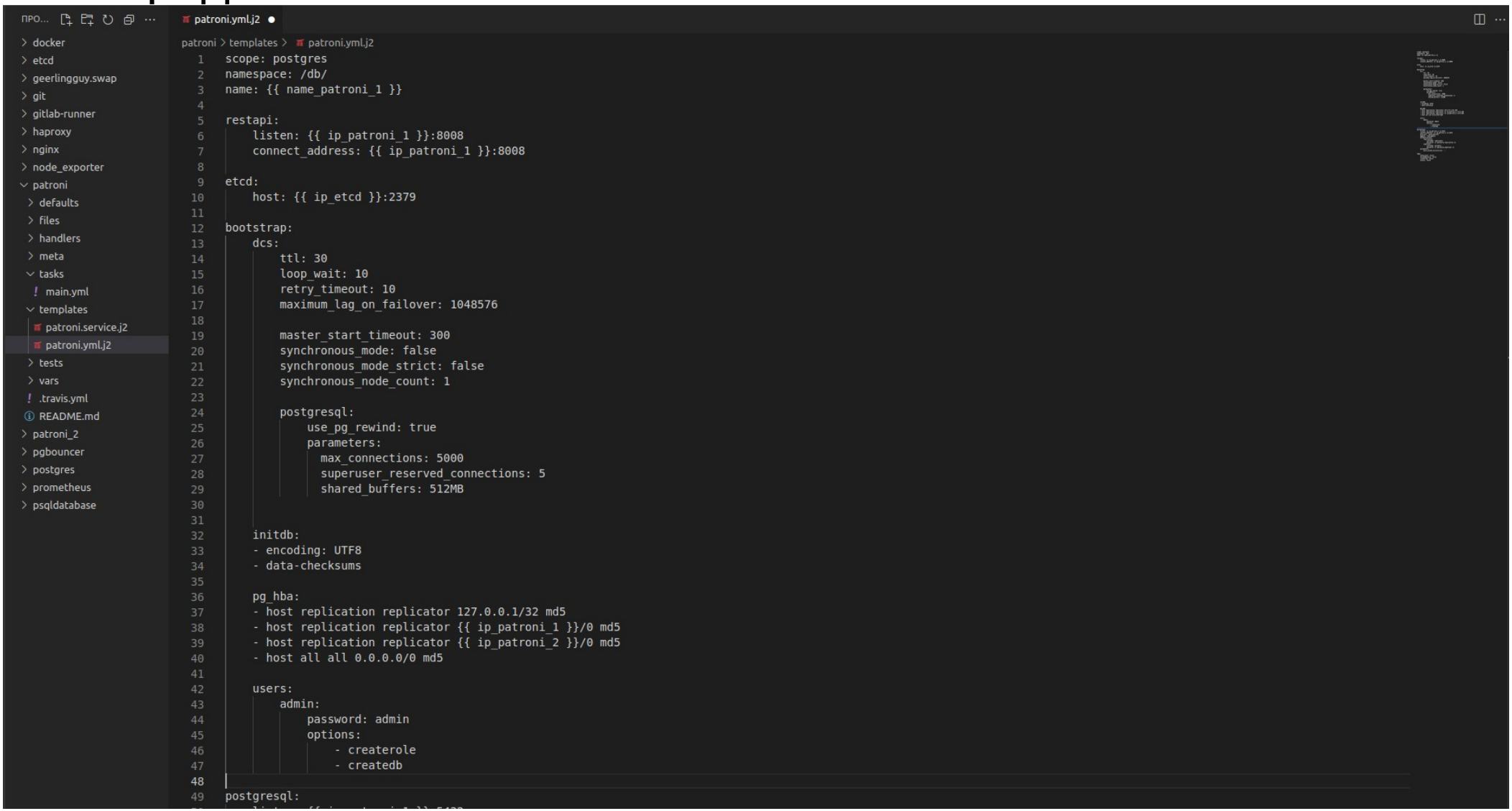
Прописал создание юнит файла для patroni из шаблона j2. Запуск patroni.



