

Oracle, Ô désespoir

Dans le monde de l'entreprise, le SGBD d'Oracle est assez répandu. À l'attention de ceux qui ne connaissent pas Oracle : c'est l'équivalent d'Access en un peu plus puissant. En fait en beaucoup plus puissant, c'est comme comparer une voiture à pédales avec une formule un.

Bonne nouvelle, Oracle est compatible Linux ; mauvaise nouvelle, il n'est compatible qu'avec certaine version de Linux (RedHat et SUSE notamment). Donc, le plus simple pour installer ce SGBD est d'installer une de ses distributions ou, tout simplement, d'utiliser un autre SGBD tel que MySQL ou PostgreSQL. Vous voulez conserver votre Debian et en plus, vous voulez Oracle ! Ok, voyons ce que nous pouvons faire pour vous. En cherchant bien, il s'avère que certaines solutions existent pour installer Oracle sur votre serveur.

Quelques pré-requis

Avant de se lancer comme des fous dans une installation qui se révèle un peu délicate, vérifions votre paquetage. Pour installer Oracle, voici ce dont vous aurez besoin :

- Un système Debian, ça tombe bien c'est le sujet de ce magazine.
- Une interface graphique quelle qu'elle soit (GNOME, KDE, Xfce, fluxbox...). Nous appuierons nos explications sur une installation standard de Debian et donc, avec l'interface graphique GNOME.
- Au minimum 1 Go de RAM.

Il faut également que votre swap soit correctement dimensionné. Pourquoi ? Tout simplement parce que Oracle vérifie ce paramètre avant de commencer son installa-

tion. La taille du swap dépend de la taille de votre RAM :

- RAM inférieur à 2 Go : le swap doit être au minimum égal à 1,5 x RAM.
- RAM entre 2 et 8 Go : le swap doit être au minimum égal à 1 x RAM.



