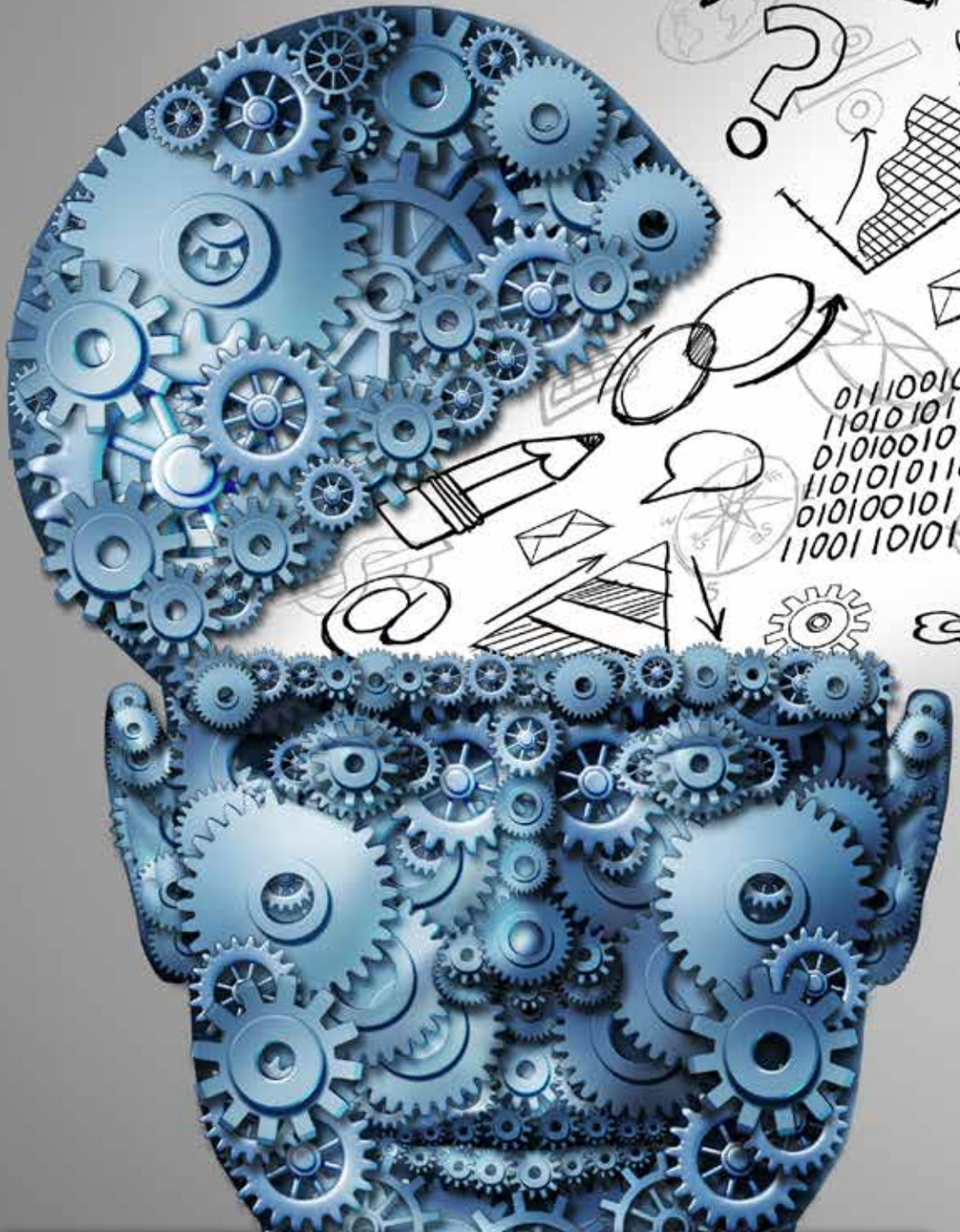


DOAG SOUG News



Article

Virtualisation de données:
Jouer avec Oracle 12c sur les
containers de Docker

SOUG

27ème rapport annuel du
président (2014)



Gaetano Bisaz
SOUG

Chers(es) adhérent(es) du SOUG.

Le lien entre le Cloud et la Virtualisation résulte d'une continuité logique en observant l'évolution des dernières années. Ceux-ci permettent d'exploiter des "Apps" plus souples et d'améliorer leur diffusion – un Time to market précieux et plus court. De nouvelles possibilités sont disponibles grâce à la combinaison de Docker et CoreOS. Il est intéressant d'expérimenter ces outils, afin d'en découvrir le potentiel. Aujourd'hui, plus facile que jamais avec les nombreux tutoriels et environnements de tests disponibles en ligne. Franck Pachot va plus loin, il les a testé et documenté pour vous – L'article en pièce jointe est captivant et invite à le faire de même.