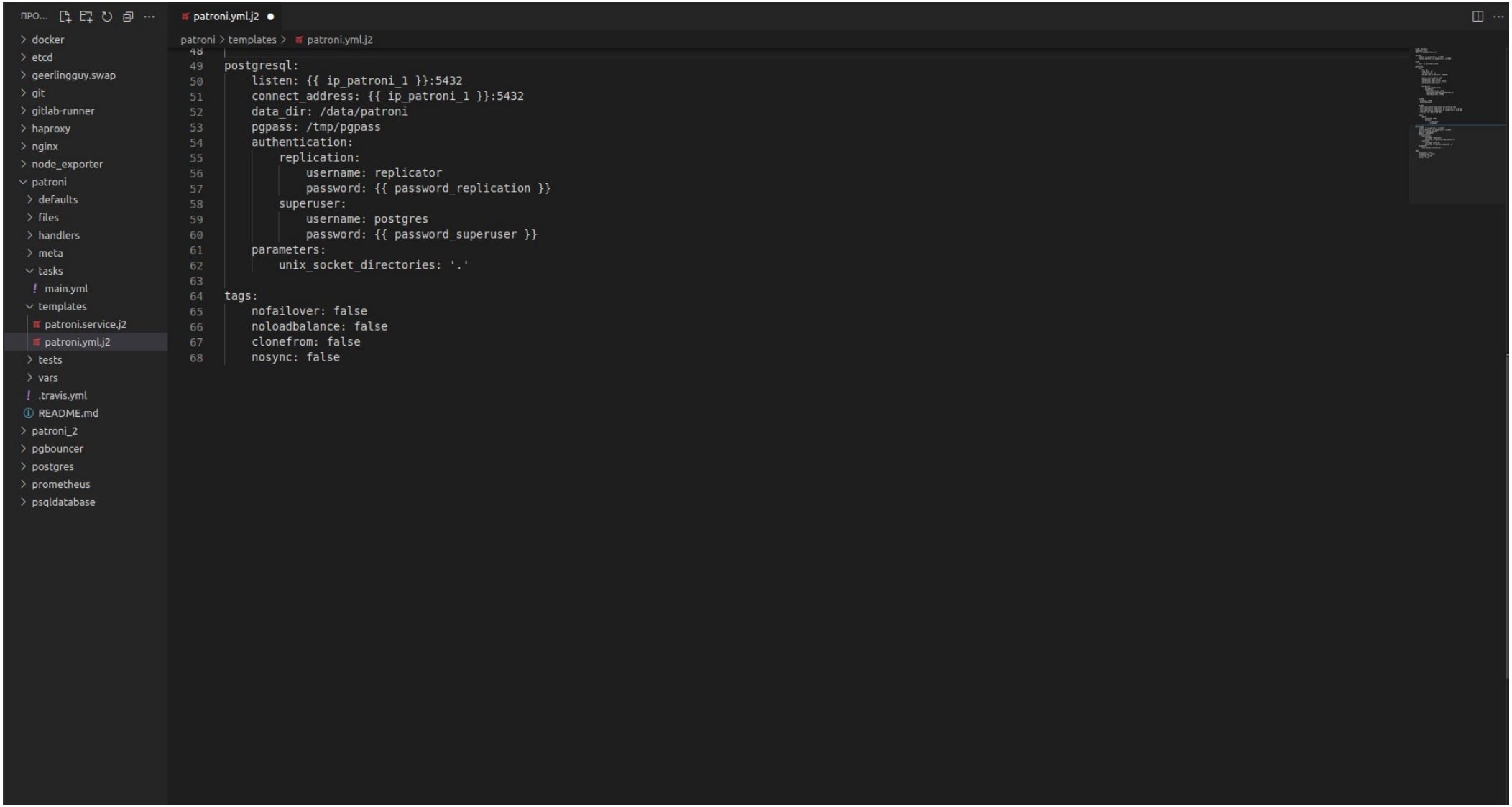
В директории templates прописал шаблон для юнит файл patroni.



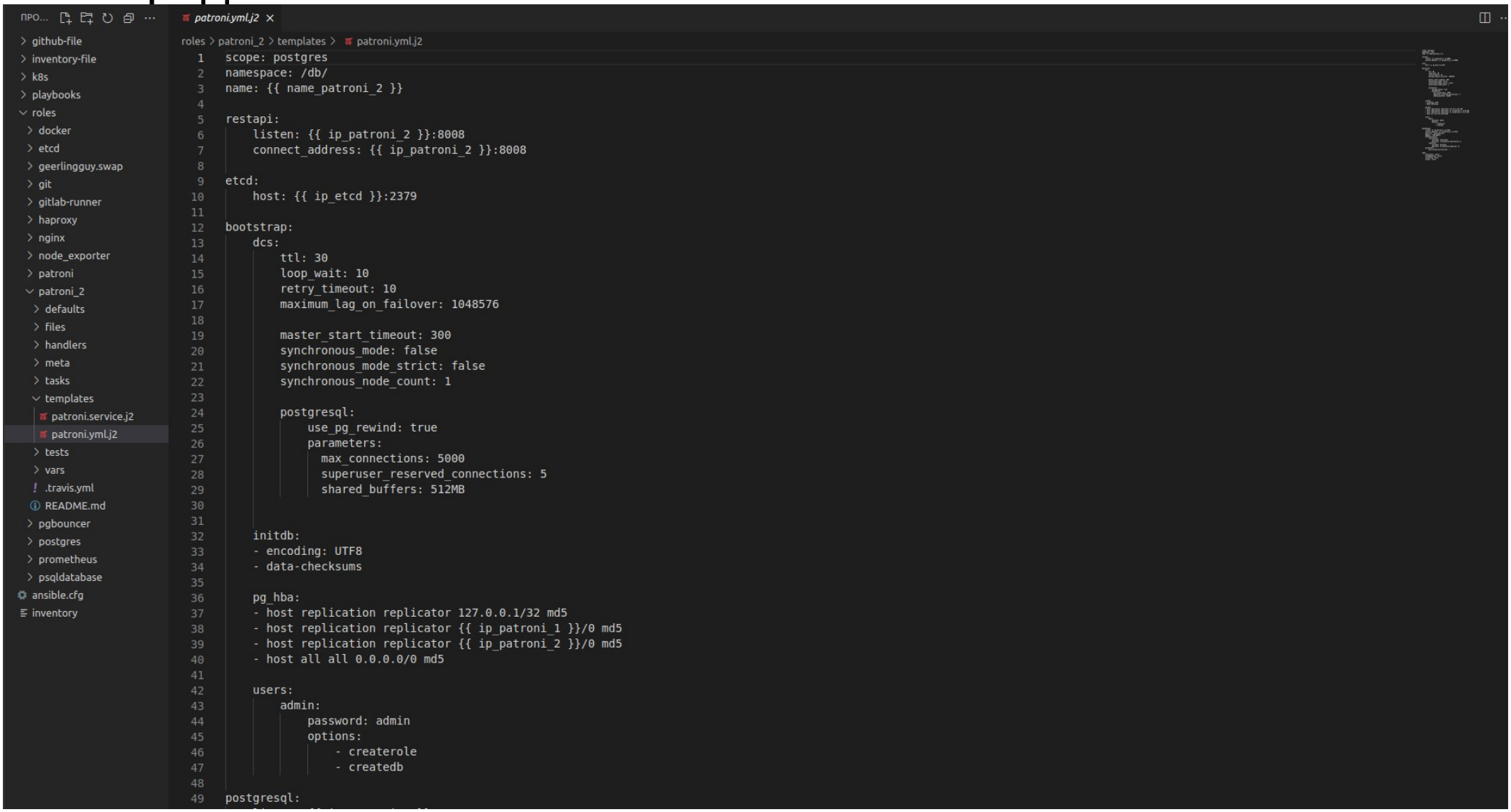
В директории templates прописал шаблон для конфиг файл patroni.
Использую переменную ip_etcd, ip_patroni 1, 2)
Их определил в inventori.



