Using Patroni for High Availability Deployments

João Vitor Foltran e Samuel Molling

QUEM SOMOS

25 anos
Blumenau, SC
Especialista PostgreSQL @ TOTVS

linkedin.com/in/joaofoltran

joao@foltrandba.com

- EnterpriseDB, iFood, GoldenGateBR
- Maluco por card games (FAB, MTG)
- Amo automação

25 anos
Campo Bom, RS
Tech Lead DBRE @ TAG IMF

linkedin.com/in/samuelmolling

samuelmolling@gmail.com

- GetNet, Stone e iFood
- Amante dos esportes
- Cloud, OpenSource e muito código

Agenda

- O que é o Patroni?
- A long time ago...
- Benefícios
- Por quê Patroni?
- Componentes
- O que é o DCS e quais as soluções disponíveis?
- O que é o algoritmo Raft?
- Como funciona o HAProxy?
- Configurações Patroni e PostgreSQL
- Create replica
- Tags
- Comandos para administração
- Exemplos de Deploy
- Demonstração

O que é o Patroni?

O Patroni é uma ferramenta de código aberto desenvolvida em Python.
Ele é utilizado para gerenciar e fornecer alta disponibilidade para clusters PostgreSQL.

Seu objetivo principal é garantir que o serviço do PostgreSQL esteja sempre disponível, mesmo em caso de falha de um nó ou de outras situações adversas.

A long time ago...

- Balanceamento de carga manual
- Integração com ferramentas de orquestração
- Configuração e monitoramento manuais
- Viradas de DNS manuais
- Recriar o antigo primário
- Failover manual

Benefícios

- **Gerenciamento de grupo:** gerenciar um grupo de servidores de forma fácil.
- **Failover:** transferir automaticamente a carga de um node para outro em caso de falha
- **Escalabilidade:** escalar seu cluster adicionando novos nodes a qualquer momento.
- **Disaster Recovery:** recuperar seu cluster de um desastre através de um servidor de backup.
- **Confiabilidade:** mantém seu cluster sempre disponível mesmo em caso de falhas.
- **API:** disponibiliza APIs para gerenciamento do cluster e health checks.
- **Facilidade de uso**
- **Código aberto**
- **Callbacks:** chamar scripts pré-configurados após determinadas ações (on start, on restart, on change role, etc).

Por quê Patroni?

- **Repmgr**
  - Não automatiza o recovery de nodes que falharam.
  - Não possibilita gerenciamento de um outro node de qualquer node do cluster.
  - Não trabalha com Quorum.

- **PAF**
  - Não automatiza a inicialização e configuração do cluster.
  - Sem suporte a configurações com NAT
  - Não suporta pg_rewind automático

Componentes

- **DCS (Distributed Configuration Store)**
- **PostgreSQL (version >= 9.3)**
- **Patroni**
- **Balanceador (HAProxy, NLB)**

github.com/vitabaks/postgresql_cluster

ops.gitlab.net/gitlab-com/runbooks/-/tree/master/docs/patroni

github.com/zalando/postgres-operator

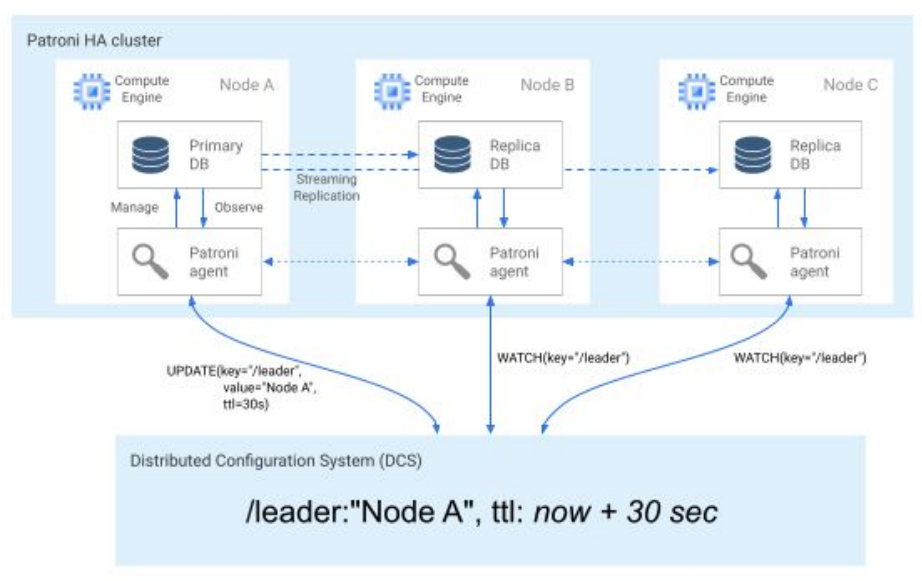
patroni.readthedocs.io/en/latest/README.html

github.com/zalando/patroni

O que é o DCS e quais as soluções disponíveis?