Le "jeu" semble effectivement (enfin) à nouveau faire son entrée dans notre branche. Celui-ci est nécessaire pour trouver les perles rares englouties dans la marée montante des Buzzwords, Hypes et des Systèmes hybrides, qui permettront de réaliser au mieux les prochains projets. L'ouverture du système gagne en importance, ce qui est un élément captivant – La coqueluche des investisseurs, celui qui fait les gros titres peut disparaître aussi vite qu'il est apparu. Brusquement. Il est donc conseillé d'avoir un plan "B". Et comment trouve-t-on ce plan "B"? De manière similaire – En testant continuellement les nouvelles offres.

Avez-vous récemment "joué" avec les nouvelles possibilités de l'environnement des bases de données professionnelle Oracle? Avez-vous expérimenté les nombreuses nouvelles fonctionnalités captivantes d'Oracle 12C? Partagez vos expériences: mail à nl@soug.ch.

Gaetano Bisaz

Agenda

11.06.2015
SOUG SIG
sekretariat@soug.ch

03.09.2015
SOUG SIG
sekretariat@soug.ch

17.-20.11.2015
DOAG 2015 Konferenz + Ausstellung

J'ai besoin de connaître mon adresse IP:

```
docker@boot2docker:~$ ip addr
...
    inet 192.168.59.104/24 brd 192.168.59.255
scope global
...

```

L'adresse IP est: 192.168.59.104

Installation d'Oracle

Avec Docker, vous ne commencez jamais à partir de zéro. Vous commencez à partir d'un conteneur que vous téléchargez (pull) à partir du dépôt. Dans le dépôt public, vous ne pouvez pas trouver une image avec Oracle déjà installé. La raison est liée aux licences : la gestion des licences interdit la distribution d'une VM avec le logiciel déjà installé. Il y a une seule exception à cela, c'est Oracle XE. Et vous pouvez trouver des images Docker avec Oracle XE dans le référentiel. Mais je veux utiliser les fonctionnalités complètes d'Oracle 12c et je vais configurer la standby avec DataGuard, donc je dois l'installer moi-même.

Ok, je vais commencer à partir d'une image qui est prête pour Oracle. C'est la 'breed85 / oracle-12c', vous pouvez l'obtenir avec cette commande:

```
docker@boot2docker:~$ docker pull breed85/oracle-12c
```

Les voici:

```
docker@boot2docker:~$ docker images
REPOSITORY          TAG          IMAGE ID      CREATED      VIRTUAL SIZE
breed85/oracle-12c  latest      9d3b9aab053b 11weeks ago 1.728GB
breed85/oracle-12c  preinstall  9d3b9aab053b 11weeks ago 1.728GB

```

Tout est là sauf oracle. Vous pouvez télécharger le logiciel sur le site d'oracle (<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/>) et obtenir les deux fichiers zip : linuxamd64_12102_database_1of2.zip et linuxamd64_12102_database_2of2.zip. Vous devez les placer dans un répertoire que vous pouvez monter dans le conteneur Docker.

Comme j'utilise boot2docker, je partage le répertoire '/Users' sous Windows dans la VM boot2docker.

Je lance maintenant l'image 'breed85/oracle-12c' avec le tag '12101' et je partage le boot2docker /Users en '/User' dans le conteneur:

```
docker@boot2docker:~$ docker run --name
12102 -ti -v /Users:/Users breed85/oracle-
12c:preinstall /bin/bash
```

Si vous n'avez pas récupéré l'image 'breed85/oracle-12c', téléchargez là. Ensuite, je dois installer les binaires 'unzip' ainsi que 'preinstall' pour être sûr de les avoirs.

```
[root@2cd790c5a069 /]# yum -y install oracle-
rdbms-server-12cR1-preinstall unzip
```

Je fais un peu d'espace dans le yum cache et je crée le répertoire /oracle où je vais mettre tout ce qui doit être accessible par oracle:

```
[root@2cd790c5a069 /]# mkdir -p /oracle/install
[root@2cd790c5a069 /]# chown -R oracle:dba /
oracle
```

Décompression des fichiers:

```
[root@2cd790c5a069 /]# unzip
/Users/linuxamd64_12102_database_1of2.zip -d
/oracle/install/linuxamd64_12102_database
[root@2cd790c5a069 /]# unzip
/Users/linuxamd64_12102_database_2of2.zip -d
/oracle/install/linuxamd64_12102_database
```

Je suis prêt à installer. Je génère un 'response file' en mode silencieux:

```
[root@2cd790c5a069 /]# cat > /oracle/db_in-
stall.rsp <<-'CAT'
oracle.install.responseFileVersion=/oracle/in-
stall/rspfmt_dbinstall_response_schema_v12.1.0
oracle.install.option=INSTALL_DB_SWONLY
ORACLE_HOSTNAME=
UNIX_GROUP_NAME=oinstall
INVENTORY_LOCATION=/oracle/oraInventory
SELECTED_LANGUAGES=en
ORACLE_HOME=/oracle/product/12102
ORACLE_BASE=/oracle
oracle.install.db.InstallEdition=EE
oracle.install.db.DBA_GROUP=dba
oracle.install.db.OPER_GROUP=dba
oracle.install.db.BACKUPDBA_GROUP=dba
oracle.install.db.DGDBA_GROUP=dba
oracle.install.db.KMDBA_GROUP=dba
CAT
```