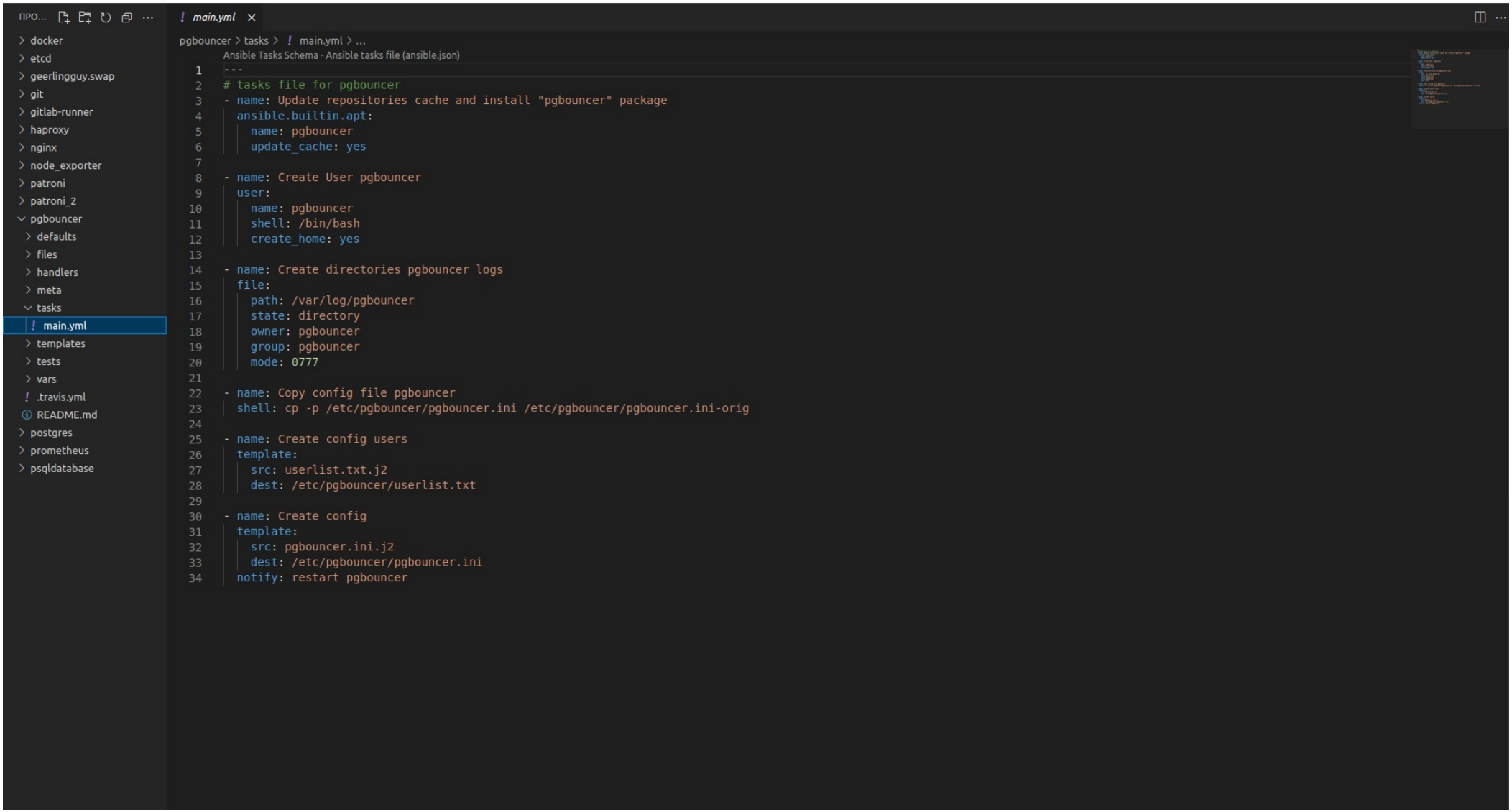
Так же использую переменные:
password_replication
password_superuser.
Они так же прописаны в inventori.



