

NO	概要	対象	ユーザー	コメント	確認 (確認コマンド等)	備考
1	既存PostgreSQLの設定変更	既存PostgreSQLサーバ	postgres	\$ W /home/postgres/data/postgresql.conf max_wal_senders = 1 (インストール時) \$ W /home/postgres/data/pg_hba.conf host replication all (新規PostgreSQLサーバのIPアドレス) trust		
2	PostgreSQLの起動	既存PostgreSQLサーバ	postgres	\$ pg_ctl start server starting.	\$ pg_ctl status pg_ctl: server is running (PID: 23121) /usr/local/pgsql/bin/postgres	
3	PostgreSQLのバージョン確認	既存PostgreSQLサーバ	postgres	\$ psql psql (9.2.4) Type "help" for help. postgres=# select version(); ----- version ----- PostgreSQL 9.2.4 on x86_64-unknown-linux-gnu, compiled by gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-3), 64-bit (1 row)		
4	ydbc-runnerのインストール	既存PostgreSQLサーバ	任意	ydbcRunnerのインストールは、1節「インストール」の手順を参照して下さい。 \$ cd /opt/tpcc	\$ psql -l ----- List of databases Name Owner Encoding Collate Ctype Access privileges ----- postgres postgres UTF8 =c/postgres + template0 postgres UTF8 =c/postgres + template1 postgres UTF8 =c/postgres + tpcc postgres UTF8 =c/postgres + ----- (6 rows)	
5	検証用データベースの作成	既存PostgreSQLサーバ	postgres		\$ psql -d tpcc psql=# \d ----- List of relations Schema Name Type Owner ----- public customer table postgres public district table postgres public history table postgres public item table postgres public new_orders table postgres public order_line table postgres public orders table postgres public stock table postgres public warehouse table postgres ----- (9 rows)	
6	検証用データの投入	既存PostgreSQLサーバ	任意	\$ export CLASSPATH=\$CLASSPATH:/ydbcRunnerを参照したディレクトリ/ydbcrunner-1.2.jar \$ java -R ydbcRunnerを参照したディレクトリ/scripts/tpcc_load.py -ydbcDriver org.postgresql.Driver -ydbcUrl jdbc:postgresql://PostgreSQLサーバのIPアドレス:5432/tpcc -ydbcUser postgres -ydbcDir 検証用のログ出力ディレクトリ		
7	検証用データの件数確認	既存PostgreSQLサーバ	postgres	検証用データの投入確認のため、以下のSQL文を実行 \$ psql tpcc ----- count ----- 480000 (1 row) ----- =# select count(*) from customer; ----- count ----- 480000 (1 row) ----- =# select count(*) from district; ----- count ----- 160 (1 row) ----- =# select count(*) from history; ----- count ----- 480000 (1 row) ----- =# select count(*) from item; ----- count ----- 480000 (1 row) ----- =# select count(*) from new_orders; ----- count ----- 480000 (1 row) ----- =# select count(*) from order_line; ----- count ----- 480000 (1 row) ----- =# select count(*) from orders; ----- count ----- 480000 (1 row) ----- =# select count(*) from stock; ----- count ----- 1600000 (1 row) ----- =# select count(*) from warehouse; ----- count ----- 16 (1 row)		
8	PostgreSQL 9.2.4のインストール	新規PostgreSQLサーバ (新規PostgreSQL)		PostgreSQLのインストールは、1節「インストール」を参照して下さい。 以下コマンドを実行 \$ cd /usr/local/pgsql9.2 ./configure --prefix=/usr/local/pgsql9.2 --with-libxml ※ データベーススタは作成しません。		
9	PostgreSQL 9.3.2のインストール	新規PostgreSQLサーバ (新規PostgreSQL)		PostgreSQLのインストールは、1節「インストール」を参照して下さい。 ※ データベーススタは作成しません。		
10	新規PostgreSQLサーバ用の設定ファイルの作成	新規PostgreSQLサーバ	postgres	\$ W /home/postgres/postgres9.2_profile export PATH=/usr/local/pgsql9.2/bin:\$PATH export MANPATH=/usr/local/pgsql9.2/share/man:\$PATH export LD_LIBRARY_PATH=/usr/local/pgsql9.2/lib:\$LD_LIBRARY_PATH export PGDATA=/home/postgres/data9.2		
11	pg_upgradeのコンパイル	新規PostgreSQLサーバ (新規PostgreSQL)	postgres	\$ cd /usr/local/src/postgresql-9.3.2/contrib/pg_upgrade \$ make ※ PostgreSQLインストール時に、contribモジュールもインストールしている場合は、項書12～15の手順は不要		
12	pg_upgradeのインストール	新規PostgreSQLサーバ (新規PostgreSQL)	root	# cd /usr/local/src/postgresql-9.3.2/contrib/pg_upgrade # make install	# su - postgres \$ pg_upgrade pg_upgrade upgrades a PostgreSQL cluster to a different major version. Usage: pg_upgrade [OPTION]... (省略)	
13	pg_upgrade_supportのコンパイル	新規PostgreSQLサーバ (新規PostgreSQL)	postgres	\$ cd /usr/local/src/postgresql-9.3.2/contrib/pg_upgrade_support/ \$ make		

14	pg_upgrade_supportのインストール	新規PostgreSQLサーバ (新規PostgreSQL)	root	# cd /usr/local/src/postgresql-9.3.2/contrib/pg_upgrade_support/ # make install		
15	初期PostgreSQL用データベースクラスタ格納ディレクトリの作成	新規PostgreSQLサーバ (初期PostgreSQL)	postgres	# mkdir /home/postgres/data9.2/	# ls -l /home/postgres/data9.2 合計 0	
16	初期PostgreSQL用データベースクラスタ格納ディレクトリの作成	新規PostgreSQLサーバ (新規PostgreSQL)	postgres	# mkdir /home/postgres/data/ # mv /home/postgres/data9.2/* /home/postgres/data/	# ls -l /home/postgres/data 合計 0	
