

Создание и тестирование высоконагруженного отказоустойчивого кластера PostgreSQL на базе Patroni

PostgreSQL для администраторов баз данных и разработчиков



**Меня хорошо видно
& слышно?**



Защита проекта

Тема: Создание и тестирование высоконагруженного отказоустойчивого кластера PostgreSQL на базе Patroni



Ален Дюрон

Администратор баз данных Oracle – Компания ППР
Работаю в области баз данных (в основном Oracle) 20 лет.

Образование: Бакалавра компьютерных наук, Магистр по искусственному интеллекту

План защиты

Цель и задачи проекта

Какие технологии использовались

Что получилось

Выводы

Вопросы и рекомендации

Цель и задачи проекта

Цель проекта: Создать высокодоступный кластер PostgreSQL, развёртывание и обслуживание которого будут автоматизированы модулями Patroni и etcd, и тестировать отказоустойчивость кластера в рамках планированного и не планированного переключения роли.

1. Создание виртуальной кластеризованной среды Ubuntu с распределённым хранилищем конфигурации *etcd*
2. Настройка модуля управления *Patroni*
3. Тестирование переключения и отказоустойчивости кластера
4. Тестирование отказоустойчивости и переключение клиентов с помощью *HAProxy*.

Какие технологии использовались

Компонент	Название	Версия	Комментарии
Слой виртуализации	Oracle VM VirtualBox	7.0.22 r165102	
ОС кластера	Ubuntu	24.04.3 LTS	
DCS (распределённое хранилище конфигурации)	etcd	3.4.30	Repository (Последняя версия недоступна)
Слой отказоустойчивости Postgres	Patroni		
База данных	Postgres	18	
LoadBalancing	HAProxy		



Что получилось – [etcd]

```
aduron@ubt-pg-aduron-dbnode2:~$ etcl endpoint status -w table
```

ENDPOINT	ID	VERSION	DB SIZE	IS LEADER	IS LEARNER	RAFT TERM
https://ubt-pg-aduron-etcd1:2379	8e9e05c52164694d	3.4.30	20 kB	true	false	8
https://ubt-pg-aduron-etcd2:2379	8e9e05c52164694d	3.4.30	20 kB	true	false	11
https://ubt-pg-aduron-etcd3:2379	8e9e05c52164694d	3.4.30	20 kB	true	false	15

⚠ Warning

- Быть внимательным при использовании клонов ВМ
- Осторожно с использованием генерации сертификатов
- Аккуратно проверить состояние и ID участников кластера

💡 Tip

- Аккуратно выбрать способ установки (версию)
- Удаление папки *members* после клонирования ВМ
- Создать *alias* для работы с *etcdctl* и сертификатами

```
aduron@ubt-pg-aduron-dbnode1:~$ etcl endpoint status -w table
```

ENDPOINT	ID	VERSION	DB SIZE	IS LEADER	IS LEARNER	RAFT TERM
https://ubt-pg-aduron-etcd1:2379	75ae20a713d9d171	3.4.30	20 kB	false	false	2
https://ubt-pg-aduron-etcd2:2379	6e58acf468311e04	3.4.30	20 kB	true	false	2
https://ubt-pg-aduron-etcd3:2379	9444b20a64655d13	3.4.30	20 kB	false	false	2

Что получилось – [HAProxy]

```
aduron@ubt-pg-aduron-cluster3:~$ sudo vi /etc/haproxy/haproxy.cfg
```

```
...
```

```
frontend postgres_ssl
```

```
bind *:5432
```

```
default_backend postgres_backend
```

```
...
```

```
backend postgres_backend
```

```
option httpchk OPTIONS /master
```

```
http-check expect status 200
```

```
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
```

```
server ubt-pg-aduron-dbnode1 192.168.56.10:5432 check check-ssl verify none port 8008 crt /etc/haproxy/combined.crt ca-file /etc/haproxy/ca.crt
```

```
server ubt-pg-aduron-dbnode2 192.168.56.20:5432 check check-ssl verify none port 8008 crt /etc/haproxy/combined.crt ca-file /etc/haproxy/ca.crt
```

```
...
```

💡 Tip

- *HAProxy* пользуется таким же API, как *patronictl*
- Если *Patroni* настроен с сертификатами, их нужно тоже использовать в рамках настройки *haproxy*



Что получилось – [HAProxy]

Statistics Report for HAProxy

Небезопасно 192.168.56.30:8404/stats

HAProxy version 2.8.16-0ubuntu0.24.04.1, released 2025/12/03

Statistics Report for pid 2617

> General process information

pid = 2617 (process #1, nbproc = 1, nbthread = 2)
uptime = 0d 0h00m21s; warnings = 1
system limits: memmax = unlimited; ulimit-n = 8248
maxsock = 8248; maxconn = 4096; reached = 0; maxpipes = 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec; bit rate = 19.305 kbps
Running tasks: 0/19; 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

