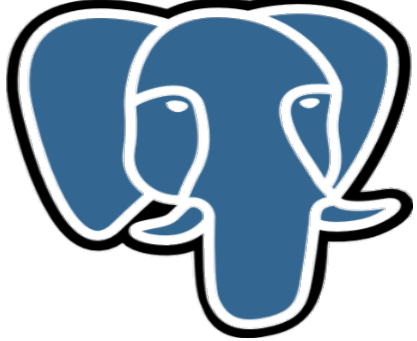
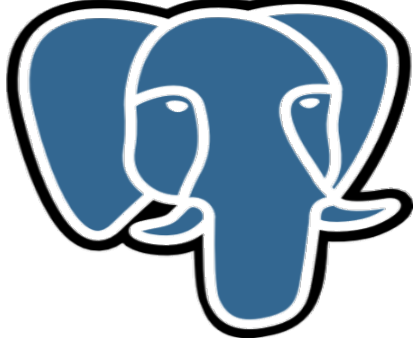
Replicación usando Slony-I

por Jaime Casanova

PostgreSQL

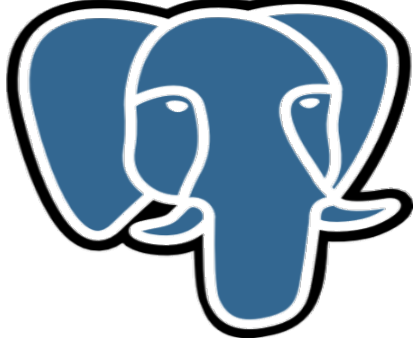


- Conceptos
- Requisitos
- Configuración
- Iniciando la replicación
- Mantenimiento



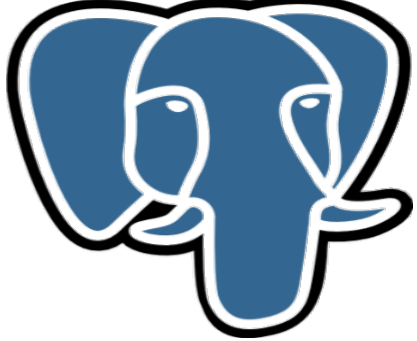
Conceptos

- Cluster: Es el conjunto de instancias de bases de datos PostgreSQL que están envueltos en la replicación.
- Nodo: Se le llama así a cada una de las bases de datos envueltas en la replicación.
- Replication set: Es el conjunto de tablas y/o secuencias a ser replicadas. En un mismo cluster pueden haber varios sets.



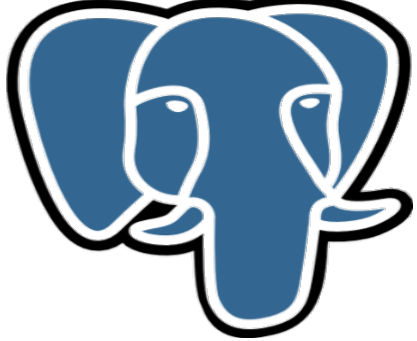
Conceptos

- Origin: Es el nodo principal (maestro), es el único en el que se puede escribir.
- Subscribers: Son todos los demás nodos en el cluster (esclavos), son los que reciben los datos en la réplica.
- Providers: Es un nodo subscriber (esclavo) que sirve como proveedor para un subconjunto de nodos en el cluster (actúa como un nodo origin pero no se permite a ninguna aplicación escribir en él).



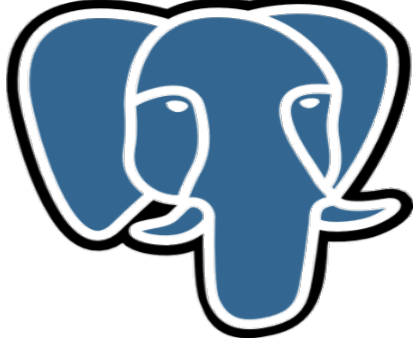
Requisitos

- Postgresql >= 7.3.3 (7.4.8 o superior es recomendado).
  - Verificar que postgres este aceptando conexiones (listen_addresses='*' y el pg_hba.conf)
- Definir la estructura del cluster (a cada nodo del cluster se le debe identificar con un número).