17	データベースクラスタの作成	新規PostgreSQLサーバ (新規PostgreSQL)	postgres	\$ initdb -E UTF-8 --pgdata=/home/postgres/data/ --no-localize (データベースクラスタ作成時のオプションは、前バージョンでの値に合わせる)	\$ ls -l /home/postgres/data 合計 88 ----- 1 postgres postgres 4 9月 13 19:45 2014 PG_VERSION drwxr-xr-x 5 postgres postgres 4096 3月 13 19:45 2014 base drwxr-xr-x 2 postgres postgres 4096 3月 13 19:45 2014 global drwxr-xr-x 2 postgres postgres 4096 3月 13 19:45 2014 pg_log (省略)	
18	ベースバックアップの取得	新規PostgreSQLサーバ (初期PostgreSQL)	postgres	\$ source .postgres_9.2_profile \$ pg_basebackup -h [既存PostgreSQLサーバのIPアドレス] -p 5432 -D /home/postgres/data9.2/ --log --progress --verbose log start point: 1/50000020		
19	pg_receivelogの実行	新規PostgreSQLサーバ (初期PostgreSQL)	postgres	\$ pg_receivelog -D /home/postgres/data9.2/pg_xlog -h [既存PostgreSQLサーバのIPアドレス] --verbose pg_receivelog: starting log streaming at 0/A000000 (timeline 1) (既定値が保持されるため、既存PostgreSQLサーバ停止後にCtrl + C等で停止させます)		
20	既存PostgreSQLの停止	既存PostgreSQLサーバ	postgres	\$ pg_ctl stop waiting for server to shut down...LOG: received smart shutdown request LOG: autovacuum launcher shutting down LOG: shutting down LOG: database system is shut down done server stopped	\$ pg_ctl status pg_ctl: no server running	
21	初期PostgreSQLの起動 (差分アーカイブログの適用)	新規PostgreSQLサーバ (初期PostgreSQL)	postgres	\$ pg_ctl start server starting.	\$ pg_ctl status pg_ctl: server is running (PID: 23121) /usr/local/pg9.2/bin/postgres	
22	初期PostgreSQLの停止	新規PostgreSQLサーバ (初期PostgreSQL)	postgres	\$ pg_ctl stop waiting for server to shut down...LOG: received smart shutdown request LOG: autovacuum launcher shutting down LOG: shutting down LOG: database system is shut down done server stopped	\$ pg_ctl status pg_ctl: no server running	pg_receivelogをここで停止させる
23	複製変数の設定	新規PostgreSQLサーバ (新規PostgreSQL)	postgres	\$ export PGDATAOLD=/home/postgres/data9.2/ \$ export PGDATA_NEW=/home/postgres/data/ \$ export PGBINOLD=/usr/local/pg9.2/bin/ \$ export PGBIN_NEW=/usr/local/pg9.2/bin/		
24	pg_upgradeによるデータ移行の 前準備	新規PostgreSQLサーバ (新規PostgreSQL)	postgres	\$ source .postgres_profile \$ date; pg_upgrade -c -r date 2014年 3月 13日 木曜日 20:02:12 JST ----- Performing Consistency Checks ----- Checking cluster versions ok Checking database user is a superuser ok Checking for prepared transactions ok Checking for rep* system OID user data types ok Checking for contrib/ on with bigint-passing mismatch ok Checking for presence of required libraries ok Checking database user is a superuser ok Checking for prepared transactions ok ----- "Clusters are compatible" 2014年 3月 13日 木曜日 20:02:16 JST		
25	pg_upgradeによるデータ移行 (ハードリンクモード)	新規PostgreSQLサーバ (新規PostgreSQL)	postgres	\$ date; pg_upgrade --link -d date 2014年 3月 19日 火曜日 20:12:32 JST ----- Performing Consistency Checks ----- Checking cluster versions ok Checking database user is a superuser ok Checking for prepared transactions ok Checking for rep* system OID user data types ok Checking for contrib/ on with bigint-passing mismatch ok Creating dump of global objects ok Creating dump of database schemas ok Checking for presence of required libraries ok Checking database user is a superuser ok Checking for prepared transactions ok ----- If pg_upgrade fails after this point, you must re-initdb the new cluster before continuing. Performing Upgrade ----- Analyzing all rows in the new cluster ok Freezing all rows on the new cluster ok Deleting files from new pg_xlog ok Copying old pg_log to new server ok Setting next transaction ID for new cluster ok Setting oldest multixact ID on new cluster ok Resetting WAL archives ok Setting frozenxid counters in new cluster ok Restoring global objects in the new cluster ok Adding support functions to new cluster ok Restoring database schemas in the new cluster ok Removing support functions from new cluster ok Adding "old" suffix to old global/pg_control ok ----- If you want to start the old cluster, you will need to remove the "old" suffix from /home/postgres/data_9.2/global/pg_control.old. Because "link" mode was used, the old cluster cannot be safely started once the new cluster has been started. Linking user relation files ----- Setting next OID for new cluster ok Sync data directory to disk ok Creating script to analyze new cluster ok Creating script to delete old cluster ok		