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

Display option: Scope:

External resources: [Primary site](#), [Updates \(v2.8\)](#), [Online manual](#)

[Hide 'DOWN' servers](#), [Disable refresh](#), [Refresh now](#), [CSV export](#), [JSON export \(schema\)](#)

postgres_ssl

	Queue			Session rate			Sessions				Bytes	Denied	Errors	Warnings	Server															
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	LbTot					Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk
Frontend	0	0	-	0	0	0	0	0	4 000	0	0	0	0	0	0	0	0	0	0	0	0	0	OPEN							

postgres_backend

	Queue			Session rate			Sessions				Bytes	Denied	Errors	Warnings	Server															
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	LbTot					Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk
ubt-pg-aduron-dbnode1	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22s UP	L7OK/200 in 5ms	1/1	Y	-	0	0	0s	-
ubt-pg-aduron-dbnode2	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20s DOWN	L7STS/503 in 11ms	1/1	Y	-	1	1	20s	-
Backend	0	0	0	0	0	0	0	0	410	0	0	0	0	0	0	0	0	0	0	0	0	22s UP		1/1	1	0	0	0	0s	

stats

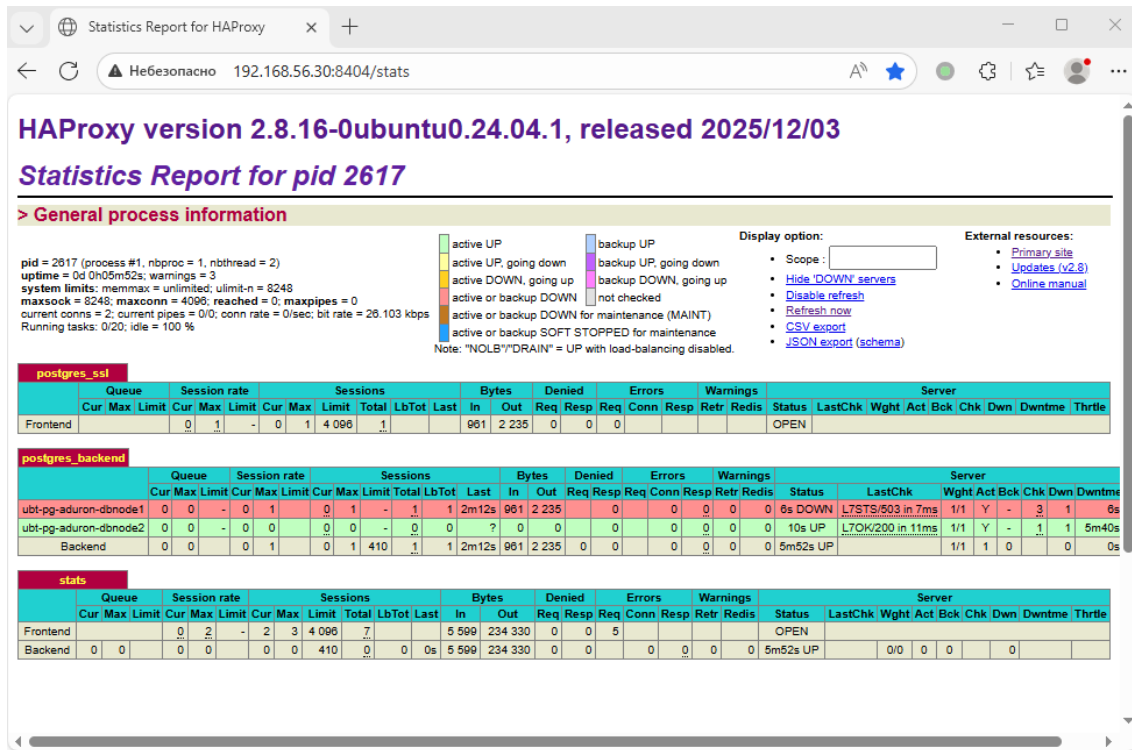
	Queue			Session rate			Sessions				Bytes	Denied	Errors	Warnings	Server															
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	LbTot					Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk
Frontend	1	1	-	1	1	4 000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OPEN								
Backend	0	0	0	0	0	0	0	0	410	0	0	0	0	0	0	0	0	0	0	0	0	22s UP		0/0	0	0	0	0	0	

```
aduron@ubt-pg-aduron-cluster3:/etc/haproxy$ psql  
'postgresql://postgres:Oracle123@localhost:5432/postgres'  
psql (18.1 (Ubuntu 18.1-1.pgdg24.04+2))  
SSL connection (protocol: TLSv1.3,  
cipher: TLS_AES_256_GCM_SHA384,  
compression: off, ALPN: postgresql)  
Type "help" for help.
```

```
postgres=# SELECT inet_server_addr() as  
ip, CASE WHEN pg_is_in_recovery() THEN  
'Replica' ELSE 'Primary' END as role;  
          ip          | role  
-----  
192.168.56.10 | Primary  
(1 row)
```