O DCS é responsável por armazenar informações de configuração (patronictl edit-config), estado dos nós, ajudar resolver a tarefa de eleições de líderes e detectar o particionamento de rede.

- etcd ou etcdv3
- consul
- zookeeper
- exhibitor
- kubernetes
- raft (deprecated)

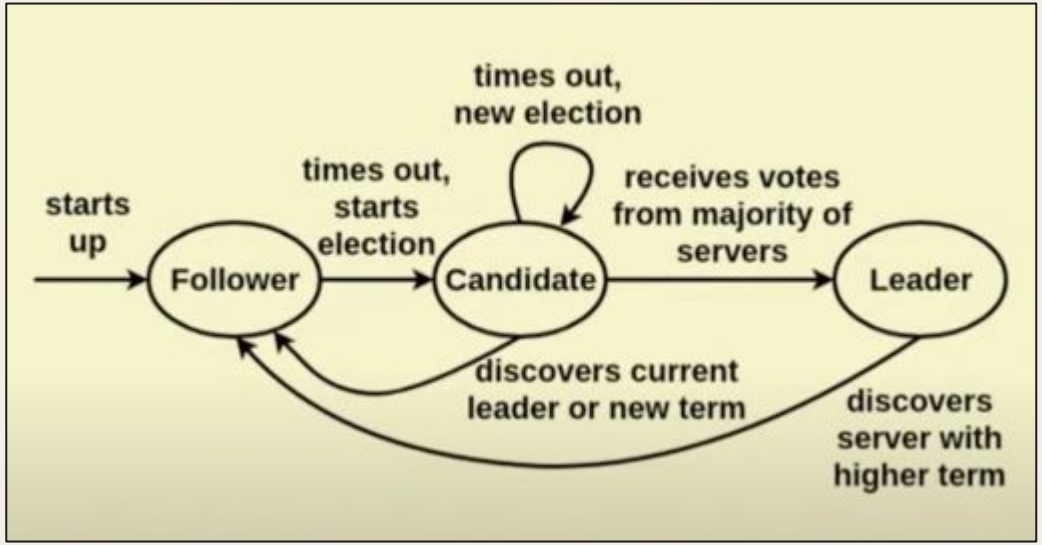


O que é algoritmo Raft?

Um servidor em um cluster raft pode ser um **líder** ou **seguidor**, porém em casos excepcionais onde ocorre uma falha com o líder, o node pode virar um **candidato**.

O **líder** é responsável por fazer a replicação de log para seus **seguidores**, ele os informa regularmente da sua existência através de uma mensagem que chamamos de **heartbeat**.

Cada seguidor possui um timeout para o recebimento desta mensagem, em caso de falha no recebimento, esse seguidor altera seu estado para **candidato** e começa uma nova eleição.



https://en.wikipedia.org/wiki/Raft_(algorithm)

Como funciona o HAProxy?

stats

	Queue			Session rate			Sessions				
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	
Frontend				1	1	-	1	1	2 000	1	
Backend	0	0		0	0		0	0	200	0	

primary

	Queue			Session rate			Sessions				
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTo
Frontend				0	0	-	0	0	2 000	0	
psql13n51	0	0	-	0	0		0	0	100	0	
psql13n52	0	0	-	0	0		0	0	100	0	
psql13n53	0	0	-	0	0		0	0	100	0	
Backend	0	0		0	0		0	0	200	0	

standbys

	Queue			Session rate			Sessions				
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbT
Frontend				0	0	-	0	0	2 000	0	
psql13n51	0	0	-	0	0		0	0	100	0	
psql13n52	0	0	-	0	0		0	0	100	0	
psql13n53	0	0	-	0	0		0	0	100	0	
Backend	0	0		0	0		0	0	200	0	