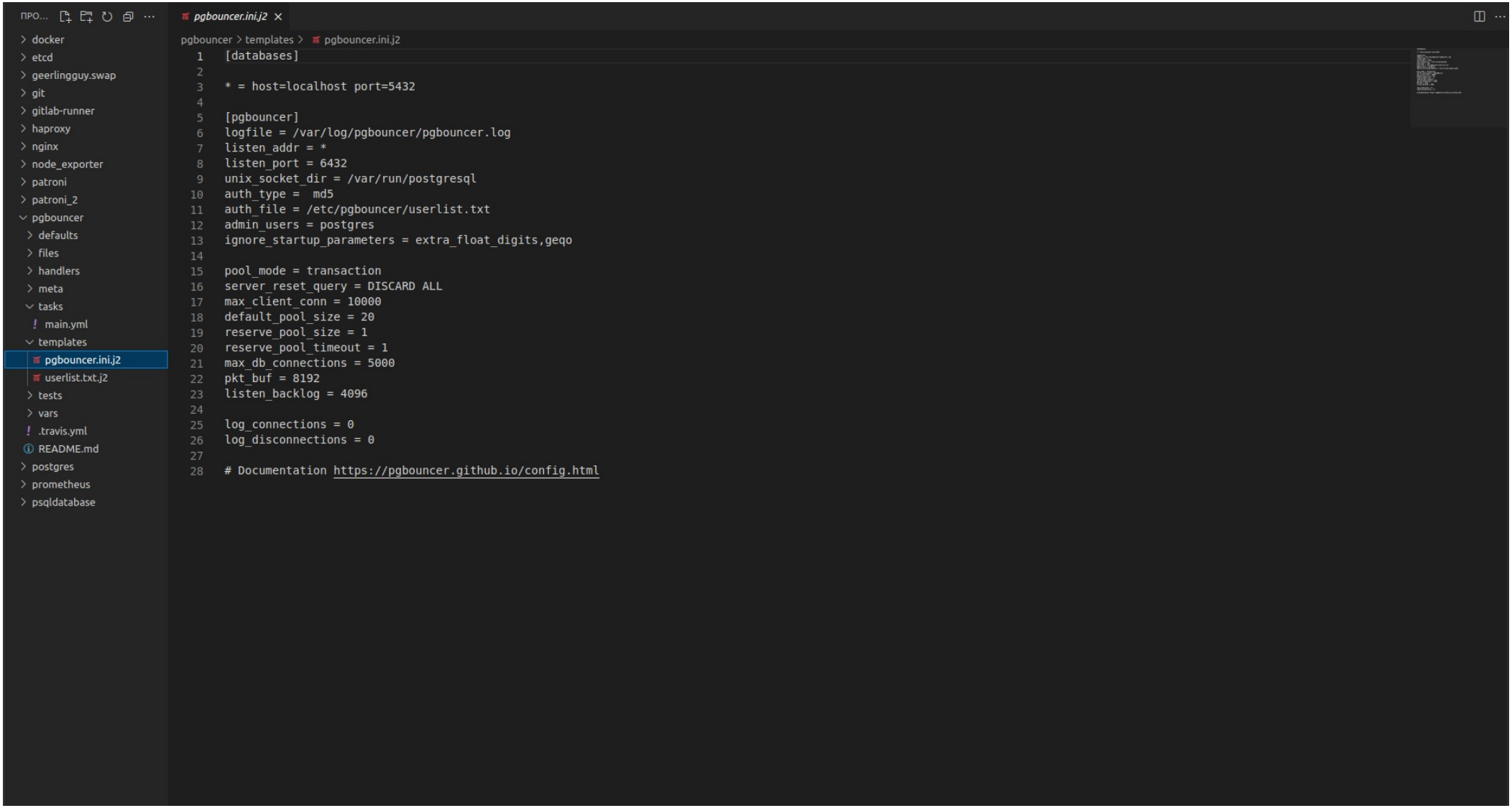
Инициализировал роль patroni 2. Отличие этой роли от роли patroni 1 только в шаблоне для конфиг файл patroni.
Использую переменную ip_etcd, ip_patroni 1, 2
Их определил в inventori.



