Logische replicatie

* publiceer en abonneer een model (publisher vs subscriber)
* publicaties op een publicatie knooppunt (node)
* één of meer abonnees die zich abonneren

Publicatie

via CREATE PUBLICATION

* alleen INSERT
* ook UPDATE, DELETE
  + REPLICA IDENTITY nodig voor de uitgever: in de praktijk vaak een primaire sleutel

Abonnement

via CREATE SUBSCRIPTION

* alleen gewone tabellen
* geabonneerde tabellen moeten worden aangemaakt, de schemadefinitie wordt niet gerepliceerd
* FQTN afbeelding/mapping (fully qualified tabel name): tabelnamen moeten hetzelfde zijn als aan de publicatiezijde
  + Ook de kolommen moeten dezelfde naam en type hebben, maar ze kunnen extra kolommen hebben

Conflicten

* Beperkingen worden toegepast en moeten worden behandeld

Opmerkingen

* tabellen bij de uitgever en abonnee kunnen verschillen, maar het moet zin hebben
* sequentiegegevens worden niet gerepliceerd
* als truncate nodig is, controleer dan de fk constraints
* monitoring: pg\_stat\_subscription
* de rol gebruikt voor replicatie moet het kenmerk replication hebben of een superuser zijn
  + ten minste het select-voorrecht aan de publicerende kant is nodig
  + wil je tabellen toevoegen aan een publicatie, dan moet je de eigenaar zijn
* abonnementen worden aangemaakt en uitgevoerd door een superuser
  + postgresql.conf: de standaardinstellingen in Debian zouden moeten voldoen
    - max\_replication\_slots op 1 of hoger voor de demo, d.w.z. het aantal abonnementen
    - max\_logical\_replication\_workers: minimaal max\_replication\_slots (dus +5)
    - max\_worker\_processes: ten minste max\_logical\_replication\_workers+1

Demo

* publicatie: wal\_level = logisch

**ROLLBACK**;

**BEGIN**;

**CREATE** **SCHEMA** replicatie\_demo;

**GRANT** **usage** **on** **SCHEMA** replication\_demo **TO** **public**;

**SET** search\_path **TO** replication\_demo;

CREËER TABLE demo\_users(

user\_id serieel PRIMARY KEY,

voornaam varchar(32) NOT NULL,

achternaam varchar(64)

);

**GRANT** **select**, **insert** **ON** demo\_users **TO** **public**;

**CREATE** PUBLICATION mypub **FOR** **TABLE** demo\_users;

*--COMMIT;*

* abonnee, als superuser:

**CREATE** **SCHEMA** replicatie\_demo;

**GRANT** **usage** **on** **SCHEMA** replication\_demo **TO** **public**;

**SET** search\_path **TO** replication\_demo;

CREËER TABLE demo\_users(

user\_id serieel PRIMARY KEY,

voornaam varchar(32) NOT NULL,

achternaam varchar(64)

);

**CREATE** SUBSCRIPTION mysub

**CONNECTION** 'dbname=probeer host=databanken.ucll.be user=local\_userid port=year\_cluster\_port password=mypass sslmode=require'

PUBLICATIE mypub;

de gebruiker die verbinding maakt als abonnee heeft het replicatieprivilege aan de publicatiekant nodig

Ref: <https://www.postgresql.org/docs/current/logical-replication.html>

Oefening voor elke student

Schrijf je in op de publicatie *dataman\_pub* op machine *fuji.ucll.be* (poort van het academiejaar, tenzij anders meegedeeld). Deze publicatie geeft de wijzingen van de tabellen in het *chinook* schema van de *dataman* databank door. Dit is een andere server dan *databanken.ucll.be*.

* + - * met local\_r-nummer en local pass (52324 en database = dataman)

locale databank dit uitvoeren:

CREATE SUBSCRIPTION JentheVZ

CONNECTION 'dbname=dataman host=fuji.ucll.be user=local\_r0887523 port=52324 password=7xBVhXNBLV#oM3}f sslmode=require'

PUBLICATION dataman\_pub;

Hiermee zie je de subcriptions op databank fuji in dataman databank.

**select** \* **from** pg\_stat\_replication;

* Voeg data toe in 1 of meerdere tabellen rechtstreeks op de publicatie server (fuji), en kijk na of dit op je lokale ingeschreven server (jouw pc bv) binnenkomt
  + Voeg gerust meerdere rijen met je favoriete muziek toe.
* Verwijder (of pas) data aan in 1 of meerdere tabellen rechtstreeks op de publicatie server (fuji), en kijk na of dit op je lokale ingeschreven server (jouw pc bv) binnenkomt
  + Teneide de tabellen niet te groot te laten worden voor deze demo oefening. (Minder dan 10000 rijen/tabel is geen probleem)

Het is aan jou om na te kijken op welke tabellen je welke rechten hebt.

Indien je dit samen met andere studenten doet, dan zou je ook hun aanpassingen moeten zien.