Configuração Patroni e PostgreSQL

- **Dinâmica - Podem ser definidas no DCS a qualquer momento, se não forem configurações de inicialização, será feito async com restart.**
- **Local (patroni.yml) - Tem precedência sobre as alterações dinâmicas e pode ser alterado utilizando um reload.**
- **Ambiente - Variáveis de ambiente**

Create replica

- pg_basebackup (default)
- pgBackrest
- wal_e
- Barman
- Custom script

```
postgresql:
    create_replica_methods:
        - pgbackrest
        - basebackup
    pgbackrest:
        command: /usr/bin/pgbackrest
--stanza=<scope> --delta restore
        keep_data: True
        no_params: True
    basebackup:
          max-rate: '100M'
```

```
postgresql:
    create_replica_methods:
        - wal_e
        - basebackup
    wal_e:
        command: patroni_wale_restore
        no_leader: 1
        envdir: {{WALE_ENV_DIR}}
        use_iam: 1
    basebackup:
          max-rate: '100M'
```

Tags

Tags são utilizadas para alterar o comportamento dos nodes que elas estão aplicadas.

- **nofailover:** node não participará em eleições ou virar o master
- **noloadbalance:** node não será incluso no load balancer
- **clonefrom:** o node será utilizado como base para criação de outros nodes
- **nosync:** node nunca será síncrono
- **replicatefrom:** utilizado para realizar replicação em cascata

Comandos para administração

- patronictl list
- patronictl show-config
- patronictl reload
- patronictl pause
- patronictl resume
- patronictl switchover
- patronictl failover
- patronictl reinit

patronictl list

```
postgres@patroni1:~$ patronictl list
+ Cluster: demo --------+---------+-----------+----+-----------+
| Member    | Host      | Role    | State     | TL | Lag in MB |
+-----------+-----------+---------+-----------+----+-----------+
| patroni1  | 172.24.0.5 | Leader  | running   |  1 |           |
| patroni2  | 172.24.0.3 | Replica | streaming |  1 |         0 |
| patroni3  | 172.24.0.4 | Replica | streaming |  1 |         0 |
+-----------+-----------+---------+-----------+----+-----------+
postgres@patroni1:~$
```

patronictl show-config / edit-config

```
postgres@patroni1:~$ patronictl show-config
loop_wait: 10
maximum_lag_on_failover: 1048576
postgresql:
  parameters:
    max_connections: 100
  pg_hba:
  - local all all trust
  - host replication replicator all md5
  - host all all all md5
  use_pg_rewind: true
retry_timeout: 10
ttl: 30
```

```
postgres@patroni1:~$ patronictl edit-config
---
+++
@@ -2,7 +2,7 @@
 maximum_lag_on_failover: 1048576
 postgresql:
   parameters:
-    max_connections: 100
+    max_connections: 200
   pg_hba:
   - local all all trust
   - host replication replicator all md5

Apply these changes? [y/N]: y
Configuration changed
postgres@patroni1:~$
```

patronictl restart

```
postgres@patroni1:~$ patronictl list
+ Cluster: demo --------+---------+-----------+----+-----------+-----------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Pending restart |
+----------+------------+---------+-----------+----+-----------+-----------------+
| patroni1 | 172.21.0.8 | Replica | streaming |  1 |         0 | *               |
| patroni2 | 172.21.0.2 | Replica | streaming |  1 |         0 | *               |
| patroni3 | 172.21.0.7 | Leader  | running   |  1 |           | *               |
+----------+------------+---------+-----------+----+-----------+-----------------+
postgres@patroni1:~$ patronictl restart demo
+ Cluster: demo --------+---------+-----------+----+-----------+-----------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Pending restart |
+----------+------------+---------+-----------+----+-----------+-----------------+
| patroni1 | 172.21.0.8 | Replica | streaming |  1 |         0 | *               |
| patroni2 | 172.21.0.2 | Replica | streaming |  1 |         0 | *               |
| patroni3 | 172.21.0.7 | Leader  | running   |  1 |           | *               |
+----------+------------+---------+-----------+----+-----------+-----------------+
When should the restart take place (e.g. 2023-08-10T03:07)  [now]:
Are you sure you want to restart members patroni1, patroni2, patroni3? [y/N]: y
Restart if the PostgreSQL version is less than provided (e.g. 9.5.2)  []:
Success: restart on member patroni1
Success: restart on member patroni2
Success: restart on member patroni3
```