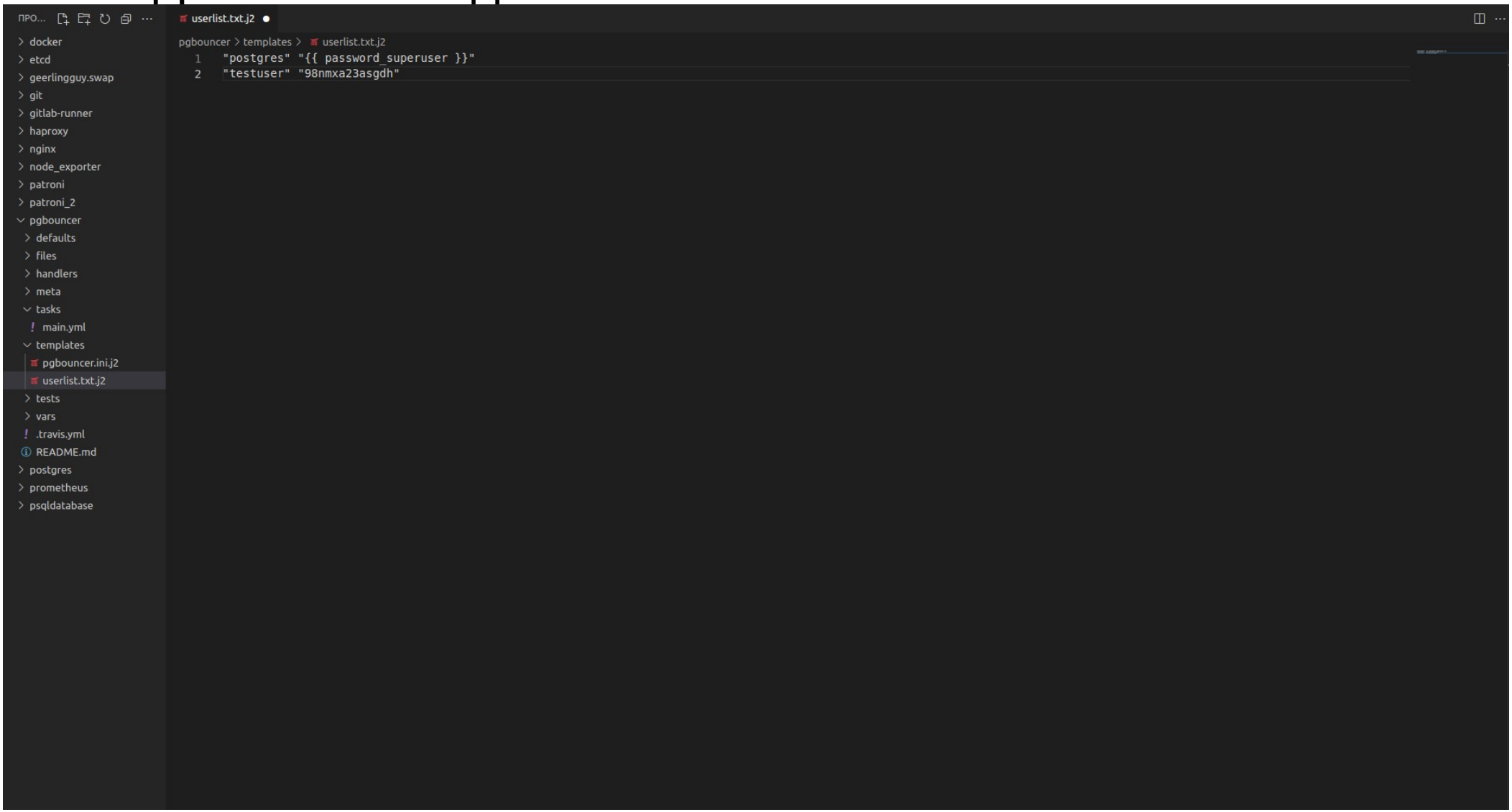
Инициализировал роль pgbouncer
В директории task в файле main.yaml прописал установку pgbouncer и создание для него юзера, директории логов и конфиг файлов из шаблонов j2.