Et je l'installe en tant qu'utilisateur oracle:

```
[root@2cd790c5a069 /]# su - oracle -c
«/oracle/install/linuxamd64_12102_database/
database/runInstaller -silent -ignoreSysPre
reqs -ignorePrereq -silent -noconfig -respon
seFile /oracle/db_install.rsp»
```

J'ignore beaucoup de prérequis car je ne suis de toute façon pas sur une configuration supportée. À ce moment-là, je dois patienter 5 minutes pour l'installation, puis exécuter le script root.sh

```
[root@2cd790c5a069 /]# /oracle/oraInventory/
oraInstRoot.sh
[root@2cd790c5a069 /]# /oracle/product/12102/
root.sh
```

Je n'ai pas beaucoup d'espace dans ma VM Docker, donc je supprime les fichiers décompressés dont je n'ai plus besoin ainsi que d'autres dont je n'aurais pas besoin (Ai-je déjà dit de ne pas le faire en production?)

```
[root@2cd790c5a069 /]# rm -rf /var/cache/yum/
x86_64
/oracle/install/linuxamd64_12102_database
/oracle/product/12102/inventory/backup/*
/oracle/product/12102/assistants/dbca/templates
```

Bien sûr, j'aurais pu créer la VM Docker un peu plus grande, mais c'est juste un test:

```
[root@2cd790c5a069 /]# exit
```

Ok, c'était la partie complète de la préparation, téléchargement, décompression et installation. Je l'enregistre dans une nouvelle image.

```
docker@boot2docker:~$ docker commit 12102
breed85/oracle-12c:12102
```

Je pourrais vous aider et juste mettre cette image dans le référentiel docker, mais je suis désolé, je ne veux pas payer les licences Entreprise Edition pour les CPUs qui sont derrière ;)

Voici mes images:

```
docker@boot2docker:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          VIRTUAL SIZE
breed85/oracle-12c 12102       9cc040ef1d6e    5min ago        6.264GB
breed85/oracle-12c latest      9d3b9aab053b    11weeks ago     1.728GB
breed85/oracle-12c preinstall 9d3b9aab053b    11weeks ago     1.728GB
```

Et je peux supprimer le container que je viens de valider:

```
docker@boot2docker:~$ docker ps -as
CONTAINER ID IMAGE          COMMAND
CREATED      STATUS        PORTS
NAMES SIZE
2cd790c5a069 breed85/oracle-12c:latest «/bin/
bash» 18 minutes ago Exited (0) 5 minutes ago
12102 4.536 GB

docker@boot2docker:~$ docker rm 12102
```

Dupliquer la base de données

Comme mon but est de simuler la virtualisation d'une base de données existante, je la dupliquerai avec RMAN et afin d'en faire une physical standby. Ma base de données source est créée simplement avec dbca et toutes les options par défaut (Sauf que je la crée comme une non-CDB pour garder notre exemple simple).

Ma base de données source est appelée DEMO111 et est sur l'hôte 192.168.78.111 et le mot de passe SYS est «oracle» et j'ai paramétré dg_broker_start=true comme je créerais la standby avec le broker de DataGuard. Ma VM docker a l'adresse IP 192.168.59.104. Donc si vous faites un copier/coller, vous devriez changer ces valeurs.

J'exécute un shell dans un container depuis l'image que j'ai créé auparavant. Je l'appelle 'DEMO111_SBY' comme hostname. Je n'ai plus besoin du répertoire /Users. Mais maintenant, je vais rediriger le port 1521 vers le port 9000. Cela signifie que quand je démarre le listener sur le port 1521 dans le container (écoute de //DEMO111_SBY:1521 dans le container) je serai en mesure d'accéder depuis l'extérieur avec l'adresse IP //192.168.59.104:9000

```
RMAN> report schema;
```

```
using target database control file instead of recovery catalog
Report of database schema for database with db_unique_name DEMO111
```

```
List of Permanent Datafiles
```

```
=====
File Size(MB) Tablespace          RB segs Datafile Name
-----
1    780      SYSTEM                YES      /u02/.../o1_mf_system_bf67nzw0_.dbf
3    600      SYSAUX                NO       /u02/.../o1_mf_sysaux_bf67mwl5_.dbf
4    255      UNDOTBS1              YES      /u02/.../o1_mf_undotbs1_bf67pgdj_.dbf
6     5       USERS                 NO       /u02/.../o1_mf_users_bf67pf5q_.dbf
```

```
List of Temporary Files
```

```
=====
File Size(MB) Tablespace          Maxsize(MB) Tempfile Name
-----
1    60       TEMP                    32767    /u02/.../o1_mf_temp_bf67qnwz_.tmp
```

La base de données source

Voici ma commande à exécuter (-h est pour le hostname, -ti correspond au terminal interactif):

```
docker@boot2docker:~$ docker run --name
«DEMO111_SBY» -ti -h «DEMO111_SBY» -p 9000:1521
breed85/oracle-12c:12102 su - oracle
```