patronictl reload

```
postgres@patroni1:~$ patronictl reload demo patroni1
+ Cluster: demo --------+---------+-----------+----+-----------+
| Member   | Host       | Role    | State     | TL | Lag in MB |
+----------+------------+---------+-----------+----+-----------+
| patroni1 | 172.24.0.5 | Leader  | running   |  1 |           |
| patroni2 | 172.24.0.3 | Replica | streaming |  1 |         0 |
| patroni3 | 172.24.0.4 | Replica | streaming |  1 |         0 |
+----------+------------+---------+-----------+----+-----------+
Are you sure you want to reload members patroni1? [y/N]: y
Reload request received for member patroni1 and will be processed within 10 seconds
postgres@patroni1:~$ patronictl list
+ Cluster: demo --------+---------+-----------+----+-----------+-------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Tags        |
+----------+------------+---------+-----------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Leader  | running   |  1 |           | pgday: true |
| patroni2 | 172.24.0.3 | Replica | streaming |  1 |         0 |             |
| patroni3 | 172.24.0.4 | Replica | streaming |  1 |         0 |             |
+----------+------------+---------+-----------+----+-----------+-------------+
```

patronictl pause / resume

```
postgres@patroni1:~$ patronictl pause
Success: cluster management is paused
postgres@patroni1:~$ patronictl list
+ Cluster: demo --------+---------+-----------+----+-----------+-------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Tags        |
+----------+------------+---------+-----------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Leader  | running   |  1 |           | pgday: true |
| patroni2 | 172.24.0.3 | Replica | streaming |  1 |         0 |             |
| patroni3 | 172.24.0.4 | Replica | streaming |  1 |         0 |             |
+----------+------------+---------+-----------+----+-----------+-------------+
 Maintenance mode: on
postgres@patroni1:~$ patronictl resume
Success: cluster management is resumed
```

patronictl switchover

```
postgres@patroni1:~$ patronictl switchover
Current cluster topology
+ Cluster: demo --------+---------+-----------+----+-----------+-------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Tags        |
+----------+------------+---------+-----------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Leader  | running   |  1 |           | pgday: true |
| patroni2 | 172.24.0.3 | Replica | streaming |  1 |         0 |             |
| patroni3 | 172.24.0.4 | Replica | streaming |  1 |         0 |             |
+----------+------------+---------+-----------+----+-----------+-------------+
Primary [patroni1]:
Candidate ['patroni2', 'patroni3'] []: patroni2
When should the switchover take place (e.g. 2023-08-04T18:34 )  [now]:
Are you sure you want to switchover cluster demo, demoting current leader patroni1? [y/N]: y
2023-08-04 17:34:24.78785 Successfully switched over to "patroni2"
+ Cluster: demo --------+---------+---------+----+-----------+-------------+
| Member   | Host       | Role    | State   | TL | Lag in MB | Tags        |
+----------+------------+---------+---------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Replica | stopped |    |   unknown | pgday: true |
| patroni2 | 172.24.0.3 | Leader  | running |  1 |           |             |
| patroni3 | 172.24.0.4 | Replica | running |  1 |         0 |             |
+----------+------------+---------+---------+----+-----------+-------------+
```