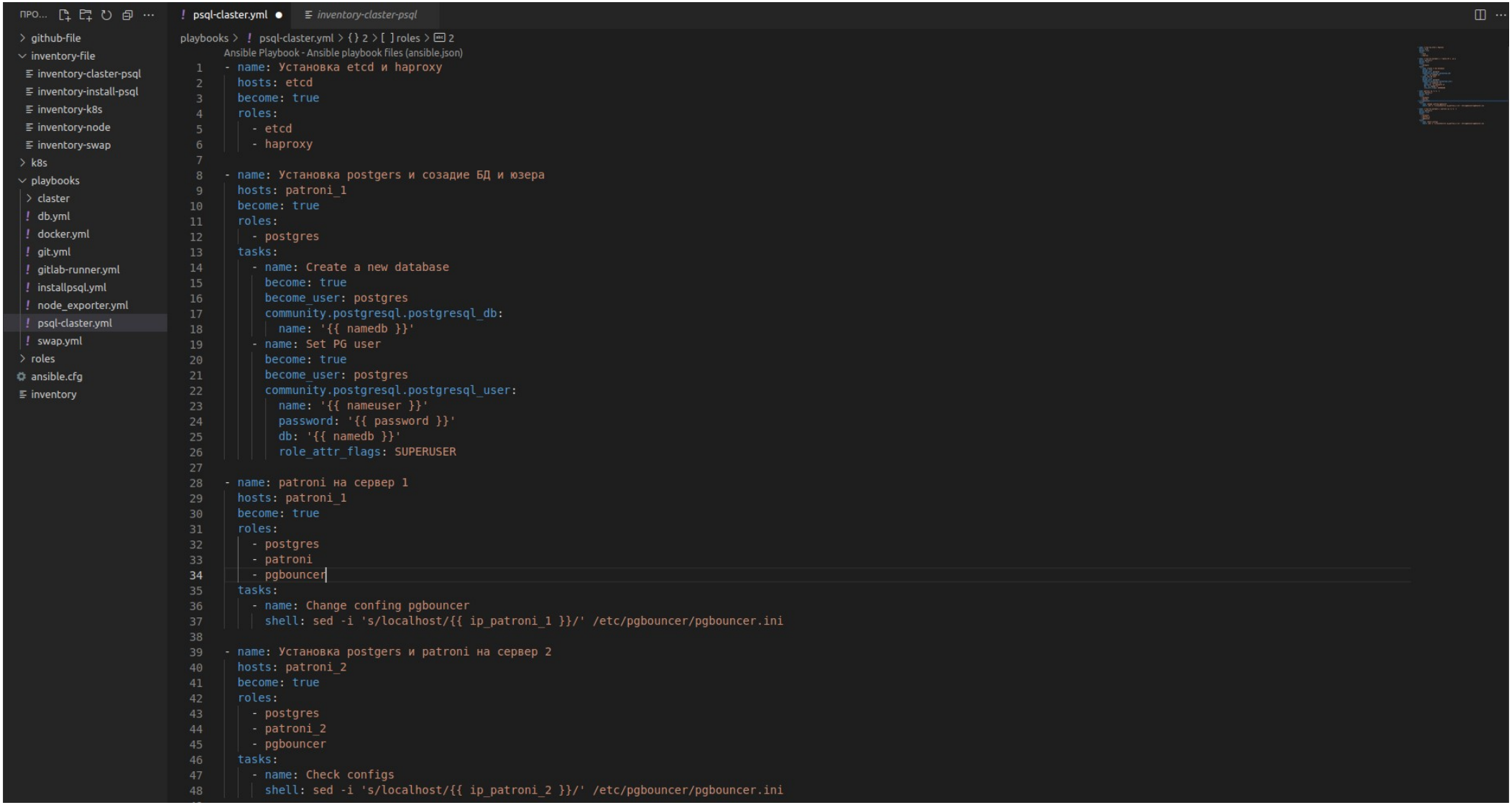
В директории templates прописал шаблон для конфиг файл pgbouncer.



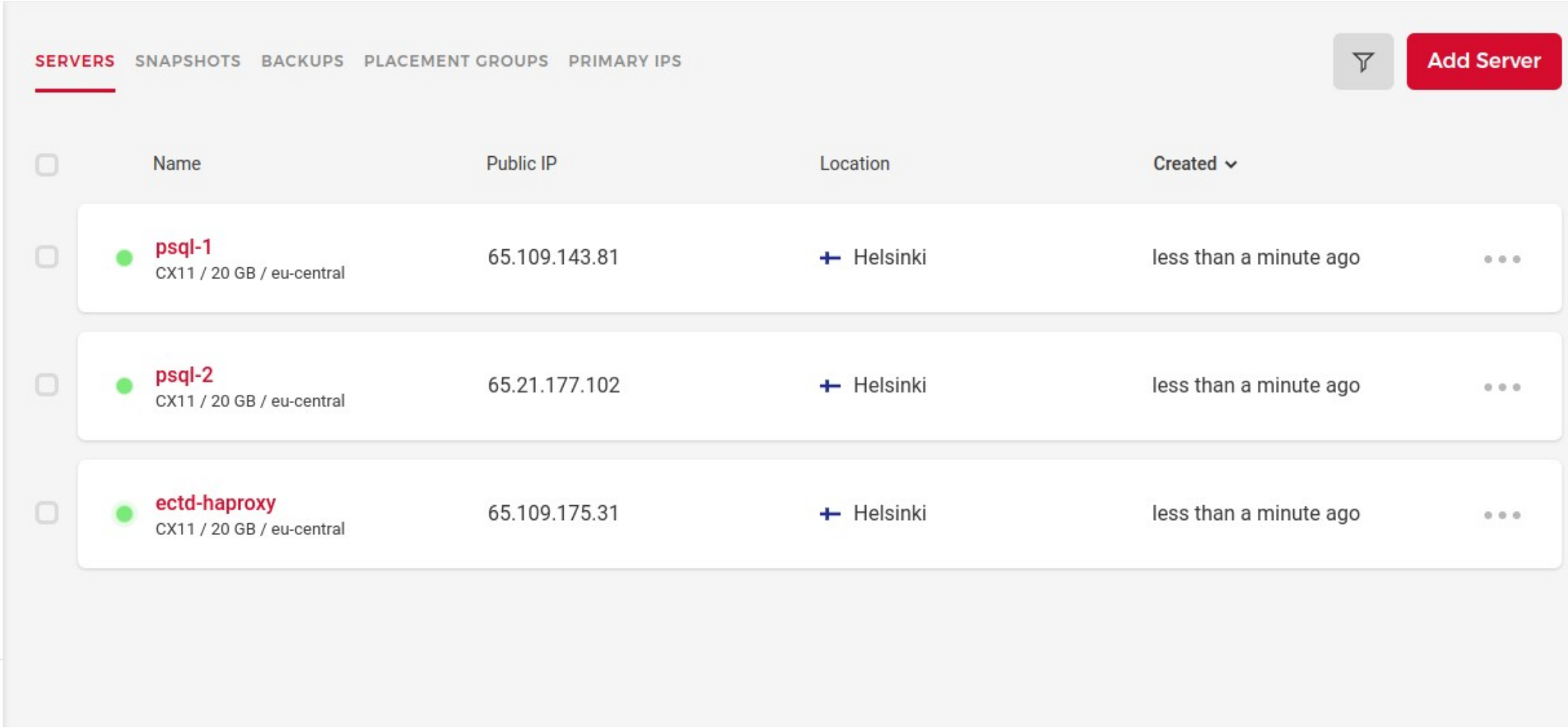
Так же создал шаблон для списка юзеров.
В нем используются переменная password_superuser.
Указал тестового юзера и пароль для него. Их можно указать в зашифрованном виде.