J'ai configuré mon environnement. Le nom de l'instance sera DEMO111:

```
[oracle@DEMO111_SBY ~]$ export ORACLE_HOME=/
oracle/product/12102
[oracle@DEMO111_SBY ~]$ export ORACLE_
SID=DEMO111
```

et j'ai créé un password file identique à la base de donnée primaire:

```
[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/
orapwd file=$ORACLE_HOME/dbs/orapwDEMO111
password=oracle
```

J'ai besoin que le listener soit configuré en tant que service statique. Gardons cela au plus simple:

```
[oracle@DEMO111_SBY ~]$ cat > $ORACLE_HOME/net-
work/admin/listener.ora <<-CAT
LISTENER=(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=DEMO111_SBY) (PORT=1521)))
SID_LIST_LISTENER=(SID_LIST=(SID_DESC=(GLOBAL_
DBNAME=DEMO111_SBY_DGMGRL) (ORACLE_HOME=$ORACLE_
HOME) (SID_NAME=DEMO111)))
CAT
```

Et je le démarre:

```
[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/lsn-
rctl start
```

Voici le message de sortie où nous pouvons voir le point d'écoute et le service statique:

```
...
Listener Parameter File /oracle/product/12102/
network/admin/listener.ora
Listener Log File /oracle/diag/tnslsnr/
DEMO111_SBY/listener/alert/log.xml
Listening Endpoints Summary...
 (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
(HOST=DEMO111_SBY) (PORT=1521)))
Services Summary...
Service «DEMO111_SBY_DGMGRL» has 1 instance(s).
 Instance «DEMO111», status UNKNOWN, has 1
handler(s) for this service...
The command completed successfully
```

J'utiliserai OMF et vais donc créer les répertoires:

```
[oracle@DEMO111_SBY ~]$ mkdir /oracle/oradata /
oracle/recovery_area
```

J'ai besoin d'un init.ora pour démarrer mon instance:

```
[oracle@DEMO111_SBY ~]$ cat > /tmp/init.ora <<-
CAT
db_name='DEMO111'
compatible=12.1.0.2.
db_unique_name='DEMO111_SBY'
db_create_file_dest='/oracle/oradata'
db_recovery_file_dest_size='1G'
db_recovery_file_dest='/oracle/recovery_area'
dg_broker_start=true
CAT
```

Créez alors le spfile et démarrez l'instance:

```
[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/
sqlplus / as sysdba <<- 'SQL'
create spfile from pfile='/tmp/init.ora';
startup nomount;
SQL
```

Testons la connexion au réseau:

```
[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/tnsp-
ing «//192.168.78.111:1521/DEMO111»
...
Attempting to contact (DESCRIPTION=(CONNECT_
DATA=(SERVICE_NAME=DEMO111))
(ADDRESS=(PROTOCOL=TCP) (HOST=192.168.78.111)
(PORT=1521)))
OK (0 msec)

[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/tnsp-
ing «//192.168.59.104:9000/DEMO111_SBY_DGMGRL»
...
Attempting to contact (DESCRIPTION=(CONNECT_
DATA=(SERVICE_NAME=DEMO111_SBY_DGMGRL))
(ADDRESS=(PROTOCOL=TCP) (HOST=192.168.59.104)
(PORT=9000)))
OK (0 msec)
```

Tout est parfait. Comme je peux me connecter à la source ainsi qu'à la cible, faisons un duplicate from active avec rman. Une bonne pratique est de déclarer la chaîne de connexion dans le fichier tnsnames.ora mais faisons le rapidement avec la chaîne ezconnect (qui fonctionne dans la mesure où elle est inférieure à 64 caractères).

```
[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/rman

RMAN> connect target sys/ora-
cle@//192.168.78.111:1521/DEMO111;
connected to target database: DEMO111
(DBID=2551031863)

RMAN> connect auxiliary sys/ora-
cle@//192.168.59.104:9000/DEMO111_SBY_DGMGRL;
connected to auxiliary database: DEMO111 (not
mounted)
```

```

RMAN> duplicate database for standby from ac-
tive database;
Starting Duplicate...
...
Finished Duplicate Db...

```

```

breed85/oracle-12c DEMO111_SBY 31068a2689fd About a min ago
8.394GB
breed85/oracle-12c 12102 9cc040efd6e About an hour ago 6.264GB
breed85/oracle-12c latest 9d3b9aab053b 11 weeks ago 1.728GB
breed85/oracle-12c preinstall 9d3b9aab053b 11 weeks ago 1.728GB

```

Configurer la standby

J'utilise le broker DataGuard pour configurer rapidement la standby:

```

[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/dgmgrl
<<-DG
connect sys/oracle@»//192.168.78.111:1521/
DEMO111»
create configuration 'DEMO111' as primary da-
tabase is 'DEMO111' connect identifier is
'192.168.78.111:1521/DEMO111';
add database 'DEMO111_SBY' as connect identifier
is '//192.168.59.104:9000/DEMO111_SBY_DGMGRL ';
enable configuration;
edit database 'DEMO111_SBY' set state=apply-on;
DG