Figure 1. Le SGBD Oracle est extrêmement robuste

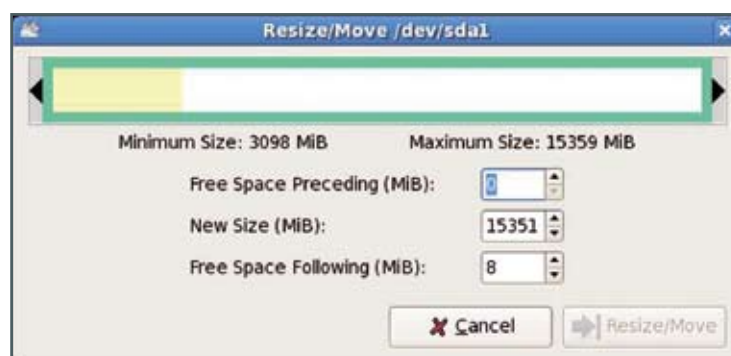


Figure 2. Avec GParted, il est possible de re-dimensionner vos partitions



Figure 3. Le véritable répertoire home d'Oracle se trouve à Redwood Shores

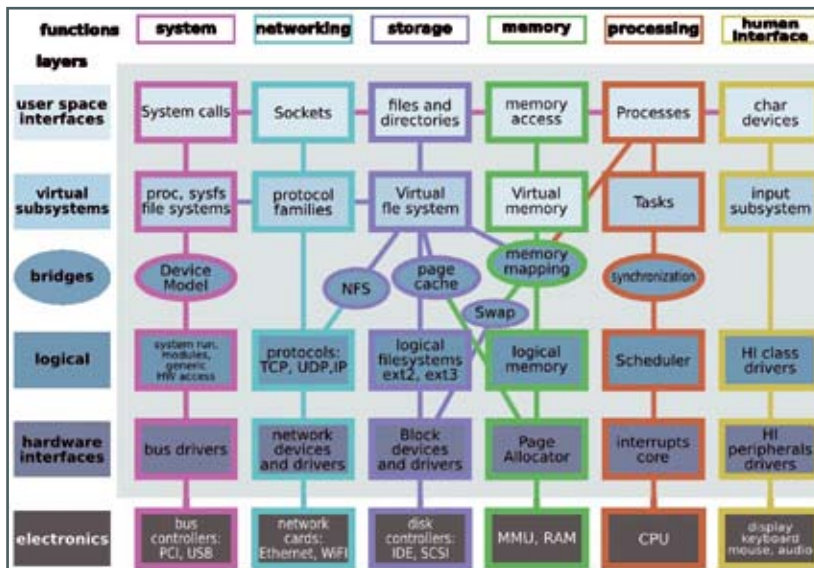


Figure 4. Une explication simplifiée du fonctionnement du kernel

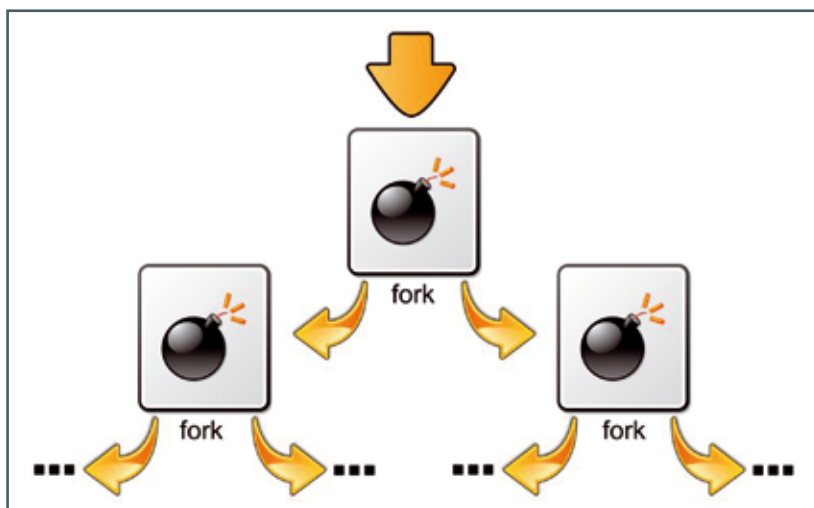


Figure 5. Évitez les Fork Bomb en fixant des limites

- RAM supérieur à 8 Go : le swap doit être au minimum égal à 0,75 x RAM.

Si votre swap n'est pas correct, vous pouvez toujours redimensionner vos partitions : l'excellent GParted devrait vous simplifier la vie.

Concernant les processeurs de votre machine, pas de restriction particulière, il suffit juste de télécharger la bonne version d'Oracle qui correspond à votre architecture.

Dernier petit conseil, pour un logiciel tel que Oracle, il est bon d'avoir une partition dédiée : /oracle par exemple mais ceci n'est pas une obligation.

Les copains d'abord

Cette fois, nous y allons, nous pouvons commencer à manipuler. Première des choses à faire : préparer un nid douillet pour notre Oracle. Et oui, même si c'est un SGBD robuste, il a besoin de confort, il a ses habitudes, il ne faut pas le brusquer : Oracle est un peu susceptible lors de l'installation.

Nous allons créer les groupes et l'utilisateur nécessaires à Oracle ; bien entendu, les noms peuvent changer mais nous vous conseillons toutefois de conserver ceux qui suivent pour simplifier la suite de l'installation. Par ailleurs, bon nombre d'aides sur Internet utilisent ces noms, alors pourquoi s'embêter ? Pour créer ces groupes et cet utilisateur, connectez-vous sous le profil de l'utilisateur root (su root par exemple) et tapez ceci :

```
groupadd oinstall
groupadd dba
useradd oracle -m -d /oracle \
-g oinstall -G dba -s /bin/ \
bash
```

Votre utilisateur oracle est désormais créé, Il ne reste plus qu'à lui attribuer un mot de passe :

```
passwd oracle
```

Choisissez votre mot de passe en respectant les précautions d'usage, surtout si votre serveur est accessible en SSH.

Voilà, vous avez un petit Oracle qui a un répertoire home bien vide mais ne vous inquiétez pas, nous allons vite le remplir.