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



Развернул три сервера



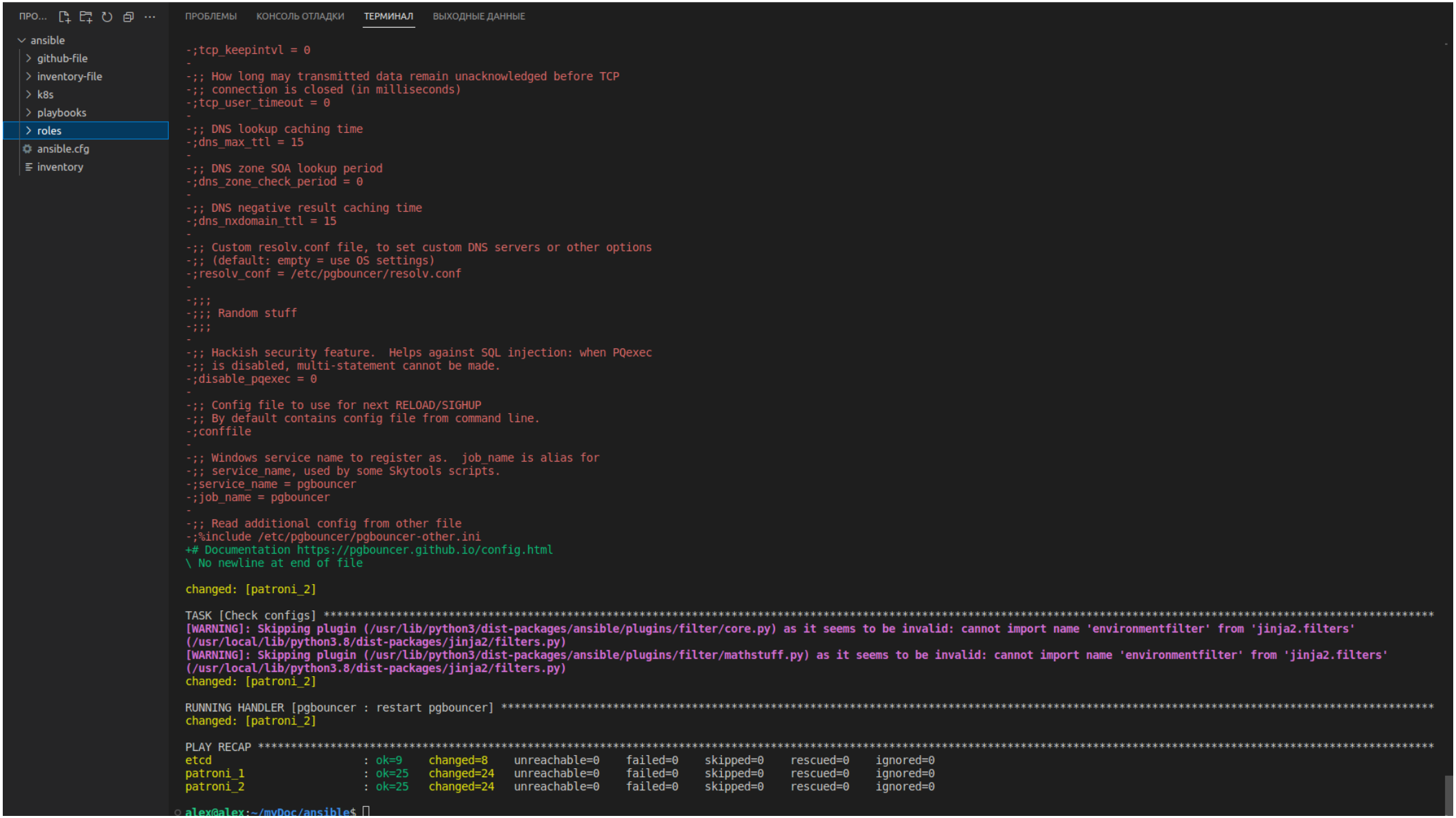
Создал inventory файл прописал в нем переменные