Что получилось – [HAProxy] - Switchover



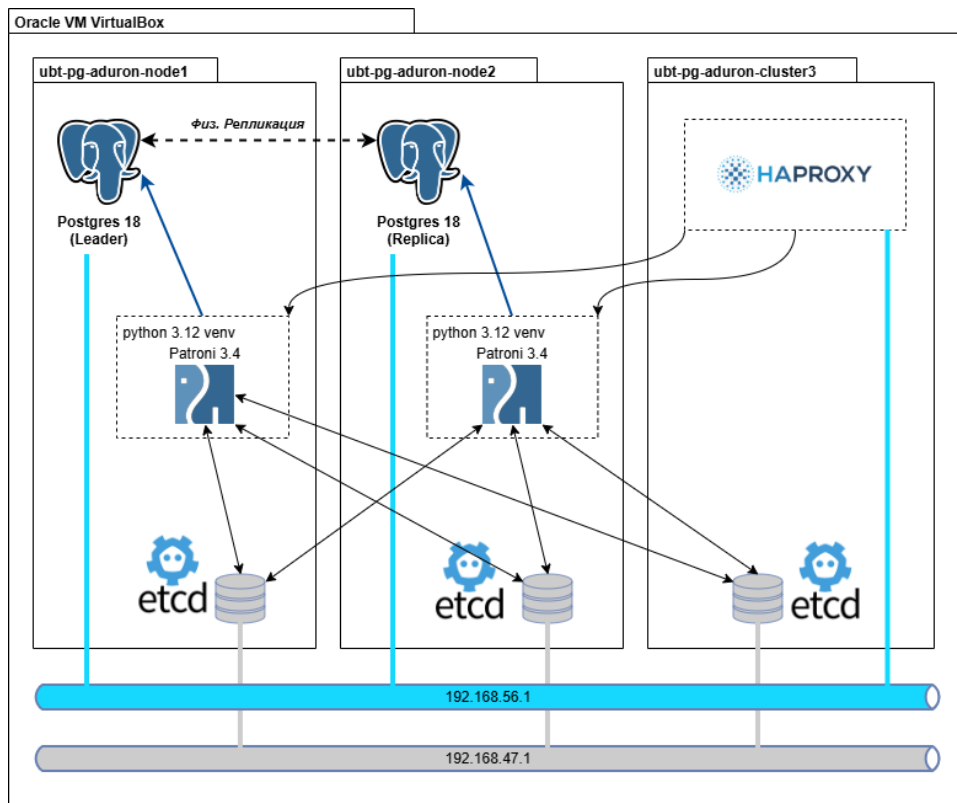
```
postgres=# SELECT inet_server_addr() as  
ip, CASE WHEN pg_is_in_recovery() THEN  
'Replica' ELSE 'Primary' END as role;
```

SSL error: unexpected eof while reading
The connection to the server was lost.
Attempting reset: Succeeded.
SSL connection (protocol: TLSv1.3,
cipher: TLS_AES_256_GCM_SHA384,
compression: off, ALPN: postgresql)

```
postgres=# SELECT inet_server_addr() as  
ip, CASE WHEN pg_is_in_recovery() THEN  
'Replica' ELSE 'Primary' END as role;  
-----+-----  
192.168.56.20 | Primary  
(1 row)
```



Что получилось



Выводы

- В целом получилось достичь все цели, которые были поставлены (кластер Postgres хорошо управляется модулем Patroni, настройки которого применяются с помощью etcd. Switchover и Failover хорошо выполняются. Подключения через HAProxy)
- Проведение тестирования (failover / HAProxy) оказалось легко, однако возникли трудности с управлением сертификатами и с пониманием порядка применения настроек Patroni (listen_addresses). *Некоторые проблемы возникли из за выбранного способа установления (клонирование)*
- Проект занял в целом 2 недели (по вечерам) и оказался очень полезным (10)
- Среди остальных вопросов: возникли ли у однопользователей такие же трудности как у меня (особенно, в связи с неопределенными в документации настройками – например **enable-grpc-gateway...**)
- Развивать эту тему дальше планирую на работе, так как мы активно разворачиваем кластеры на базе Patroni

Ссылка к проекту на GitHub:

[PostgreSQL-OTUS-DZ / Проектная работа](#)



Демо



Вопросы и рекомендации



если есть вопросы



если вопросов нет



Спасибо за внимание!

oTus