Kernel et autres sémaphores

Cette fois, nous passons à des choses plus pointues : configurer le kernel et les sémaphores. Que recouvrent ces termes compliqués ? Il s'agit juste de valeurs limites à attribuer à notre noyau, comme l'explique la documentation d'Oracle. En voici un résumé direct et concis pour vous simplifier la vie. Toujours en ligne de commande, lancez les commandes présentées dans le Script 1.

Pour prendre en compte directement ces modifications, redémarrez le serveur par la commande `sysctl -p /etc/sysctl.conf`. L'affichage des valeurs précédemment renseignées confirmera cette prise en compte.

Dépasser les limites

Pour éviter à Oracle de dépasser les limites, nous allons paramétrer le fichier `/etc/security/limits.conf`. Comme son nom l'indique, il s'agit de sécurité ; ce fichier est utilisé, par exemple, pour limiter le nombre de processus et ainsi se préserver des Fork Bomb (ou logical bomb).

Comme toujours, ceci ce passe en ligne de commande, lancez ces instructions configurer le fichier :

```
echo "oracle soft nproc 2047" \
  >> /etc/security/limits. \
  conf
echo "oracle hard nproc 16384" \
  >> /etc/security/limits. \
  conf
echo "oracle soft nofile 1024" \
  >> /etc/security/limits. \
  conf
echo "oracle hard nofile \
  65536" >> /etc/security/ \
  limits.conf
```

Ces valeurs seront prises en compte lors du prochain redémarrage de la machine ; en attendant, il n'y a pas de limite, ce qui n'est pas très prudent mais cette situation ne dérange pas l'installation d'Oracle.

Script 1. Configuration du kernel et des sémaphores

```
echo "kernel.sem = 250 32000 100 128" >> /etc/sysctl.conf
echo "kernel.shmall = 2097152" >> /etc/sysctl.conf
echo "kernel.shmmax = 1200000000" >> /etc/sysctl.conf
echo "kernel.shmmni = 4096" >> /etc/sysctl.conf
echo "fs.file-max = 6815744" >> /etc/sysctl.conf
echo "net.ipv4.ip_local_port_range = 9000 65000" >> /etc/ \
  sysctl.conf
echo "net.core.rmem_default = 262144" >> /etc/sysctl.conf
echo "net.core.rmem_default = 262144" >> /etc/sysctl.conf
echo "net.core.rmem_max = 4194304" >> /etc/sysctl.conf
echo "net.core.wmem_default = 262144" >> /etc/sysctl.conf
echo "net.core.wmem_max = 1048576" >> /etc/sysctl.conf
echo "fs.aio-max-nr = 1048576" >> /etc/sysctl.conf
```

Script 2. Paramétrage de l'environnement Oracle

```
echo 'umask 022' >> /oracle/.bashrc
echo 'unset USERNAME' >> /oracle/.bashrc
echo 'export ORACLE_BASE=/oracle/app/' >> /oracle/.bashrc
echo 'export ORACLE_HOME=/oracle/app/11g' >> /oracle/.bashrc
echo '#bdtest étant le nom de votre base de donnée' >> / \
  oracle/.bashrc
echo 'export ORACLE_SID=bdtest' >> /oracle/.bashrc
echo 'export ORACLE_TERM=xterm' >> /oracle/.bashrc
echo 'export PATH=$ORACLE_HOME/bin:/usr/sbin:$PATH' >> / \
  oracle/.bashrc
echo 'export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib' \
  >> /oracle/.bashrc
echo 'export CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/ \
  jlib:$ORACLE_HOME/rdbms/jlib' >> /oracle/.bashrc
echo 'export ORATAB=/etc/oratab' >> /oracle/.bashrc
echo 'export ORACLE_HOME_LISTENER=$ORACLE_BASE' >> /oracle. \
  bashrc
./oracle/.bashrc
```