inventory-claster-psql

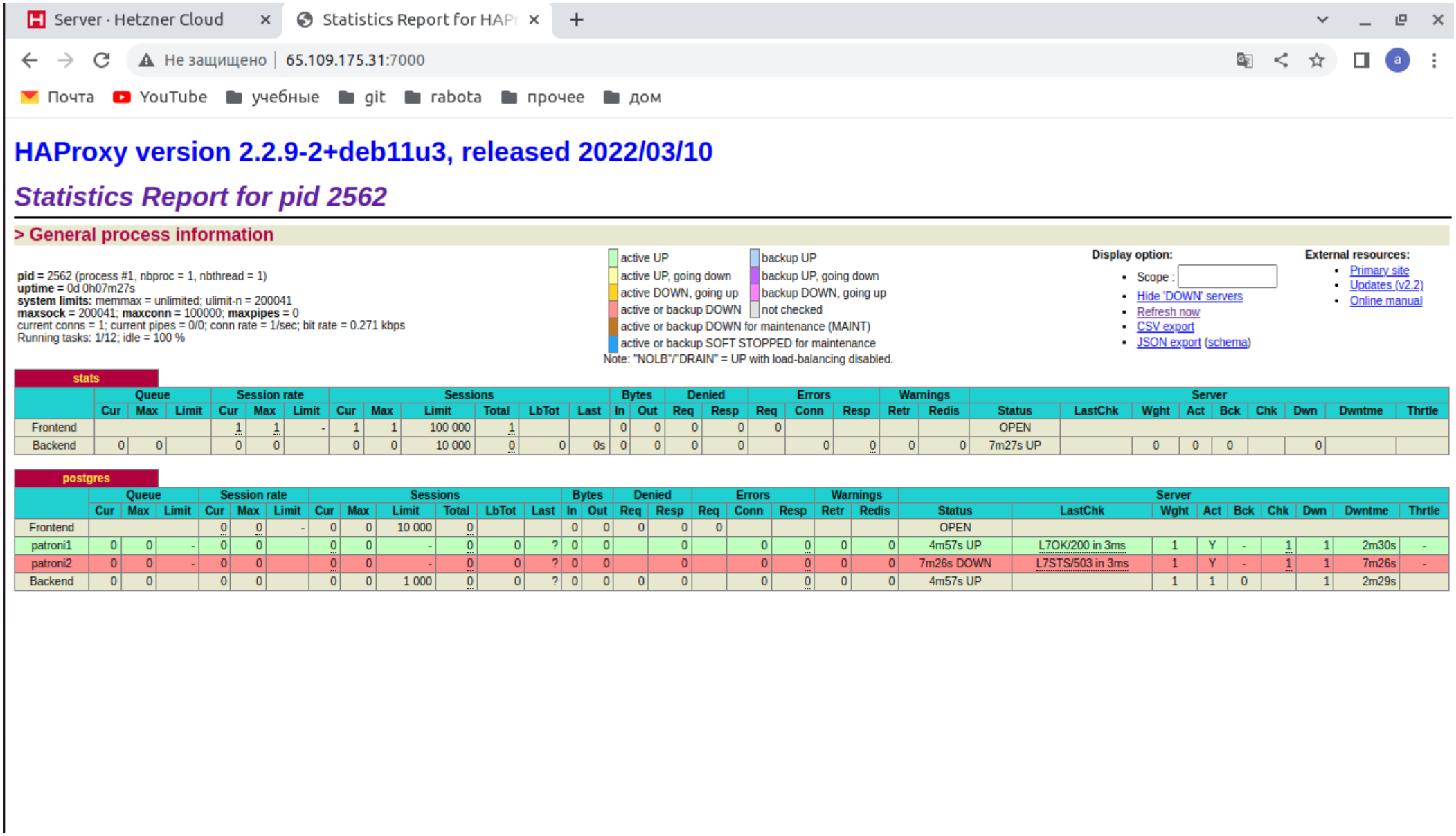
ansible > inventory-file > inventory-claster-psql

```
[Claster]
etcd ansible_host=65.109.175.31
patroni_1 ansible_host=65.109.143.81
patroni_2 ansible_host=65.21.177.102

[Claster:vars]
#####переменные etcd
    ip_etcd=65.109.175.31
#####перемнные haproxy
    ip_patroni_1=65.109.143.81
    ip_patroni_2=65.21.177.102
#######переменные для patroni и postgres
     nameuser=testuser
     namedb=testdb
     password=98nmxa23asgdh
     name_patroni_1=patroni_1
     name_patroni_2=patroni_2
     password_replication=QWEasdzxc
     password_superuser=QWEasdzxc
###для pgbouncer
#пользователь postgres пароль QWEasdzxc

```

Запустил плэйбук



По ip адресу сервера с haproxy на порту 7000 есть страница со статистикой.
Видим что patroni 1 сейчас master



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 haproxy 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		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	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 статисткой 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	