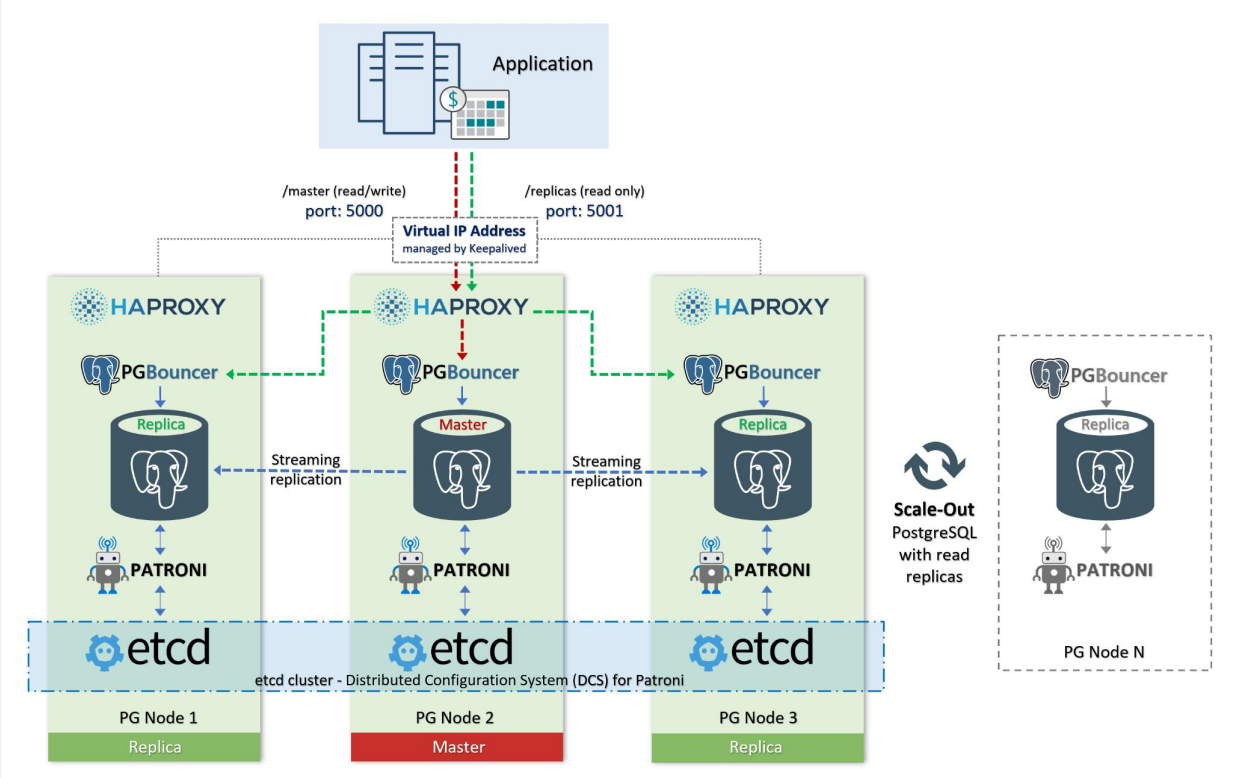
patronictl failover

```
postgres@patroni1:~$ patronictl failover demo
Current cluster topology
+ Cluster: demo --------+---------+-----------+----+-----------+-------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Tags        |
+----------+------------+---------+-----------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Replica | streaming |  2 |         0 | pgday: true |
| patroni2 | 172.24.0.3 | Leader  | running   |  2 |           |             |
| patroni3 | 172.24.0.4 | Replica | streaming |  2 |         0 |             |
+----------+------------+---------+-----------+----+-----------+-------------+
Candidate ['patroni1', 'patroni3'] []: patroni1
Are you sure you want to failover cluster demo, demoting current leader patroni2? [y/N]: y
2023-08-04 17:40:37.56115 Successfully failed over to "patroni1"
+ Cluster: demo --------+---------+---------+----+-----------+-------------+
| Member   | Host       | Role    | State   | TL | Lag in MB | Tags        |
+----------+------------+---------+---------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Leader  | running |  2 |           | pgday: true |
| patroni2 | 172.24.0.3 | Replica | stopped |    |   unknown |             |
| patroni3 | 172.24.0.4 | Replica | running |  2 |         0 |             |
+----------+------------+---------+---------+----+-----------+-------------+
postgres@patroni1:~$ patronictl list
+ Cluster: demo --------+---------+-----------+----+-----------+-------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Tags        |
+----------+------------+---------+-----------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Leader  | running   |  3 |           | pgday: true |
| patroni2 | 172.24.0.3 | Replica | streaming |  3 |         0 |             |
| patroni3 | 172.24.0.4 | Replica | streaming |  3 |         0 |             |
+----------+------------+---------+-----------+----+-----------+-------------+
```