```

N'oubliez pas que le but est juste d'avoir une standby qui est synchronisée. Nous ne basculerons jamais vers la base du container Docker ! Voici l'écran de sortie: Ok, il suffit d'attendre un peu et il est synchronisé:

```

DGMGRL> connect /
Connected as SYSDBA.
DGMGRL> show database "DEMO111_SBY"
Database - DEMO111_SBY

Role:                PHYSICAL STANDBY
Intended State:      APPLY-ON
Transport Lag:       0 seconds (computed 20
seconds ago)
Apply Lag:           0 seconds (computed 20
seconds ago)
Average Apply Rate: 30.00 KByte/s
Real Time Query:    OFF
Instance(s):
DEMO111

```

Et c'est tout. Afin de conserver cette configuration, nous pouvons quitter le shell interactif et enregistrer cette image:

```

docker@boot2docker:~$ docker commit DEMO111_SBY
breed85/oracle-12c:DEMO111_SBY

```

Supprimer le container:

```

docker@boot2docker:~$ docker rm DEMO111_SBY

```

Et vérifier la nouvelle image qui a 2.13GB supplémentaire pour notre base de données:

```

docker@boot2docker:~$ docker images
REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE

```

Rafraîchir la standby

Notre but est d'avoir une standby rafraîchit en permanence, nous allons donc exécuter un container que nous allons laisser tourner. Je peux le mettre en arrière-plan, mais pour les tests, je vais le laisser tourner dans une fenêtre. C'est la même ligne de commande sauf que j'appelle le container 'refresh'.

```

docker@boot2docker:~$ docker run --name «re-
fresh» -ti -h «DEMO111_SBY» -p 9000:1521
breed85/oracle-12c:DEMO111_SBY su - oracle

```

Et je lance le minimum requis, démarrer le listener:

```

[oracle@DEMO111_SBY ~]$ export ORACLE_HOME=/
oracle/product/12102
[oracle@DEMO111_SBY ~]$ export ORACLE_
SID=DEMO111
[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/lsn-
rctl start

```

Et l'instance:

```

[oracle@DEMO111_SBY ~]$ $ORACLE_HOME/bin/sql-
plus / as sysdba
startup nomount;
alter database mount standby database;

```

Je démarre docker dans une autre fenêtre:

```

cd «C:\Program Files\Boot2Docker for Windows»
boot2docker ssh
docker@boot2docker:~$

```

Approvisionnement de la base de données virtuelle

J'ai les images suivantes:

```

docker@boot2docker:~$ docker images
REPOSITORY TAG IMAGE ID CREATED VIRTUAL SIZE
breed85/oracle-12c DEMO111_SBY 31068a2689fd 41minutes ago
8.394GB
breed85/oracle-12c 12102 9cc040efd6e 2hours ago 6.264 GB
breed85/oracle-12c latest 9d3b9aab053b 11weeks ago 1.728 GB
breed85/oracle-12c preinstall 9d3b9aab053b 11weeks ago 1.728GB

```

Les images étant construites les unes à partir des autres, je n'aurai pas besoin physiquement de la totalité de l'espace virtuel affiché ci-dessus.

Le 'refresh' container est en cours de fonctionnement.

```
docker@boot2docker:~$ docker ps -as
...IMAGE      COMMAND      PORTS      NAMES      SIZE
...12c:DEMO111_SBY «su - oracle» 0.0.0.0:9000->1521/tcp refresh
1.793 GB

docker@boot2docker:~$ docker ps -as
CONTAINER ID  IMAGE      COMMAND      CREATED
STATUS      PORTS      NAMES      SIZE
9f2d6e5333bb  breed85/oracle-12c:DEMO111_SBY «su - oracle»
20 minutes ago Up 20 minutes 0.0.0.0:9000->1521/tcp refresh
1.808 GB
```

Nous voyons la taille (qui est la taille ajoutée à l'image de base) et le port de redirection. Le statut est en cours.

Maintenant, je sauvegarde cet état dans une autre image:

```
docker@boot2docker:~$ docker commit refresh
breed85/oracle-12c:vdb1
```

J'ai donc une nouvelle image:

```
docker@boot2docker:~$ docker images
REPOSITORY TAG IMAGE ID CREATED VIRTUAL
SIZE
breed85/oracle-12c vdb1 be0ba1450a98 42seconds ago 10.2
GB
breed85/oracle-12c DEMO111_SBY 31068a2689fd 46minutes ago 8.394
GB
breed85/oracle-12c 12102 9cc040ef1d6e 2hours ago 6.264
GB
breed85/oracle-12c latest 9d3b9aab053b 11weeks ago 1.728
GB
breed85/oracle-12c preinstall 9d3b9aab053b 11weeks ago 1.728
GB
```

Vous voyez que j'ai maintenant un serveur de base de données virtuel supplémentaire qui fait 10.2GB (Système + logiciel Oracle + fichiers de données + fichiers d'archive) mais je n'ai besoin que de 1.79GB pour le stocker – ce qui correspond à la taille de la couche 'vdb1'.