Voorwaarde om de oefening te kunnen maken

Je gebruiker moet het attribuut REPLICATION hebben, dit kan je instellen op fuji in de databank administration via:

**SELECT** user\_administration.get\_replication();

Insert log

Stel je wil apart een log bijhouden van bepaalde data die wordt toegevoegd. Hierbij de gegevens die je nodig van de draaiende publicatie. Abonneer je ook op deze publicatie. Test ook dit met het toevoegen, aanpassen en verwijderen van een rij. Wat merk je op?

server zijde:

**create** publication album\_insert **for** **table** chinook."Album" **with** (publish='insert');

client zijde:

**create** subscription dataman\_test **connection** 'dbname=dataman host=fuji.ucll.be port=ac\_jaar\_poort user=local\_jestudentnr password=mypass sslmode=require' publication album\_insert;

🡪

create subscription dataman\_test

connection 'dbname=dataman host=fuji.ucll.be port=52324 user=local\_r0887523 password=7xBVhXNBLV#oM3}f sslmode=require'

publication album\_insert;

Opvolging

Publisher kant

In eerste instantie via

*-- actieve verbindingen*

**select** \* **from** pg\_stat\_replication;

*-- alle voorziene slots*

**select** \* **from** pg\_replication\_slots;

Meer info:

**select** \* **from** pg\_publication;

**select** \* **from** pg\_publication\_rel ;

**select** \* **from** pg\_publication\_tables ;

**select** \* **from** pg\_stat\_activity **where** backend\_type='walsender';

Bv

**select** \* **from** pg\_stat\_replication;

-[ RECORD 1 ]*----+------------------------------*

pid | 120310

usesysid | 16722

usename | local\_u0082489

application\_name | dataman\_sub

client\_addr | 94.225.42.148

client\_hostname |

client\_port | 40528

backend\_start | 2021-05-06 11:17:24.659576+02

backend\_xmin |

state | streaming

sent\_lsn | 0/CAE4500

write\_lsn | 0/CAE4500

flush\_lsn | 0/CAE4500

replay\_lsn | 0/CAE4500

write\_lag |

flush\_lag |

replay\_lag |

sync\_priority | 0

sync\_state | async

reply\_time | 2021-05-07 11:27:57.338105+02

..

**select** pg\_current\_wal\_lsn();

pg\_current\_wal\_lsn

*--------------------*

0/CAE4500

Wat zie je hier:

* geen meetbare vertraging (lag), dat is goed.
* 0/CAE4500 komt overal overeen, dus we zijn bij en alles is persistent in de tabellen opgeslagen
  + lsn staat hier voor log sequence number (cf uitleg concept transactielog die wordt afgespeeld)
* async, cf CAP theorema

Subscriber kant

In eerste instantie:

*-- info over de status van de aanwezige subscripties (cluster breed)*

**select** \* **from** pg\_stat\_subscription ;

*-- info over de aanwezige aangemaakte subscripties (cluster breed)*

**select** \* **from** pg\_subscription;

Meer info:

**select** \* **from** pg\_replication\_origin;

**select** \* **from** pg\_replication\_origin\_status ;

**select** \* **from** pg\_subscription;

**select** \* **from** pg\_subscription\_rel;

**select** \* **from** pg\_stat\_activity **where** backend\_type='logical replication worker';

Bv

**select** \* **from** pg\_stat\_subscription ;

-[ RECORD 1 ]*---------+------------------------------*

subid | 25000

subname | dataman\_sub

pid | 24337

relid |

received\_lsn | 0/CAE4500

last\_msg\_send\_time | 2021-05-07 11:43:29.295348+02

last\_msg\_receipt\_time | 2021-05-07 11:43:29.310246+02

latest\_end\_lsn | 0/CAE4500

latest\_end\_time | 2021-05-07 11:43:29.295348+02

Wat zie je hier:

* 0/CAE4500 komt overeen, dus we zijn bij

Verwijderen subscriptie

Via DROP SUSBSCRIPTION, hiermee verwijder je niet alleen lokaal je subscriptie, maar geef je ook deze boodschap aan de publisher.

Verwijder je supcripties/abonnees als ze voldoende getest zijn. Het is normaal niet erg als ze lang draaien (bv verschillende dagen, weken), het kan wel zijn dat er periodiek aan de publicatie kant wordt opgekuist, waardoor je abonnee niet meer werkt.

Zorg ervoor dat je je vlot kan abonneren (en opzeggen).

|  |  |
| --- | --- |
| **NOTE** | Als de publisher niet meer bereikbaar is, zal je eerst je subscriptie moeten aanpassen. Dit is een ingebouwde bescherming. Je kan hiervoor ALTER SUBSCRIPTION .. SET (slot\_name = NONE) gebruiken. |

Problemen?

Enkele suggesties:

* kijk de subscriptie na
  + kan de gebruiker rechtstreeks aan de tabellen op de publishing server?
  + heeft de gebruiker het replicatie attribuut op de publishing server?
  + heb je de juiste parameters gebruikt?
* is de subscriptie actief?
* is de view *pg\_subscription\_rel* niet leeg of onvolledig? (er zijn bv tabellen op de publishing server bijgekomen)
* algemeen:
  + foutmeldingen nakijken
  + log server nakijken
* replicatie via de pooling poort is niet ondersteund

Je kan de subscriptie aanmaken, aanpassen, verwijderen via de gebruikelijke sleutelwoorden (create, alter, drop).

Bv.

*-- je ziet bv niet de publicaties van de tabellen (pg\_subscription\_rel)*

**alter** subscription dataman\_sub **refresh** publication;

*-- staat bv niet aan (pg\_subscription)*

**alter** subscription dataman\_sub **enable**;

Interessant

Het is altijd interessant om inzicht te hebben in de structuur van de databank, probeer uit te zoeken wat het datamodel is van de databank die je repliceerd hebt.

Uitbreiding

Zet nu ook zelf een publicatie op en abonneer je hierop.

* lokaal op je cluster van databankX naar databankY
* van clusterX naar clusterY

Last updated 2024-04-26 14:40:32 +0200