Requisitos

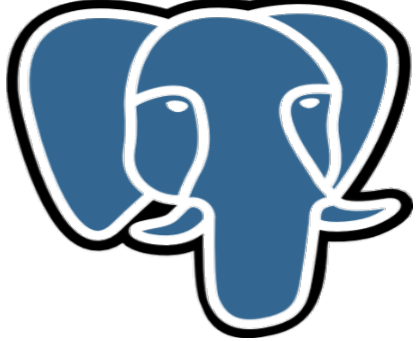
- Definir los replication sets (llaves para las tablas que no tienen un PK, tablas, secuencias).
    - Tablas relacionadas por un FK deberían estar en el mismo replication set.



Configuración

IMPORTANTE

- Todos los nodos involucrados en la replicación deben estar usando un timezone reconocido por PostgreSQL (en Ecuador lo correcto es GMT+5)
    - Se debe setear en el archivo postgresql.conf



Configuración

- Usaremos un rol especifico para la replicación.

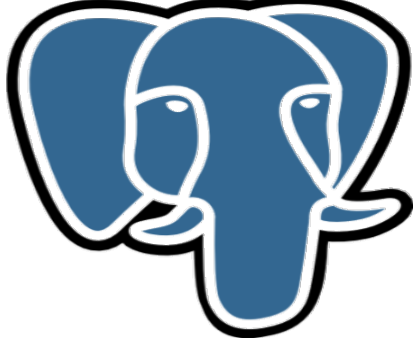
  create user slony superuser;

- Se deben crear los roles y tablespaces en los nodos.

  pg_dumpall -r | psql -h dir_ip -p puerto
  pg_dumpall -t | psql -h dir_ip -p puerto

- Crear la base de datos y el esquema de la base de datos en los nodos.

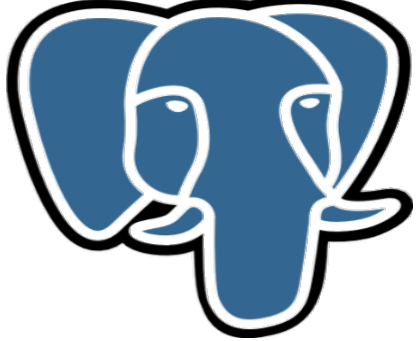
  pg_dump -C -s base_datos | psql -h dir_ip -p puerto



Iniciando la replicación

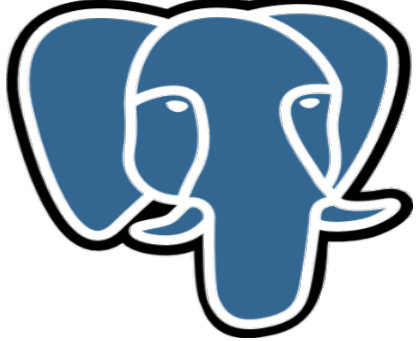
Las siguientes líneas se deben incluir en todos los scripts de slonik, grabelas en un archivo que será incluido en los demás scripts. Yo lo llame common.slonik:

```
cluster name = $nombre_cluster
node 1 admin conninfo = 'host=$ip_maestro dbname=$base user=$usuario';
node 2 admin conninfo = 'host=$ip_esclavo dbname=$base user=$usuario';

[añada tantas líneas como nodos tenga su cluster...]
```



Iniciando la replicación

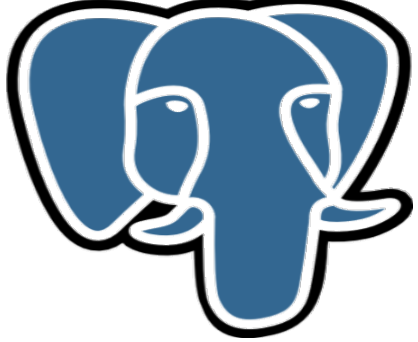
- Inicializar el cluster y los nodos.
- Guardar las rutas para comunicar los diferentes nodos entre sí.
- Iniciar los procesos slon.
- Crear un replication set.
- Añadir tablas y secuencias al replication set.
- Subscribir los nodos al replication set.