Ce sont des tailles virtuelles. La taille physique réelle est visible sur le système de fichiers AUFS qui est utilisé par docker:

```
docker@boot2docker:~$ df -h /mnt/sda1/var/lib/docker/aufs
Filesystem Size Used Available Use%
Mounted on
/dev/sda1 18.2G 11.3G 5.9G 66%
/mnt/sda1
```

Quel est le coût d'approvisionnement d'une autre base de données virtuelle?

```
docker@boot2docker:~$ docker commit refresh
breed85/oracle-12c:vdb2
Filesystem Size Used Available Use%
Mounted on
/dev/sda1 18.2G 13.0G 4.2G 76%
/mnt/sda1
```

```
docker@boot2docker:~$ docker images
REPOSITORY TAG IMAGE ID CREATED
VIRTUAL SIZE
breed85/oracle-12c vdb2 61d372cc903d About a minute ago
10.2 GB
breed85/oracle-12c vdb1 be0ba1450a98 6minutes ago
10.2 GB
breed85/oracle-12c DEMO111_SBY 31068a2689fd 51minutes ago
8.394 GB
breed85/oracle-12c 12102 9cc040ef1d6e 2hours ago
6.264 GB
breed85/oracle-12c latest 9d3b9aab053b 11weeks ago
1.728 GB
breed85/oracle-12c preinstall 9d3b9aab053b 11weeks ago
1.728 GB
```

La nouvelle base de données utilise physiquement 1.7GB mais nous pouvons voir une copie virtuelle de 10.2GB.

Ouverture de la base de données virtuelle

Alors, de quoi avons-nous besoin pour utiliser la nouvelle base de données virtuelle ? Il suffit juste de lancer l'image dans un nouveau container et de rediriger le port 1521 vers un autre:

```
docker@boot2docker:~$ docker run --name «vdb1»
-ti --rm -h «DEMO111_vdb1» -h «vdb1» -p
9001:1521 breed85/oracle-12c:vdb1 su - oracle
```

J'ai ajouté l'option '-rm' de sorte que le container est supprimé quand je quitte le shell. Je le fais quand je veux juste l'utiliser pour une courte période de temps.

Je vais être capable d'y accéder en utilisant le port redirigé 9001 de sorte que la chaîne de connexion sera: //192.169.59.104:9001/DEMO111_SBY mais pour le moment, je dois démarrer le listener. Notez que j'ai supprimé le listener.ora qui était configuré pour un autre nom d'hôte:

```
[oracle@vdb1 ~]$ export ORACLE_HOME=/oracle/
product/12102
[oracle@vdb1 ~]$ export ORACLE_SID=DEMO111
[oracle@vdb1 ~]$ rm $ORACLE_HOME/network/admin/
listener.ora
[oracle@vdb1 ~]$ $ORACLE_HOME/bin/lsnrctl start
```

Puis démarrer l'instance:

```
[oracle@vdb1 ~]$ $ORACLE_HOME/bin/sqlplus / as
sysdba
startup nomount;
alter database mount standby database;
```

Et activer la standby pour qu'elle devienne la primaire afin de l'ouvrir en lecture/écriture, ce qui est appelé un failover.

```
[oracle@vdb1 ~]$ $ORACLE_HOME/bin/dgmgml /
failover to «DEMO111_SBY» immediate;
```

...


```
Performing failover NOW, please wait...
Failover succeeded, new primary is «DEMO111_
SBY»
```

Bien sûr, ce serait une bonne idée d'isoler le container lors du démarrage afin d'être sûr que la nouvelle primaire n'essaie pas de communiquer avec les bases primaire et standby actuelles. Nous pouvons changer le nom de la base de données mais le container peut fournir l'isolation dont on a besoin. Je ne veux qu'il y ait qu'une seule chose qui puisse passer à travers: la redirection de port 9001 -> 1521. Le résultat est que j'ai maintenant deux containers qui tournent:

IMAGE	CREATED	STATUS	PORTS	NAMES	SIZE
12c:vdb1	9 minut	Up 9 m	...:9001->1521/tcp	vdb1	2.128 GB
12c:DEMO111_SBY	About a	Up Abo	...:9000->1521/tcp	refresh	1.813 GB

J'ai fait quelques mises à jour sur 'vdb1', c'est pourquoi la taille a augmenté et un 'refresh' applique toujours les redo depuis la primaire.

Je peux exécuter autant de base de données virtuelle que je veux. Elles sont gérées comme des containers Docker. Et si vous voulez retrouver les informations concernant les ressources, les voici:

```
Mem: 1889632K used, 166604K free, 0K shrd, 376K
buff, 376K cached
CPU: 0.1% usr 0.0% sys 0.0% nic 99.8% idle
0.0% io 0.0% irq 0.0% sirq
Load average: 0.20 0.32 0.41 2/411 8023
...
 5960  5908  54321   S    662m 32.8   5  0.0
{ora_pmon_demo11} ora_pmon_DEMO111
...
 7691  7631  54321   S    662m 32.8   1  0.0
{ora_pmon_demo11} ora_pmon_DEMO111
...
```

Toutes les instances sont en train de tourner (vous voyez de multiples pmon). Elles tournent toutes sur le même serveur mais isolées dans des containers.