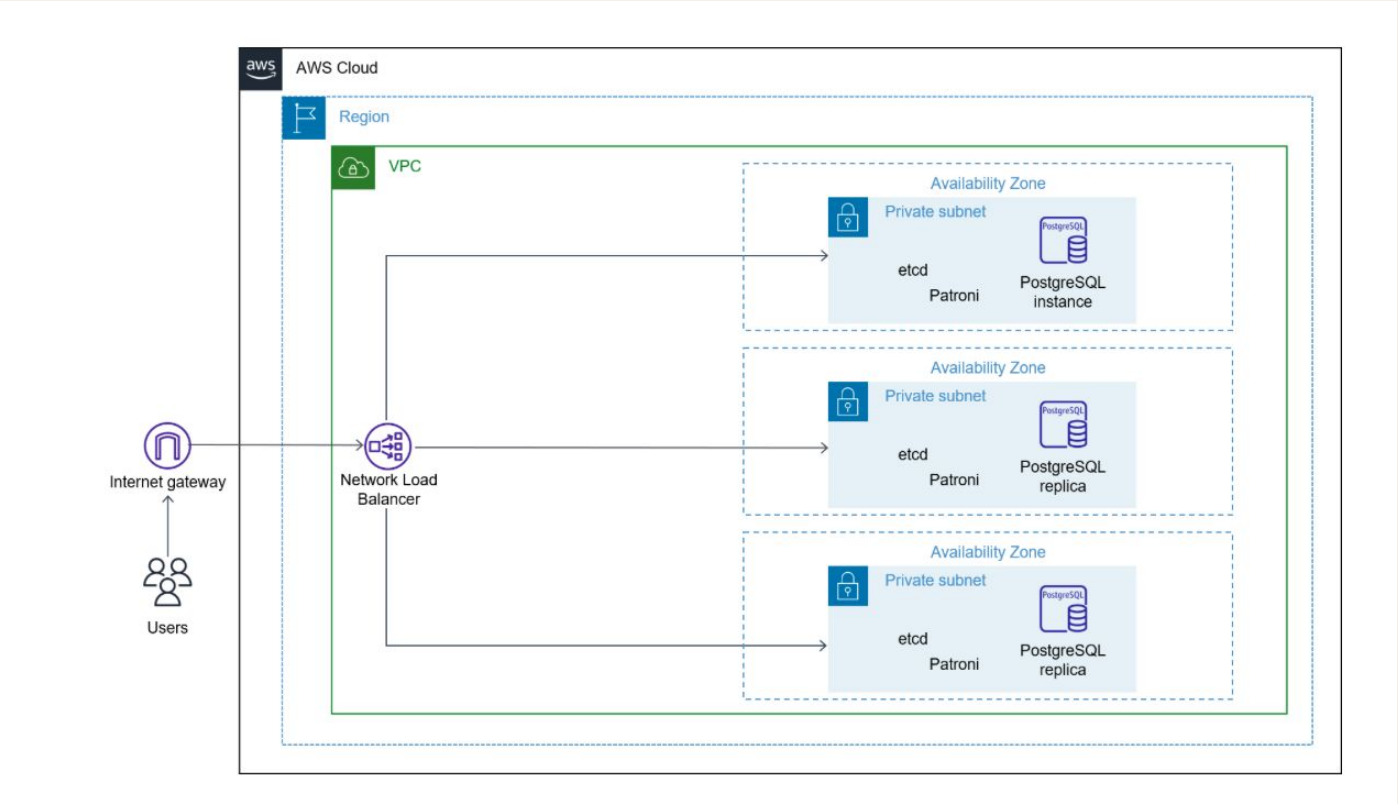
patronictl reinit

```
+----------+-----------+---------+-----------+----+-----------+-------------+
postgres@patroni1:~$ patronictl reinit demo patroni1
+ Cluster: demo --------+---------+-----------+----+-----------+-------------+
| Member   | Host       | Role    | State     | TL | Lag in MB | Tags        |
+----------+------------+---------+-----------+----+-----------+-------------+
| patroni1 | 172.24.0.5 | Replica | streaming |  2 |         0 | pgday: true |
| patroni2 | 172.24.0.3 | Leader  | running   |  2 |           |             |
| patroni3 | 172.24.0.4 | Replica | streaming |  2 |         0 |             |
+----------+------------+---------+-----------+----+-----------+-------------+
Are you sure you want to reinitialize members patroni1? [y/N]: y
Success: reinitialize for member patroni1
```

Exemplo de deploy 1



Exemplo de deploy 2



VÍDEO 1

VÍDEO 2

VÍDEO 3

Referências

https://www.secops.com.tr/tr/cluster/postgresql-patroni-pgbouncer-haproxy
https://patroni.readthedocs.io/en/master/index.html
https://patroni.readthedocs.io/en/master/dynamic_configuration.html
https://patroni.readthedocs.io/en/master/citus.html
https://patroni.readthedocs.io/en/master/README.html#technical-requirements-installation
https://bootvar.com/useful-patroni-commands/#:~:text=patronictl%20edit%2Dconfig,parameters%20before%20saving%20configuration%20file.
https://roxpartner.com/tipos-de-cluster-postgresql/
https://www.brianstorti.com/raft/
http://thesecretlivesofdata.com/raft/

OBRIGADO