Iniciando la replicación

Debemos inicializar el cluster indicando cual será el maestro y luego incluimos los nodos:

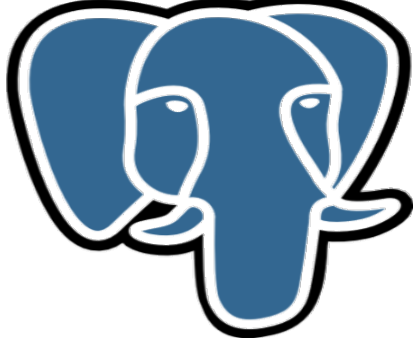
```
#!/bin/sh

slonik <<_EOF_
    include <common.slonik>;

    init cluster (id=1, comment = 'maestro');
    store node (id=2, comment = 'esclavo1');
_EOF_
```



Iniciando la replicación

Debemos indicarle como comunicar cada uno de los servidores con los otros:

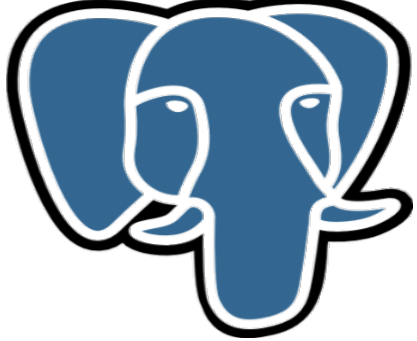
```
#!/bin/sh

slonik <<_EOF_
    include <common.slonik>;

    store path (server = 1, client = 2, conninfo='dbname=$base host=$ip_maestro
user=$usuario');
    store path (server = 2, client = 1, conninfo='dbname=$base host=$ip_esclavo
user=$usuario');
_EOF_
```



Iniciando la replicación

Iniciamos los procesos slon. Uno por cada nodo:

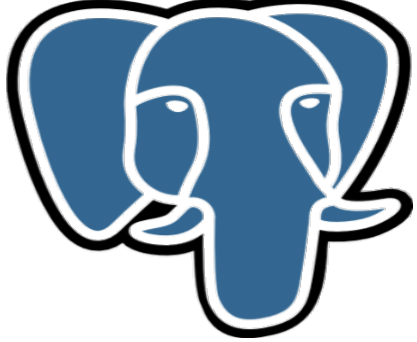
```
slon -d2 $cluster_name 'host=$ip_maestro dbname=$base user=$usuario' &
slon -d2 $cluster_name 'host=$ip_esclavo dbname=$base user=$usuario' &
```



Iniciando la replicación

Creamos un replication set, este es el conjunto de tablas y secuencias a ser copiadas:

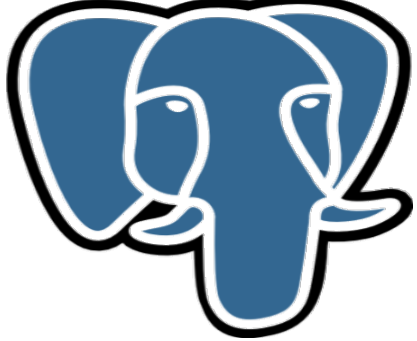
```
#!/bin/sh

slonik <<_EOF_
    include <common.slonik>;

     create set (id=1, origin=1, comment='Primer grupo de tablas');
_EOF_
```



Iniciando la replicación

Se añaden tablas y secuencias al replication set, esto debe planificarse de forma adecuada puesto que no se pueden añadir mas tablas una vez que el replication set se este replicando:

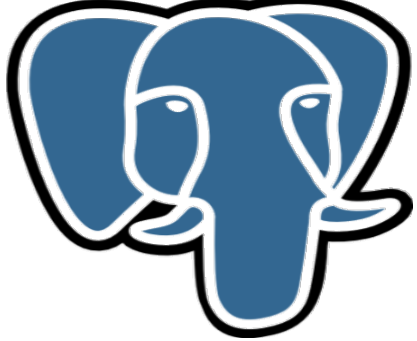
```
#!/bin/sh

slonik <<_EOF_
    include <common.slonik>;

 set add sequence (set id=1, origin=1, id=1, fully qualified name = 'public.
address_book_address_book_id_seq', comment='address book id');
 set add table (set id=1, origin=1, id=1,    fully qualified name = 'public.address_book',
comment='address booK table');

_EOF_
```



Iniciando la replicación

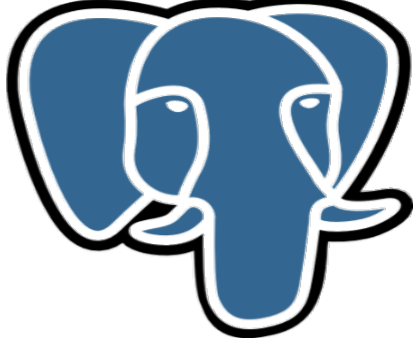
Finalmente subscribimos los nodos a los replication sets:

```
#!/bin/sh

slonik <<_EOF_
    include <common.slonik>;

    subscribe set (id=1, provider=1, receiver=2, forward=yes);

_EOF_
```



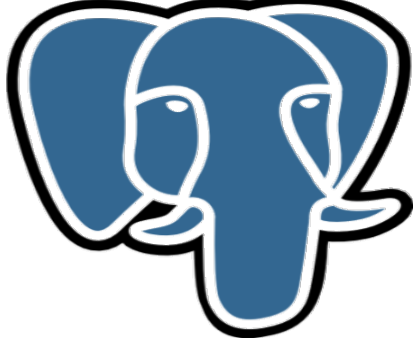
Mantenimiento: Añadir una nueva tabla

- Crear un nuevo replication set.
- Añadir la tabla y/o secuencia al nuevo set.
- Subscribir los nodos al replication set.
- Merge set.



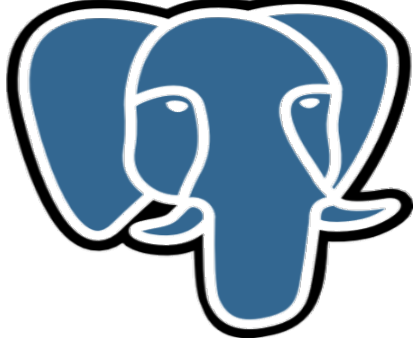
Mantenimiento: Quitar una tabla

- set drop table.
- set drop sequence.



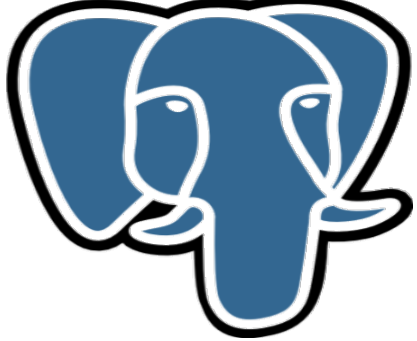
Mantenimiento: Añadir un nuevo replication set

- create set.
- set add table / set add sequences.
- subscribe set.



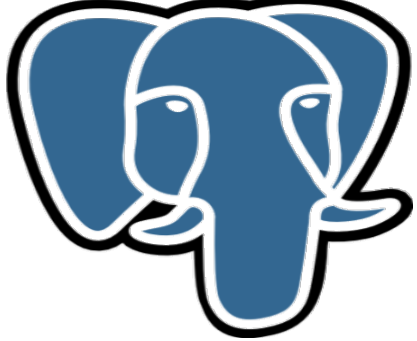
Mantenimiento: Quitar un replication set

- De un solo nodo:
  - unsubscribe set.
- De todos los nodos:
  - drop set.



Mantenimiento: Cambiar papeles

- lock set(id=1, origin=1);
- wait for event (origin = 1, confirmed = 2);
- move set (id = 1, old origin = 1, new origin = 2);
- wait for event (origin = 1, confirmed = 2);



Mantenimiento: Failover

- FAILOVER (ID = 1, BACKUP NODE = 2);