Se connecter à une base de données virtuelle

Je peux me connecter à ma vdb1 depuis n'importe quel client qui a accès à la VM docker à travers le réseau:

```
SQL> connect system/ora-
cle@//192.168.59.104:9001/DEMO111_SBY
```

Et vérifier depuis quel host il est exécuté:

```
SQL> select host_name from v$instance;
HOST_NAME
-----
DEMO111_vdb1
```

Qui est le nom d'hôte virtuel passé à la commande 'run' à travers l'argument '-h'.

Conclusion

Quel est le but d'écrire un article sur quelque chose que l'on ne peut pas faire en production? Je voulais montrer les concepts de la virtualisation des données et j'aime apprendre par la pratique. Toutes les technologies sous-jacentes (système de fichier copy-on-write, virtualisation, isolation) sont là et vous pouvez les utiliser avec des outils simples tels que Docker pour pratiquer sur votre PC portable.

Essayez-le. Téléchargez Docker. Téléchargez les fichiers zip d'Oracle. Et faites la même chose, ce qui est probablement aussi simple que de copier-coller les commandes et de remplacer seulement les adresses IP.

Bien sûr, pour votre production, vous devrez aller vers des logiciels robustes et supportés et de nombreux fournisseurs ont des bonnes solutions pour la virtualisation des données, les snapshots et les clones. Mais je suis sûr qu'une fois que vous aurez pratiqué avec ce do-it-yourself qui encore une fois, ne concerne pas une solution officiellement supportée, alors vous aurez de meilleures bases pour choisir et comparer les solutions commerciales.



Franck Pachot
franck.pachot@dbi-services.com

Franck Pachot est consultant senior chez dbi services en Suisse. Il a plus de 20 ans d'expérience dans les bases de données Oracle, dans la plupart des domaines, depuis le développement, la modélisation des données, la performance, l'administration et la formation. Oracle Certified Master et Oracle ACE, il essaye de partager ses connaissances dans les forums, via des publications et des présentations.



en français:
www.doag.org/go/News/201503/Pachot_fr



en anglais:
www.doag.org/go/News/201503/Pachot



27ème rapport annuel du président (2014)

2014, une année associative riche en événements et en réussites, touche à sa fin. Nous avons lancé de nombreux projets, afin de proposer un service plus moderne à nos membres dans les années à venir, et pérenniser l'image avantageuse de notre association. Découvrez ci-dessous les informations à propos des questions les plus importantes de l'année associative passée :

Finances

L'exercice comptable 2014 se clôture avec un bénéfice de 1 801,44 CHF. Vous en trouverez le détail dans le bilan et les comptes de résultats correspondants.

Les biens de l'association, d'une valeur de presque 100 000 CHF, ont été conservés cette année. Nous disposons donc d'une base financière suffisante pour traiter en 2015 des questions qui nous préoccupent depuis déjà longtemps. Nous prévoyons d'optimiser le service de traduction pour nos membres français, d'améliorer les capacités du département de comptabilité et d'apporter des améliorations générales à l'offre proposée aux membres. Malgré toutes ces mesures, nous pensons que nous pourrions diminuer durablement la contribution annuelle des membres de presque 10 % pour 2015 grâce à notre solide base financière et aux économies réalisées. À court terme, cette mesure provoque un effet légèrement négatif sur le budget 2015, qui devrait être compensé par l'efficacité des optimisations prévues dans les années à venir.



Kurt Meier

Évènements

En 2014, nous avons conduit huit événements : Trois rencontres du « Special Interest Groups » en Suisse alémanique, deux en Romandie, une « SOUG bi dä Lüt », le « Real World Performance Tour » à Munich avec la DOAG et l'AOUG, ainsi que la « Conférence annuelle DOAG » à Nuremberg.

Une moyenne de 31 participants à chacun de ces événements montre que ceux-ci sont toujours appréciés, et constituent une occasion favorable aux échanges sur l'expérience. Même le nouvel événement « Real World Performance Tour » a suscité des réactions positives chez de nombreux participants suisses.

La « Conférence annuelle DOAG » à Nuremberg fait partie intégrante de notre programme de l'année. En 2014, nous avons bénéficié d'un nouvel emplacement pour notre stand, grâce auquel nous avons pu attirer beaucoup plus de visiteurs. Avec le réseautage, nous avons même pu accueillir quelques nouveaux membres. Cette année, à la « Conférence annuelle DOAG », les participants suisses ont été particulièrement actifs à travers leurs conférences. La grande proportion de participants suisses s'est également constatée au cours de la soirée suisse. Plus de 40 participants se sont retrouvés pour une visite de la ville, malheureusement un peu humide, qui s'est terminée par une soirée conviviale dans un établissement local typique

de Nuremberg. La convivialité fait aussi partie de la vie associative, et doit être d'autant plus entretenue dans notre monde numérisé.

Cette année, la SOUG participera de nouveau au « Real World Performance Tour » et à une « Conférence annuelle DOAG ». Le « Real World Performance Tour » se tiendra le 25 mars 2015 à Vienne. Tous ceux qui souhaitent s'y inscrire sont les bienvenus.

Pensez à réserver vos journées du 17 au 20 novembre 2015 pour la « Conférence annuelle DOAG » à Nuremberg, et bien entendu pour la soirée suisse ! Je serai ravi de vous y rencontrer en personne.