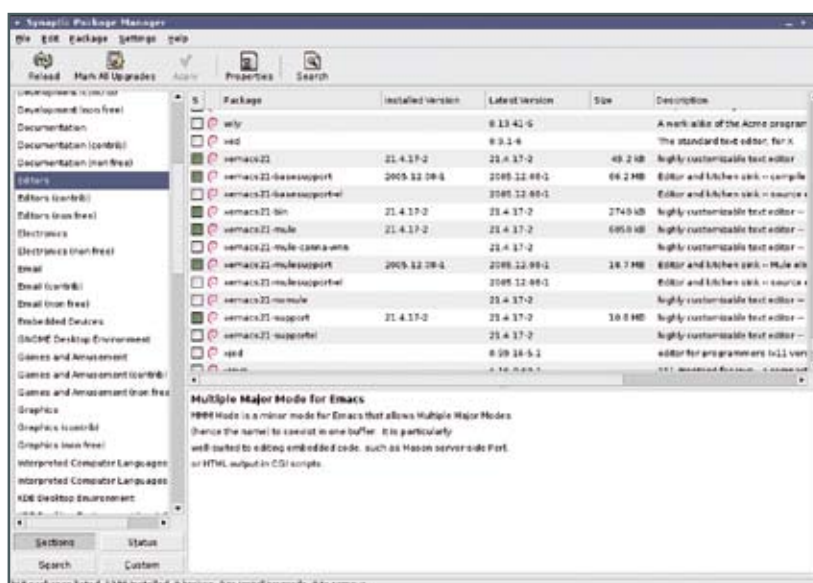


Figure 6. L'installation de paquetages se fait aussi avec Synaptic

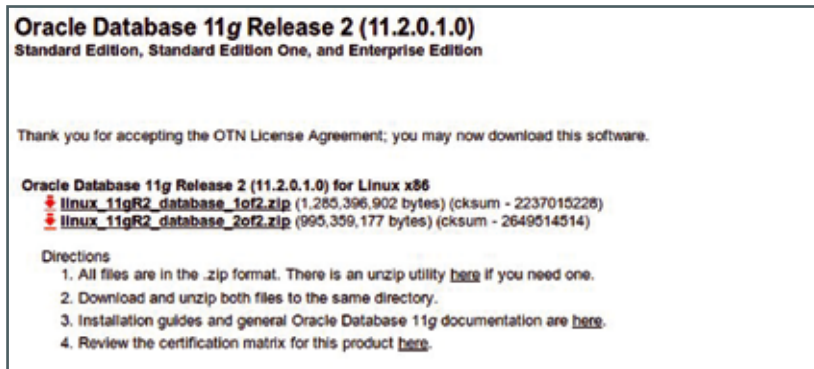


Figure 7. Notez bien la valeur des sommes de contrôle lors du téléchargement

Toujours au sujet des limites, ajoutons quelques lignes dans le fichier `/etc/profile` pour parachever cette configuration :

```
echo 'if [ $USER = "oracle" ]' \
  >> /etc/profile
echo 'then' >> /etc/profile
echo '    ulimit -u 16384 -n \
  65536' >> /etc/profile
echo 'fi' >> /etc/profile
```

Vous avez certainement remarqué que toutes ces lignes de commande contiennent le mot « oracle », il s'agit seulement d'une référence à l'utilisateur Oracle. Si vous avez choisi un autre nom d'utilisateur, vous savez ce qu'il vous reste à faire.

Quelques outils

Maintenant que vous connaissez Debian, vous savez que votre meilleur ami est `apt-get`. Vous l'avez compris, il faudra ajouter quelques paquets indispensables pour l'installation d'Oracle. Il s'agit surtout d'outils et de bibliothèques de compilation. Voici la ligne de commande à lancer :

```
apt-get install build-essential binutils make \
  autoconf automake unzip \
  bzip2 gcc less libc6-dev \
  libstdc++5 zlibc libaio1 \
  curl libelf-dev libaio-dev \
  ksh rpm lsb-rpm gcc- \
  multilib unixodbc-dev gawk \
  ksh libmotif4 alien
```