Lettre d'information

En 2014, 5 lettres d'information sont parues, un rythme de parution tout à fait habituel. Dès 2015, nous partagerons la lettre d'information avec la DOAG. Vous avez déjà dû recevoir une première version de cette nouvelle édition.

Une collaboration étroite dans ce domaine avec la DOAG nous permet de publier une 6ème édition par an, avec un contenu plus fourni. Cette collaboration est coordonnée par un journaliste professionnel. En outre, la DOAG dispose toujours d'une lettre d'information Business. Comme attendu, l'effet de synergie nous fait bénéficier de réduction de dépenses, malgré une édition plus importante.

Une dernière précision au sujet du retard de livraison survenu pour la première lettre d'information de 2015 : nous avons pu identifier le problème. Ainsi, vous recevrez la prochaine lettre d'information dans les temps. Nous nous excusons pour le dérangement.

Membres de l'association

L'année 2014 se clôt sur un nombre total de 401 membres, soit 3 % de moins que l'année précédente. Le comité directeur est cependant certain que le développement déjà en marche des questions concernant Oracle traitées par la SOUG, le développement portant sur la lettre d'information ainsi que la réduction de la contribution annuelle changeront la tendance et encourageront l'augmentation du nombre de membres.

Relations extérieures

Le travail en collaboration avec d'autres User Groups (en particulier avec la DOAG) fonctionne correctement. Les relations se resserrent, et, lorsqu'il est possible, des synergies sont mises en place. Nous travaillerons activement au développement de ces interactions, afin que nos membres puissent bénéficier des connaissances de notre personnel international qualifié.

Comité directeur

En raison du changement professionnel vers Oracle, Ansgar Wollnik ne peut plus conserver son poste au comité directeur, comme le stipule le règlement de l'association. Au même titre, Hansjörg Bütler, après plusieurs années au sein

du comité directeur, a décidé d'employer son temps libre en dehors des questions liées à Oracle. Je remercie chaleureusement ces deux hommes pour l'engagement dont ils ont fait preuve au sein de l'association. Ils ont grandement contribué à notre réussite actuelle.



Merci à Ansgar Wollnik (centre) et Hansjörg Bütler

Heureusement, nous leur avons trouvé des successeurs compétents et engagés : Patrick Schäfer et Jürgen Vitek. Patrick Schäfer travaille à la mise en place d'une nouvelle section pour l'évènementiel, portée sur l'architecture, les applications et le middleware; attendez-vous à de grandes surprises! Jürgen Vitek reprendra le flambeau de



la coordination des évènements, suite au départ de Hansjörg Bütler. Nous prévoyons en outre d'agrandir le comité directeur d'une personne au cours de l'année 2016. Notre objectif est de répartir une quantité de travail parfois importante entre plusieurs personnes, tout en conservant suffisamment de temps pour traiter les questions d'avenir au comité directeur.



Patrick Schäfer

Ainsi, je nous souhaite à tous une année associative riche et réussie pour 2015.



Peter Gübeli
Président SOUG

Oracle RAC, Oracle Data Guard, and Pluggable Databases: When MAA Meets Oracle Multitenant

Ludovico Caldara, Trivadis AG

Ludovico Caldara de Trivadis a écrit un bon article sur le fonctionnement des pluggable databases dans une architecture à disponibilité maximale (MAA) et comment créer les PDB et les services. L'article a été publié en allemand dans le magazine DOAG/SOUG News. Voici le lien pour la version "internationale" en anglais:

www.doag.org/go/News/201503/Caldara



Corrigenda

Dans la publicité pour le SIG Tuning du 21.05 à Prangins, nous avons accidentellement "migré" Christian Antognini chez un autre employeur. Christian Antognini est bien Principal

Senior Consultant et formateur chez Trivadis. Il dirige également un blog intéressant: www.antognini.ch. Nous nous excusons auprès de Christian et de Trivadis.

Newbies CH

*Richard Koller, CSS Versicherung
Laszlo Hadhazy, Accarda AG
Christina Pavalache, Diso AG
Karl-Heinz Sütterlin, Microsoft Schweiz GmbH
Jacques Kostic, Trivadis AG
Thomas Rieder, Atos AG
Dr. Stefan Meyer, Prolicense Schweiz GmbH
Tobias Schaller, Zürich Kantonalbank
Marco Kurmann, bbi software ag*

Mentions légales

secrétariat SOUG:

Dornachstrasse 192, 4053 Basel
Tel.: 061 367 93 30, Fax: 061 367 93 31
sekretariat@soug.ch

rédaction:

Geatano Bisaz
gaetano.bisaz@soug.ch

réalisation / DTP:

DOAG Dienstleistungen GmbH
Tempelhofer Weg 64, 12347 Berlin
office@doag.org

impression:

Druckerei Rindt GmbH & Co. KG
www.rindt-druck.de

rédaction Newsletter:

nl@soug.ch

inscription aux événements SOUG:

event@soug.ch

questions des membres:

sekretariat@soug.ch

site du web:

www.soug.ch

Photo couverture: © lightwise / 123RF.com