Pour éviter tout problème de dépendances, notamment dû à une mauvaise recherche d'Oracle, nous allons créer quelques liens symboliques pour que le pauvre petit SGBD ne soit pas trop perdu :

```
ln -s /usr/bin/awk /bin/awk
ln -s /usr/bin/basename /bin/ \
  basename
ln -s /usr/bin/rpm /bin/rpm
mkdir /etc/rc.d
for i in 0 1 2 3 4 5 6 S ; \
do ln -s /etc/rc$i.d /etc/ \
  rc.d/rc$i.d ; done
```

Nous allons initialiser la base de données RPM. Oui, oui, vous avez bien lu, nous parlons bien de RPM



Figure 8. Le premier écran de l'installateur Oracle

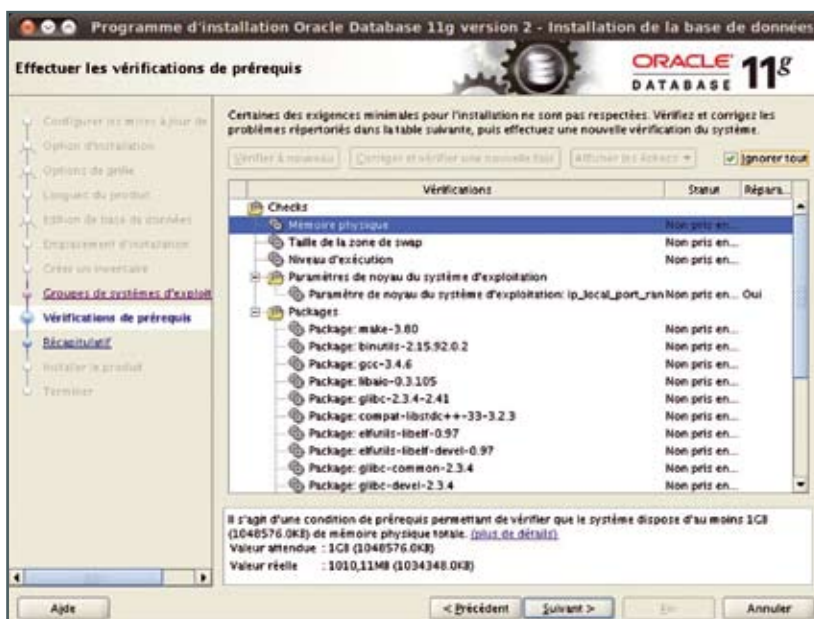


Figure 9. Les pré-requis en erreur qu'il faut ignorer...

sur un système Debian. Il s'agit juste d'une astuce pour « leurrer » Oracle. Pour initialiser la base RPM, rien de plus simple :

```
rpm --initdb
```

Voilà, c'est tout en ce qui concerne les paquetages supplémentaires, avouez qu'il ne faut pas grand chose pour installer Oracle.

Dernier ajustement

Avant de réellement installer Oracle, nous allons créer quelques répertoires supplémentaires qui nous seront utiles par la suite. Les manipulations ne sont plus à faire sous le profil du super-utilisateur root mais sous celui de l'utilisateur oracle.

Loggez-vous avec Oracle et lancez ces commandes :

```
mkdir -p /oracle/app
mkdir -p /oracle/oradata
mkdir -p /oracle/flash_ recovery_area
mkdir -p /oracle/sources/
chmod -R 775 /oracle/*
```

Paramétrons également le profil de l'utilisateur Oracle pour inclure quelques variables propres à ce petit SGBD. Vous avez l'habitude maintenant, nous continuons en lignes de commande (Script 2).

Si vous n'avez pas encore téléchargé Oracle database 11g (disponible à cette adresse : <http://http://www.oracle.com/technetwork/database/enterprise-edition/downloads/>), c'est le moment. Placez le ou les zip dans le répertoire `/oracle/sources/`. Cliquez sur See All pour obtenir les `cksum` de chaque téléchargement. Cette valeur vous permettra de vérifier l'intégrité de votre téléchargement. Pour utiliser `cksum`, rien de plus simple, lancez ces commandes (à adapter selon votre téléchargement) :

```
cksum linux_11gR2_ database_1of2.zip
cksum linux_11gR2_ database_2of2.zip
```

C'est un simple contrôle mais il vous fera gagner du temps en cas d'erreur incompréhensible d'instal-

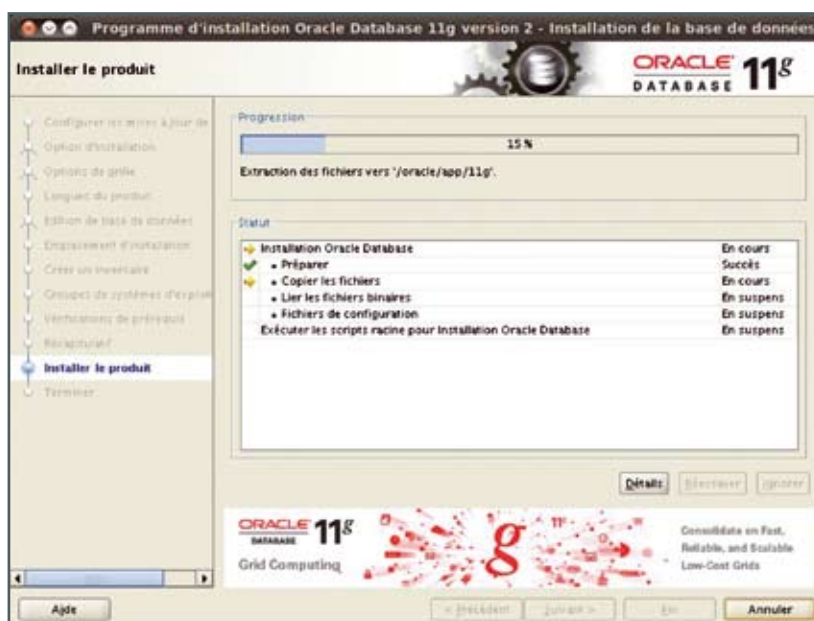


Figure 10. L'installation prend un certain temps

lation. Si tout se déroule bien, décompressez les archives :

```
cd /oracle/sources
unzip linux_11gR2_ database_1of2.zip
unzip linux_11gR2_ database_2of2.zip
```

Normalement, vous devriez obtenir un répertoire « database ». Cette fois c'est bon, tout est prêt, nous allons passer à l'installation proprement dite. Calmez vous, ne stressiez pas, ça va bien se passer ;)

Le grand saut

Nous y sommes, c'est le moment de voir si tout ce que vous avez fait est correct, grâce à cette commande :

```
cd /oracle/sources/database/
./runInstaller -ignoreSysPrereqs
```

L'installation n'est pas très compliquée, il suffit juste de suivre les écrans. Première question, certainement la plus compliquée, vous allez peut-être même rester bloqué : l'installateur vous demande votre mail ! Renseignez votre adresse mail.

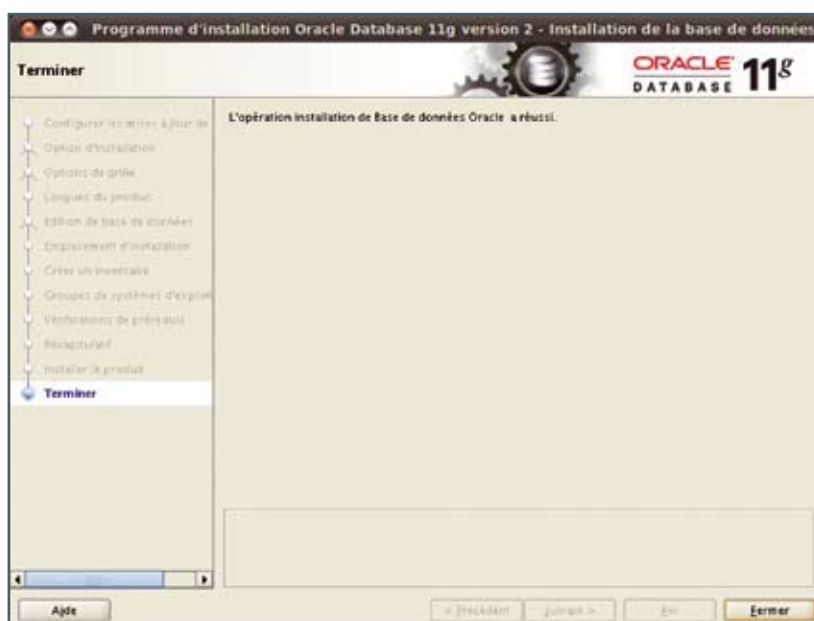


Figure 11. Installation terminée !



Figure 12. L'assistant de création de base de donnée

Ensuite, trois choix vous sont offerts ; pour cet article, nous avons choisi *Installer le logiciel de base de données uniquement*. Nous verrons comment créer une base de données un peu plus tard, mais pour gagner du temps, sélectionnez la première option (créer et configurer une base de données).

Écran suivant, ne cherchez pas à comprendre, sélectionnez *Installation d'une base de données mono-instance*. Si vous avez besoin d'utiliser l'autre choix, c'est que vous êtes plutôt avancé dans la maîtrise d'Oracle et donc, cet article ne vous sert à rien ;)

Choisissez votre langue. Le choix du type d'installation dépend de ce que vous souhaitez faire de votre Oracle, n'entrons pas dans les détails. Pour cet article, nous avons sélectionné *Enterprise Edition*, qui est un choix assez complet en terme de composant installé.

Ensuite, les deux écrans suivants portent sur une question de répertoire, ne modifiez rien, c'est ce que nous avons précédemment configuré dans le fichier `/oracle/.bashrc`. Idem pour l'écran d'après, les groupes par défaut sont bons, n'y touchez pas sauf si vous n'avez pas respecté les consignes relatives aux noms.

Enfin, Oracle vérifie tous les pré-requis et... ils sont tous en échec, c'est magnifique. La solution pour remédier à ce problème est encore plus belle : il suffit de cocher la case *Ignorer tout* pour que tous vos problèmes s'envolent, vous pouvez continuer.

Petit résumé de vos choix, validez et c'est parti, l'installateur travaille enfin. Patientez jusqu'à ce qu'une fenêtre vous demande d'exécuter deux scripts avec les droits root. Pour cela, ouvrez une console et lancez ces commandes :

```
su root
/oracle/app/11g/root.sh
/oracle/oraInventory/
oraInstRoot.sh
```

Pour toutes les questions, laissez les réponses par défaut. Même si l'installateur ne vous demande rien, exécutez ces deux scripts.

Et voilà, c'est terminé ! Ce n'était pas si compliqué, n'est-ce pas ?

Et ma base ?

Votre Oracle est installé mais ça ne vous suffit pas, maintenant vous voulez une base de données. Il est vrai qu'un SGBD sans base de données n'est pas très utile...

Dans un premier temps, nous allons créer un *listener*, c'est un fichier de paramètres essentiels qui permettra aux applications de se connecter à Oracle. Créez le fichier `/oracle/app/11g/network/admin/listener.ora` et complétez comme présenté dans le Script 3.

Dans ce fichier, rien de particulier n'est à signaler sauf « *bdtest* » qui correspond au nom de votre future base de données et 192.168.1.123 qui correspond à l'adresse IP de votre serveur.

Plusieurs méthodes permettent de créer une base de données, nous pouvons même la créer en lignes de commande mais cela requiert une certaine expérience d'Oracle et de son architecture (controlfile, tablespaces, redolog...). Nous n'allons pas nous compliquer la vie, nous allons utiliser l'interface graphique fournie par Oracle. Ouvrez-la grâce à cette commande : `dbca`



Figure 13. C'est ici que vous renseignez le nom de votre base

Cette commande est l'acronyme de « DataBase Création Assistant ». Une fenêtre s'ouvre vous souhaitant la bienvenue, cliquez sans attendre sur *Suivant*. Sélectionnez ensuite *Créer une base de données*. L'écran suivant vous propose des types de base à mettre en place, si vous ne savez pas quoi choisir, prenez l'option d'une base généraliste, ce qui devrait couvrir la plupart de vos besoins.

Ensuite, nommez votre base, indiquez le même nom dans les deux champs (SID et global de base de donnée), pour éviter tout problème.

Dans l'écran suivant de l'installateur, vous pouvez décocher *Configure Enterprise Manager*, selon votre façon de travailler. Une base n'a pas besoin de cet élément de toute manière et il sera possible de le remettre en place ultérieurement.

Pour les mots de passe, nous vous conseillons de mettre le même pour l'utilisateur `sys` et l'utilisateur `system`, ces deux utilisateurs ont à peu près les mêmes pouvoirs et seul un `dba` devrait y avoir accès.

Laissez par défaut le choix du type de stockage. En revanche, dans l'écran suivant, indiquez correctement le chemin `/oracle/flash_recovery_area` pour la zone de récupération rapide.

Pour le reste, laissez tout par défaut et validez les écrans les uns après les autres jusqu'à ce que l'installation vous renvoie un écran vous confirmant que tout s'est bien déroulé.

C'est terminé, vous avez une base Oracle sur votre Debian !

Dernier Test

Nous n'allons pas nous quitter si vite, vérifions que tout fonctionne bien. Dans un premier, démarrons tous les éléments dont nous aurons besoin :

```
dbstart $ORACLE_HOME
```

Retenez bien cette commande, il faudra la lancer à chaque redémarrage de votre serveur. D'ailleurs,

Script 3. Paramétrage du listener

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(SID_NAME = plsextproc)
(ORACLE_HOME = /oracle/app/11g)
(PROGRAM = extproc)
)
(SID_DESC =
(GLOBAL_DBNAME = bdtest)
(ORACLE_HOME = /oracle/app/11g)
(SID_NAME = bdtest)
)
)
)
LISTENER =
(DESCRIPTION_LIST =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = ipc)(KEY = extproc))
(ADDRESS = (PROTOCOL = tcp)(HOST = 192.168.1.123)(PORT = 1521))
)
)
)
ADR_BASE_LISTENER = /oracle/app
```

pour arrêter Oracle et tous ses composants, la commande est presque identique :

```
dbshut $ORACLE_HOME
```

Le test maintenant, il s'agit en fait de « pinguer » la base ou plutôt, de vérifier qu'elle est accessible :

```
tnsping bdtest
```

Le nom « bdtest » étant bien entendu le nom de votre base. Si cette commande se finit par *OK (xx ms)*, c'est une réussite. Votre base répond, c'est déjà encourageant.

Dernier petit test pour vérifier que votre base est effectivement en fonction (monté et ouvert) ; cette fois, nous utiliserons du `sq` et véri-

fierons un paramètre dans une vue dynamique d'Oracle :

```
sqlplus / as sysdba
SELECT open_mode FROM v$database;
```

Si vous obtenez le résultat suivant :

```
OPEN_MODE
-----
READ WRITE
```

c'est gagné, votre base fonctionne, il n'y a plus qu'à l'utiliser et surtout, à la remplir. Et oui, Oracle est une base qui permet de stocker des millions d'informations, c'était peut-être un peu riche pour stocker les résultats de vos courses avec vos amis sur Gran Turismo 5 ;) ■

```
oracle@srv-oracle:~$ tnsping bdtest
TNS Ping Utility for Linux: Version 11.2.0.1.0 - Production on 12-OCT-2011 10:57:50
Copyright (c) 1997, 2009, Oracle. All rights reserved.

Used parameter files:
/oracle/app/11g/network/admin/sqlnet.ora

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = srv-oracle.nimp.com)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = bdtest)))
OK (12 msec)
oracle@srv-oracle:~$
oracle@srv-oracle:~$
```

Figure 14. Le résultat de la commande « tnsping »